F E A P
A FINITE ELEMENT ANALYSIS PROGRAM
Version 01/2017
Description and Users’ Manual

Robert L. Taylor

University of California, Berkeley, U.S.A.

Teil 1: Allgemeine Beschreibung
Teil 2: Eingabekommandos
Teil 3: Makrokommandos
Teil 4: Graphische Ausgabe
Teil 5: 3D-Material-Bibliothek
Teil 6: Elemente
Teil 7: Prinzip-Beispiel
Teil 8: FEAP - Editor
Teil 9: FEAP - Netzgenerierung NEGE
Teil 10: FEAP - Netzgenerierung GMESH
Teil 11: FEAP - ISOGEO
Teil 12: FEAP - Fliesslinien Platten CYLT
Teil 13: Beispielsammlung
Teil 14: Neue Elemente
Teil 15: Theorie
Teil 16: Anwendungen und FAQs
Hinweis zur Anwendung


Copyright

• Ohne Genehmigung des Autors ist es nicht gestattet, dieses Heft ganz oder teilweise auf fotomechanischem Wege (Fotokopie, Mikrokopie) zu vervielfältigen.

• Stand: July 4, 2017

• © Prof. Dr.-Ing. W. Wagner
  Institut für Baustatik
  Karlsruhe Institute of Technology
  Postfach 6980
  76128 Karlsruhe

  Telefon: (0721) 608–42280
  Telefax: (0721) 608–46015
  E–mail: info@.ibs.kit.edu
  Internet: http://www.ibs.kit.edu
## Contents

1 Introduction to Users Manual 13

1.1 Allgemeine Beschreibung 13
1.2 Eingabekommandos 14
1.3 Macrokommandos 14
1.4 Graphische Ausgabe 15
1.5 Elemente 15
1.6 Prinzipbeispiel 15
1.7 FEAP–Editor 15
1.8 FEAP–Netzgenerierung NEGE 15
1.9 FEAP–Netzgenerierung GMESH 16
1.10 FEAP–Netzgenerierung CYLT 16
1.11 Beispielsammlung 16
1.12 Hinweise zum Hinzufügen neuer Elemente 16
1.13 Theorie 16

2 Mesh Commands 17

2.1 Available Macros 17
2.2 Overview on commands 18
2.3 Overview on actions 20
2.4 Aufbau FEAP–Eingabefile 22
2.5 Startoptionen für FEAP 22
  2.5.1 Ausführung als Windowsprogramm 22
2.6 Anwendung im reinen Batchbetrieb 22
2.7 Macros in detail 23

3 Macro Commands 127

3.1 Available Macros 127
3.2 Overview on commands 128
3.3 Overview on actions 130
3.4 Macros in detail 132
## 4 Plot Commands

4.1 Available Macros ................................................. 259
4.2 Overview on commands ........................................... 260
4.3 Overview on actions ............................................. 262
4.4 Macros in detail .................................................. 264

## 5 3D-Material library .................................................. 362

## 6 Elements in Student Version ....................................... 371

6.1 Available elements ............................................... 371
6.2 Element numbers .................................................. 373
6.3 Short description of elements .................................... 375
6.4 Detailed description of elements ................................. 378
   6.4.1 ELMT01 .................................................. 378
   6.4.2 ELMT02 .................................................. 379
   6.4.3 ELMT03 .................................................. 381
   6.4.4 ELMT04 .................................................. 383
   6.4.5 ELMT05 .................................................. 385
   6.4.6 ELMT06 .................................................. 387
   6.4.7 ELMT07 .................................................. 388
   6.4.8 ELMT08 .................................................. 390
   6.4.9 ELMT09 .................................................. 392
   6.4.10 ELMT10 .................................................. 395
   6.4.11 ELMT11 .................................................. 396
   6.4.12 ELMT12 .................................................. 398
   6.4.13 ELMT13 .................................................. 403
   6.4.14 ELMT14 .................................................. 405
   6.4.15 ELMT15 .................................................. 406
   6.4.16 ELMT16 .................................................. 408
   6.4.17 ELMT17 .................................................. 409
   6.4.18 ELMT18 .................................................. 411
   6.4.19 ELMT19 .................................................. 413
   6.4.20 ELMT20 .................................................. 414
   6.4.21 ELMT21 .................................................. 415
   6.4.22 ELMT22 .................................................. 418
   6.4.23 ELMT23 .................................................. 419
   6.4.24 ELMT24 .................................................. 420
   6.4.25 ELMT25 .................................................. 422
6.4.26 ELMT26 ......................................................... 424
6.4.27 ELMT27 ......................................................... 425
6.4.28 ELMT28 ......................................................... 425
6.4.29 ELMT29 ......................................................... 425
6.4.30 ELMT30 ......................................................... 426
6.4.31 ELMT31 ......................................................... 427
6.4.32 ELMT32 ......................................................... 427
6.4.33 ELMT33 ......................................................... 428
6.4.34 ELMT34 ......................................................... 430
6.4.35 ELMT35 ......................................................... 430
6.4.36 ELMT36 ......................................................... 430
6.4.37 ELMT37 ......................................................... 430
6.4.38 ELMT38 ......................................................... 430
6.4.39 ELMT39 ......................................................... 430
6.4.40 ELMT40 ......................................................... 431
6.4.41 ELMT41 ......................................................... 431
6.4.42 ELMT42 ......................................................... 431
6.4.43 ELMT43 ......................................................... 431
6.4.44 ELMT44 ......................................................... 431
6.4.45 ELMT45 ......................................................... 432
6.4.46 ELMT46 ......................................................... 434
6.4.47 ELMT47 ......................................................... 434
6.4.48 ELMT48 ......................................................... 434

7 Principal example ................................................. 435

8 F E A P - E D Editor ................................................. 439

9 F E A P - NEGE Mesher ........................................... 440

10 F E A P - GMESH Mesher ........................................ 452

11 F E A P - ISOGEO .................................................. 459
   11.1 Overview ..................................................... 459
   11.2 Terminology .................................................. 459
   11.3 Known Errors ................................................ 459

12 F E A P – CYLT Mesh Generator for Yield-Line Predictions ................. 460

Institut für Baustatik, KIT
13 F E A P - Beispielsammlung

13.1 Beispiele für 2D/3D - Fachwerkelement

13.1.1 Beispiel 1 - Dachbinder

13.1.1.1 System und Belastung

13.1.1.2 Eingabedatensatz (file: it01)

13.1.1.3 Ergebnisse

13.1.1.4 Vergleichslösung

13.1.1.5 verformte Struktur (20-fach überhöht)

13.1.2 Beispiel 2 - Kran

13.1.2.1 System und Belastung

13.1.2.2 Eingabedatensatz (file: it02)

13.1.2.3 Ergebnisse

13.1.2.4 Vergleichslösung

13.1.2.5 verformte Struktur (20-fach überhöht)

13.1.3 Beispiel 3 - Brückenhöhe

13.1.3.1 System und Belastung

13.1.3.2 Eingabedatensatz (file: it03)

13.1.3.3 Ergebnisse

13.1.3.4 Struktur und Verschiebungen

13.1.4 Beispiel 4 - Raumfachwerk

13.1.4.1 System und Belastung

13.1.4.2 Eingabedatensatz (file: it04)

13.1.4.3 Ergebnisse

13.1.4.4 Struktur und Verschiebungen

13.2 Beispiele für 2D - Stabelement

13.2.1 Beispiel 1 - Zweigelenkrahmen

13.2.1.1 System und Belastung

13.2.1.2 Eingabedatensatz für Lastfall 1 (file: ib2d1a)

13.2.1.3 Eingabedatensatz für Lastfall 2 (file: ib2d1b)

13.2.1.4 Eingabedatensatz für Lastfall 3 (file: ib2d1c)

13.2.1.5 Ergebnisse

13.2.1.6 Vergleichslösung

13.2.2 Beispiel 2 - Rahmentragwerk

13.2.2.1 System und Belastung

13.2.2.2 Eingabedatensatz (file: ib2d2)

13.2.2.3 Ergebnisse

13.2.2.4 Vergleichslösung

Institut für Baustatik, KIT
13.2.2.5 Schnittkraftlinie ................................. 484
13.2.3 Beispiel 3 - elastisch gebetteter Balken ......................... 486
  13.2.3.1 System und Belastung .................................. 486
  13.2.3.2 Eingabedatensatz (file: ib2d3) ............................. 486
  13.2.3.3 Ergebnisse ............................................. 486
  13.2.3.4 Vergleichslösung ........................................ 486
13.2.4 Beispiel 4 - Halbrahmen ........................................ 487
  13.2.4.1 System und Belastung ..................................... 487
  13.2.4.2 Eingabedatensatz (file: ib2d4) ............................. 487
  13.2.4.3 Ergebnisse ............................................. 487
  13.2.4.4 Vergleichslösung ........................................ 487
13.3 Beispiele für 3D - Stabelement .................................... 488
  13.3.1 Beispiel 1 - räumliche Rahmenecke ......................... 488
    13.3.1.1 System und Belastung ................................. 488
    13.3.1.2 Eingabedatensatz (file: ib3d1) ......................... 488
    13.3.1.3 Ergebnisse ............................................. 488
    13.3.1.4 Vergleichslösung ........................................ 488
  13.3.2 Beispiel 2 - räumlicher Rahmen ................................. 489
    13.3.2.1 System und Belastung .................................. 489
    13.3.2.2 Eingabedatensatz (file: ib3d2) ............................. 489
    13.3.2.3 Ergebnisse ............................................. 489
    13.3.2.4 Vergleichslösung ........................................ 489
13.4 Beispiele für Scheibenelemente .................................... 490
  13.4.1 Beispiel 1 - Kragarm mit Einzellast ......................... 490
    13.4.1.1 System und Belastung .................................. 490
    13.4.1.2 Eingabedatensatz für Beispiel 1 (file: is1) .............. 490
    13.4.1.3 Ergebnisse ............................................. 491
    13.4.1.4 Vergleichslösung ........................................ 491
  13.4.2 Beispiel 2 - Wandscheibe mit Gleichstreckenlast auf zwei Stützen .................................................................................................................. 492
    13.4.2.1 System und Belastung .................................. 492
    13.4.2.2 Eingabedatensatz (file: is2) ............................. 493
    13.4.2.3 Ergebnisse ............................................. 494
    13.4.2.4 Vergleichslösung ........................................ 495
  13.4.3 Beispiel 3 - Wandscheibe mit Loch ............................... 496
    13.4.3.1 System und Belastung .................................. 496
    13.4.3.2 NEGE-Eingabedatensatz (file: is3.neg) .................... 497
    13.4.3.3 Ergebnisse – verformtes Netz ............................. 498
13.4.3.4 Vergleichslösung .......................................................... 498
13.5 Beispiele für Plattenelement .................................................. 499
  13.5.1 Beispiel 1 - Platte mit verschiedenen Lagerbedingungen .......... 499
    13.5.1.1 Systeme und Belastung ........................................ 499
    13.5.1.2 Eingabedatensatz ............................................... 500
    13.5.1.3 Ergebnisse ................................................... 501
    13.5.1.4 Vergleichslösungen ........................................... 502
    13.5.1.5 Bemessung der unterstützten Platte .......................... 503
13.6 Beispiele für Rotationsschalenelement ..................................... 504
  13.6.1 Beispiel 1 - Zylinder mit Randlast .................................. 504
    13.6.1.1 System und Belastung ........................................ 504
    13.6.1.2 Eingabedatensatz (file: isr-1) .............................. 504
    13.6.1.3 Ergebnisse ................................................... 505
  13.6.2 Beispiel 2 - Zylinder mit Wasserfüllung ............................ 506
    13.6.2.1 System und Belastung ........................................ 506
    13.6.2.2 Eingabedatensatz (file: isr-2) .............................. 506
    13.6.2.3 Ergebnisse ................................................... 507
  13.6.3 Beispiel 3 - Zylinder mit gleichmäßiger Temperaturbelastung ...... 508
    13.6.3.1 System und Belastung ........................................ 508
    13.6.3.2 Eingabedatensatz (file: isr-6) .............................. 508
    13.6.3.3 Ergebnisse ................................................... 508
  13.6.4 Beispiel 4 - Zylinder mit ungleichmäßiger Temperaturbelastung ... 509
    13.6.4.1 System und Belastung ........................................ 509
    13.6.4.2 Eingabedatensatz (file: isr-7) .............................. 509
    13.6.4.3 Ergebnisse ................................................... 509
  13.6.5 Beispiel 5 - Kegel mit Randmoment .................................. 510
    13.6.5.1 System und Belastung ........................................ 510
    13.6.5.2 Eingabedatensatz (file: isr-3) .............................. 510
    13.6.5.3 Ergebnisse ................................................... 511
  13.6.6 Beispiel 6 - Kugel mit Innendruck .................................. 512
    13.6.6.1 System und Belastung ........................................ 512
    13.6.6.2 Eingabedatensatz (file: isr-4) .............................. 512
    13.6.6.3 Ergebnisse ................................................... 513
  13.6.7 Beispiel 7 - Kreisplatte mit Flächenlast ............................ 514
    13.6.7.1 System und Belastung ........................................ 514
    13.6.7.2 Eingabedatensatz (file: isr-5) .............................. 514
  13.6.8 Beispiel 8 - Kreisplatte mit Loch unter Temperaturbelastung ..... 515
13.6.8.1 System und Belastung ........................................ 515
13.6.8.2 Eingabedatensatz (file: isr-8) ............................... 515
13.7 Beispiele für allgemeines Schalenelement ....................... 516
  13.7.1 Beispiel 1 - Eingespannter T-Träger ......................... 516
    13.7.1.1 System und Belastung ................................ 516
    13.7.1.2 Systemdaten für eine analytische Vergleichslösung ........ 516
    13.7.1.3 Eingabedatensatz (file: isg-1) .......................... 517
  13.7.2 Beispiel 2 - Beidseitig eingespannter Zylinder mit Einzellasten . 518
    13.7.2.1 System und Belastung ................................ 518
    13.7.2.2 Eingabedatensatz (file: isg-2) .......................... 518
    13.7.2.3 Ergebnisse und verformtes Netz .......................... 519
  13.7.3 Beispiel 3 - Membrangelagertes Tonnendach ................ 520
    13.7.3.1 System und Belastung ................................ 520
    13.7.3.2 Eingabedatensatz (file: isg-3) .......................... 520
    13.7.3.3 Schnittkraftverläufe und Ergebnisvergleich ............... 521

14 Adding elements 522
  14.1 General information ........................................... 522
    14.1.1 Name of Subprogram, parameters ......................... 522
    14.1.2 Control Switch Parameter Values ......................... 525
  14.2 Common Blocks ............................................... 526
  14.3 Treatment of history terms .................................... 529
    14.3.1 Assigning amount of storage for each element ............ 529
    14.3.2 Accessing history data for each element ................. 530
    14.3.3 Typical 2D-element framework for history data .......... 531
  14.4 Typical element framework .................................... 532
  14.5 Use of Parameters and Expression Inputs in New Elements .. 533
  14.6 Error analysis ............................................... 534
  14.7 Logical Flags ................................................. 534
  14.8 3D-Material library .......................................... 535
  14.9 Adding elements for FE² ...................................... 536
  14.10 Plot of user defined data .................................... 537
  14.11 Developing 2D-/3D-elements with higher order shape functions . 538
# 15 Theory Manual

15.1 Programming structure of FEAP .......................................................... 539
   15.1.1 General structure ................................................................. 539
   15.1.2 Calculating arrays on element level .......................................... 540
   15.1.3 Displacement arrays .............................................................. 540
   15.1.4 Making FEAP faster ............................................................... 541

15.2 Nonlinear Iteration Algorithms ......................................................... 541

15.3 Stability analysis .............................................................................. 542

15.4 Time integration procedures ............................................................ 544
   15.4.1 Newmark–method .................................................................. 544
   15.4.2 Overview Implicit Schemes ...................................................... 545
   15.4.3 Explicit Schemes .................................................................... 545
   15.4.4 Energy-conserving algorithm ................................................ 546
   15.4.5 Implicit composite scheme - Bathe ......................................... 547
   15.4.6 Automatic time stepping procedure ....................................... 548

15.5 PCG-Methods .................................................................................. 550

15.6 PGMRES-Methods ......................................................................... 552

15.7 Error Analysis .................................................................................. 553

15.8 Eigenvalue Computations ................................................................. 554
   15.8.1 Subspace Iteration .................................................................. 554
   15.8.2 Lanczos Iteration .................................................................... 555

15.9 Augmented Lagrange Formulation .................................................. 557

15.10 Theory of element formulations ...................................................... 558
   15.10.1 3D-Material library ................................................................. 558
      15.10.1.1 Material model 1: .......................................................... 558
      15.10.1.2 Material model 2: .......................................................... 559
      15.10.1.3 Material model 3: .......................................................... 559
      15.10.1.4 Material model 4: .......................................................... 560
      15.10.1.5 Material model 5: .......................................................... 560
      15.10.1.6 Material model 6: .......................................................... 561
      15.10.1.7 Material model 7: .......................................................... 561
      15.10.1.8 Material model 8: .......................................................... 562
      15.10.1.9 Material model 9: .......................................................... 570
      15.10.1.10 Material model 10: ...................................................... 570
      15.10.1.11 Material model 11: ....................................................... 570
      15.10.1.12 Material model 12: ....................................................... 571
      15.10.1.13 Material model 13: ....................................................... 571
15.10.1.1 Material model 14: .................................................. 571
15.10.1.15 Material model 15: .................................................. 572
15.10.1.16 Material model 16: .................................................. 573

16 Applications and FAQs .................................................. 574

16.1 FAQs ........................................................................ 574
  16.1.1 Crash within start procedure ................................ 574
  16.1.2 Mixing elements .................................................... 574

16.2 Some details using FEAP ............................................. 575
  16.2.1 Use of macro prop .................................................. 575
  16.2.2 Use of macro newf .................................................. 577
Chapter 1

Introduction to Users Manual

1.1 Allgemeine Beschreibung


Aufgrund der allgemeinen Einsetzbarkeit des Programmes, aufgrund der ständigen Modifikationen und des wissenschaftlichen Versuchscharakters können und wollen die Programmersteller nicht mit kommerziellen Finite-Element-Programmen konkurrieren, insbesondere den Spezialprogrammen im Baubereich.

Das Programm ist so ausgelegt, daß – auf der Basis einer mittels Editor erstellten Eingabedatei – die Bearbeitung eines Problems, d.h. die Berechnung und die graphische Auswertung interaktiv erfolgt. Dabei steht eine ‘On–Line’–Dokumentation aller zu verwendenden Befehle zur Verfügung.

Die Beschreibung der Dateneingabe, der Berechnung und der graphischen Darstellung ist in den Teilen 2-4 beschrieben. Teil 5 enthält die Beschreibung der zur Verfügung stehenden Elemente, während in Teil 6 eine umfangreiche Beispielsammlung zur Verfügung steht.

Die nachfolgenden Teile 2 - 6.1 sind in englischer Sprache dargestellt, was in der direkten Zusammenarbeit mit Prof. Taylor aus Berkeley begründet ist. Man betrachte dies nicht als Nachteil, sondern vielmehr als Ansporn, sich mit einer anderen Sprache auseinanderzusetzen und als ersten Schritt, auch wissenschaftliche Literatur in Fremdsprachen zu lesen.
1.2 Eingabekommandos


Die Dateneingabe läßt sich wesentlich vereinfachen, wenn zu Beginn des Datensatzes unter dem Macro para Parameter belegt werden, die bei der späteren Eingabe berücksichtigt werden. So können einem Knoten beispielsweise die Koordinaten \( x = l, y = 0.5 \cdot l, z = (l + 3) \cdot 2 \) zugewiesen werden. Das Beispiel zeigt, daß auch arithmetische Ausdrücke zugelassen sind.

Die wesentliche Daten, die zur Definition eines Netzes benötigt werden, sind in der folgenden Tabelle aufgeführt:

<table>
<thead>
<tr>
<th>Eingabe</th>
<th>Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Globale Problemdefinition</td>
<td>feap</td>
</tr>
<tr>
<td>* Parameterdefinition</td>
<td>para</td>
</tr>
<tr>
<td>* Knotenkoordinaten</td>
<td>coor</td>
</tr>
<tr>
<td>* Elementtopologie (Zusammenhang des FE-Netzes)</td>
<td>elem</td>
</tr>
<tr>
<td>* Randbedingungen</td>
<td>boun</td>
</tr>
<tr>
<td>* Belastungen</td>
<td>load</td>
</tr>
<tr>
<td>* Materialeigenschaften, Elementzuordnung</td>
<td>mate</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Eingabe</th>
<th>Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Generierung von 2D-Netzen</td>
<td>bloc</td>
</tr>
<tr>
<td>* Generierung von 2D-Netzen</td>
<td>nege</td>
</tr>
<tr>
<td>* Generierung von 2D-Netzen</td>
<td>gmesh</td>
</tr>
<tr>
<td>* Generierung von Randbedingungen</td>
<td>ebou</td>
</tr>
<tr>
<td>* Generierung von Randbedingungen</td>
<td>edge</td>
</tr>
<tr>
<td>* Generierung von Randlasten</td>
<td>eloa</td>
</tr>
<tr>
<td>* Umrechnung von Polar- auf Kartesisches KOS</td>
<td>pola</td>
</tr>
</tbody>
</table>

Im Teil 2 des Manuals werden die oben angegebenen Macros in alphabetischer Reihenfolge näher beschrieben und an Beispielen gezeigt, wie die jeweilige Eingabe bei einem Macro vorzunehmen ist.

1.3 Macrokommandos

1.4 Graphische Ausgabe

In diesem Teil des Manuals werden die graphischen Ausgabemöglichkeiten des Programmes FEAP beschrieben. Das Ansprechen der Graphik geschieht wieder über Macrobefehle analog den Kommandos auf der Macroebene. Ein typischer Plotbefehl lautet z.B.

\texttt{load, n1, n2, n3 = load, -1, 1, 2}

Damit werden alle Lasten auf das verformte Netz (n1) mit einem Vergrößerungsfaktor (n3) gezeichnet. Die Spitze der Lastvektoren zeigt dabei auf den Knoten (n2). Es können die Knoten mit Nummern, die Systeme, die Elementnummern, die Randbedingungen, die Belastungen, die Reaktionskräfte sowie die unterschiedlichen Materialtypen dargestellt werden.

Für Stäbe kann der Schnittgrößenverlauf, für Scheiben, Platten und Schalen die flächenhaften Verteilung von Schnittgrößen (Spannungen) und Verschiebungen dargestellt werden. Alle Größen können in der Regel bezogen auf das unverformte als auch bezogen auf das verformte System geplottet werden.

Weiterhin können Spannungs- und Verschiebungsverläufe bei Scheiben, Platten und Schalen in Schnitten berechnet und geplottet werden.

1.5 Elemente

Dieser Teil des Manuals enthält die Beschreibung der zur vorliegenden Studentenversion FEAP gehörenden finiten Elemente. Es werden die Eingabedaten, die möglichen Macro-daten sowie die Plotoptionen beschrieben.


1.6 Prinzipbeispiel

Hier wird ein \textit{Beispiel} exemplarisch detailliert erklärt.

1.7 FEAP–Editor

In diesem Abschnitt werden Informationen zum \textit{FEAP–Editor} gegeben.

1.8 FEAP–Netzgenerierung NEGE

In diesem Abschnitt werden Informationen zur 2D–Netzgenerierung mit \textit{NEGE} gegeben.
1.9 FEAP-Netzgenerierung GMESH

In diesem Abschnitt werden Informationen zur 2D/3D-Netzgenerierung mit GMESH gegeben.

1.10 FEAP-Netzgenerierung CYLT

In diesem Abschnitt werden Informationen zur 2D-Netzgenerierung mit CYLT gegeben.

1.11 Beispielsammlung


1.12 Hinweise zum Hinzufügen neuer Elemente

Dieser Teil des Manuals enthält Programmierinformationen zum Hinzufügen von neuen Elementen.

1.13 Theorie

Der letzte Teil des Manuals enthält ausgewählte theoretische Hintergründe zu Algorithmen und einzelnen Elementen.
Chapter 2

Mesh Commands

2.1 Available Macros

The following entries are available for the control information and input of the mesh:

<table>
<thead>
<tr>
<th>feap</th>
<th>aloa</th>
<th>angl</th>
<th>back</th>
<th>base</th>
<th>blco</th>
<th>bloc</th>
<th>blox</th>
</tr>
</thead>
<tbody>
<tr>
<td>boun</td>
<td>btem</td>
<td>cmod</td>
<td>coor</td>
<td>curv</td>
<td>cylt</td>
<td>debu</td>
<td>disp</td>
</tr>
<tr>
<td>eang</td>
<td>ebou</td>
<td>edge</td>
<td>edis</td>
<td>elem</td>
<td>elfr</td>
<td>eloa</td>
<td>el3b</td>
</tr>
<tr>
<td>end</td>
<td>epsq</td>
<td>fast</td>
<td>fixe</td>
<td>gbou</td>
<td>gcor</td>
<td>gele</td>
<td>geom</td>
</tr>
<tr>
<td>gmesh</td>
<td>icon</td>
<td>impf</td>
<td>inte</td>
<td>icor</td>
<td>isec</td>
<td>jint</td>
<td>knv1</td>
</tr>
<tr>
<td>knv2</td>
<td>link</td>
<td>load</td>
<td>loa0</td>
<td>macro</td>
<td>mate</td>
<td>mesn</td>
<td>nege</td>
</tr>
<tr>
<td>neco</td>
<td>ndvi</td>
<td>nmpq</td>
<td>nopa</td>
<td>nopr</td>
<td>opti</td>
<td>para</td>
<td>pars</td>
</tr>
<tr>
<td>poin</td>
<td>pola</td>
<td>pres</td>
<td>prin</td>
<td>qloa</td>
<td>rbou</td>
<td>regi</td>
<td>rndm</td>
</tr>
<tr>
<td>rot</td>
<td>rsum</td>
<td>segm</td>
<td>sloa</td>
<td>solv</td>
<td>sphe</td>
<td>stop</td>
<td>temp</td>
</tr>
<tr>
<td>tie</td>
<td>tran</td>
<td>trib</td>
<td>vang</td>
<td>vbou</td>
<td>ybou</td>
<td>ycon</td>
<td>yedg</td>
</tr>
<tr>
<td>yloa</td>
<td>ynod</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each of these commands describes a specific function to be used in defining the problem to be solved.

A short overview on commands is given in section 2.2, see macro Index in FEAP, whereas possible actions can be found in section 2.3, see macro Action in FEAP.
## 2.2 Overview on commands

<table>
<thead>
<tr>
<th>Macro</th>
<th>Aufgabe</th>
</tr>
</thead>
<tbody>
<tr>
<td>aloa</td>
<td>Lasten auf Gebieten</td>
</tr>
<tr>
<td>angl</td>
<td>Basis lokal: Knoten</td>
</tr>
<tr>
<td>back</td>
<td>Hintergrundnetz bei Generierung</td>
</tr>
<tr>
<td>bico</td>
<td>Knoten+Elemente: Generierung Interface Zone</td>
</tr>
<tr>
<td>bloc</td>
<td>Knoten+Elemente: 2D+3D Generierung</td>
</tr>
<tr>
<td>base</td>
<td>Dreibeinberechnung</td>
</tr>
<tr>
<td>blocx</td>
<td>Knoten+Elemente: 2D Generierung</td>
</tr>
<tr>
<td>boun</td>
<td>Randbedingungen an Knoten</td>
</tr>
<tr>
<td>btem</td>
<td>Temperaturen: 2D+ 3D Generierung</td>
</tr>
<tr>
<td>cmod</td>
<td>Koordinaten: spezielle Modifikationen</td>
</tr>
<tr>
<td>coor</td>
<td>Koordinaten</td>
</tr>
<tr>
<td>curv</td>
<td>Definition von Kurven</td>
</tr>
<tr>
<td>cylt</td>
<td>Titelzeile + Generierungstyp bei CYLT</td>
</tr>
<tr>
<td>debu</td>
<td>Debug - Option</td>
</tr>
<tr>
<td>eang</td>
<td>Basis lokal: Linie</td>
</tr>
<tr>
<td>ebour</td>
<td>Randbedingungen für eine feste Koord.</td>
</tr>
<tr>
<td>edge</td>
<td>Randbedingungen für eine Linie</td>
</tr>
<tr>
<td>edis</td>
<td>Werte für Randbedingungen für eine feste Koord.</td>
</tr>
<tr>
<td>el3b</td>
<td>Elemente fuer 3D - Stab</td>
</tr>
<tr>
<td>elem</td>
<td>Elemente</td>
</tr>
<tr>
<td>elfr</td>
<td>Elemente, freie Numerierung</td>
</tr>
<tr>
<td>eloa</td>
<td>Linienlasten</td>
</tr>
<tr>
<td>end</td>
<td>Ende Eingabedaten</td>
</tr>
<tr>
<td>epsq</td>
<td>Verzerrungslasten</td>
</tr>
<tr>
<td>fast</td>
<td>Binär-Inputfile einlesen</td>
</tr>
<tr>
<td>feap</td>
<td>Titelzeile + Problemgrössen</td>
</tr>
<tr>
<td>fixe</td>
<td>Randbedingungen an Segmenten bei Generierung</td>
</tr>
<tr>
<td>gbou</td>
<td>Koordinaten bei Generierung GMESH</td>
</tr>
<tr>
<td>gcor</td>
<td>Elemente bei Generierung GMESH</td>
</tr>
<tr>
<td>gelem</td>
<td>Koordinaten der Segmente bei Generierung</td>
</tr>
<tr>
<td>gmesh</td>
<td>Titelzeile + Problemgroessen bei GMESH</td>
</tr>
<tr>
<td>icon</td>
<td>Kontaktdateneingabe</td>
</tr>
<tr>
<td>icor</td>
<td>Koordinaten: Imperfektionen</td>
</tr>
<tr>
<td>impf</td>
<td>Imperfektionen</td>
</tr>
<tr>
<td>inte</td>
<td>Interaktive Eingabe der Macros</td>
</tr>
<tr>
<td>isec</td>
<td>Verschneidungen</td>
</tr>
<tr>
<td>jint</td>
<td>$J_2$-Integraleingabe</td>
</tr>
<tr>
<td>knv1</td>
<td>Knotenvektoren $\Xi^1$ für FEAP - ISOGEO</td>
</tr>
<tr>
<td>knv2</td>
<td>Knotenvektoren $\Xi^2$ für FEAP - ISOGEO</td>
</tr>
<tr>
<td>link</td>
<td>Verbindung von Freiheitsgraden</td>
</tr>
<tr>
<td>load</td>
<td>Knotenlasten</td>
</tr>
<tr>
<td>loa0</td>
<td>ständige Knotenlasten F0</td>
</tr>
<tr>
<td>Macro</td>
<td>Aufgabe</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>macr</td>
<td>Macro - Modus</td>
</tr>
<tr>
<td>mate</td>
<td>Materialdaten</td>
</tr>
<tr>
<td>mesn</td>
<td>Anwender Eingabe Macro</td>
</tr>
<tr>
<td>ndvi</td>
<td>Unterteilung bei Generierung GMESH</td>
</tr>
<tr>
<td>neco</td>
<td>Definition von Konstanten bei Generierung</td>
</tr>
<tr>
<td>nege</td>
<td>Titelzeile + Problemgrössen bei NEGE</td>
</tr>
<tr>
<td>nmpq</td>
<td>Geometrieparameter f. FEAP - ISOGEO</td>
</tr>
<tr>
<td>nopa</td>
<td>Dateneingabe nur Zahlen</td>
</tr>
<tr>
<td>norp</td>
<td>Datenausgabe: aus</td>
</tr>
<tr>
<td>opti</td>
<td>Knotennummernoptimierung</td>
</tr>
<tr>
<td>optn</td>
<td>Knotennummernoptimierung</td>
</tr>
<tr>
<td>para</td>
<td>Definition von Konstanten</td>
</tr>
<tr>
<td>pars</td>
<td>Dateneingabe allgemeine Ausdrücke</td>
</tr>
<tr>
<td>poin</td>
<td>Randbedingungen an einem Punkt</td>
</tr>
<tr>
<td>poin</td>
<td>Knotenlasten an einem Punkt</td>
</tr>
<tr>
<td>pola</td>
<td>Koordinaten: polar → kartesisch</td>
</tr>
<tr>
<td>pres</td>
<td>Linienlasten auf Segmenten bei Generierung</td>
</tr>
<tr>
<td>prin</td>
<td>Datenausgabe: ein</td>
</tr>
<tr>
<td>qloa</td>
<td>Elementlasten</td>
</tr>
<tr>
<td>rbou</td>
<td>Randbedingungen in einem Zylinderbereich</td>
</tr>
<tr>
<td>nsum</td>
<td>Koordinaten: Zufallsveränderung</td>
</tr>
<tr>
<td>rsum</td>
<td>Reaktionskräfte bei mehreren Knoten</td>
</tr>
<tr>
<td>regi</td>
<td>Regionen bei Generierung</td>
</tr>
<tr>
<td>rot</td>
<td>Koordinaten: Rotation um Achse</td>
</tr>
<tr>
<td>segm</td>
<td>Segmente bei Generierung</td>
</tr>
<tr>
<td>sloa</td>
<td>Oberflächen- bzw. Folgelasten</td>
</tr>
<tr>
<td>solv</td>
<td>Gleichungslöser</td>
</tr>
<tr>
<td>sph</td>
<td>Koordinaten: kugel → kartesisch</td>
</tr>
<tr>
<td>stop</td>
<td>Ende der Berechnung/ letzte Karte Eingabe</td>
</tr>
<tr>
<td>temp</td>
<td>Temperaturen</td>
</tr>
<tr>
<td>tie</td>
<td>Verbindung von Knoten mit gl. Koordinaten</td>
</tr>
<tr>
<td>trans</td>
<td>Koordinaten: Translation</td>
</tr>
<tr>
<td>trib</td>
<td>Knoten/Elemente 2D+3D Generierung Dreiecke</td>
</tr>
<tr>
<td>vang</td>
<td>Basis lokal: Region</td>
</tr>
<tr>
<td>vbou</td>
<td>Randbedingungen in einer Region</td>
</tr>
<tr>
<td>ybou</td>
<td>Randbedingungen bei Fliesslinien</td>
</tr>
<tr>
<td>ycon</td>
<td>Definition von Konstanten bei Generierung CYLT</td>
</tr>
<tr>
<td>yedg</td>
<td>Kantengenerierung bei Generierung CYLT</td>
</tr>
<tr>
<td>yloa</td>
<td>Lasteingabe bei Fliesslinien</td>
</tr>
<tr>
<td>ynod</td>
<td>Knoteneingabe bei Generierung CYLT</td>
</tr>
</tbody>
</table>
## 2.3 Overview on actions

<table>
<thead>
<tr>
<th>Aufgabe</th>
<th>Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anwender Eingabe Macro</td>
<td>mesn</td>
</tr>
<tr>
<td>Basis lokal: Knoten</td>
<td>angl</td>
</tr>
<tr>
<td>Basis lokal: Linie</td>
<td>eang</td>
</tr>
<tr>
<td>Basis lokal: Region</td>
<td>vang</td>
</tr>
<tr>
<td>Binär-Inputfile einlesen</td>
<td>fast</td>
</tr>
<tr>
<td>Datenausgabe: ein</td>
<td>prin</td>
</tr>
<tr>
<td>Datenausgabe: aus</td>
<td>nopr</td>
</tr>
<tr>
<td>Dateneingabe: allgemeine Ausdrücke</td>
<td>pars</td>
</tr>
<tr>
<td>Dateneingabe: nur Zahlen</td>
<td>nopa</td>
</tr>
<tr>
<td>Debug - Option</td>
<td>debu</td>
</tr>
<tr>
<td>Definition von Konstanten</td>
<td>para</td>
</tr>
<tr>
<td>Definition von Konstanten bei Fliesslinien</td>
<td>ycon</td>
</tr>
<tr>
<td>Definition von Konstanten bei Generierung</td>
<td>neco</td>
</tr>
<tr>
<td>Definition von Kurven</td>
<td>curve</td>
</tr>
<tr>
<td>Dreibeinberechnung</td>
<td>base</td>
</tr>
<tr>
<td>Elemente</td>
<td>elem</td>
</tr>
<tr>
<td>Elemente, freie Numerierung</td>
<td>elfr</td>
</tr>
<tr>
<td>Elemente bei Generierung GMESH</td>
<td>gele</td>
</tr>
<tr>
<td>Elemente fuer 3D - Stab</td>
<td>el3b</td>
</tr>
<tr>
<td>Elementlasten</td>
<td>qloa</td>
</tr>
<tr>
<td>Ende der Berechnung/ letzte Karte Eingabe</td>
<td>stop</td>
</tr>
<tr>
<td>Ende Eingabedaten Mesh</td>
<td>end</td>
</tr>
<tr>
<td>Geometrieparameter f. FEAP - ISOGEO</td>
<td>nmpq</td>
</tr>
<tr>
<td>Gleichungslöser</td>
<td>solv</td>
</tr>
<tr>
<td>Hintergrundnetz bei Generierung</td>
<td>back</td>
</tr>
<tr>
<td>Imperfektionen</td>
<td>impf</td>
</tr>
<tr>
<td>Interaktive Eingabe der Macros</td>
<td>inte</td>
</tr>
<tr>
<td>J₂-Integraleingabe</td>
<td>jint</td>
</tr>
<tr>
<td>Kantengenerierung bei Fliesslinien</td>
<td>yedg</td>
</tr>
<tr>
<td>Knoteneingabe bei Fliesslinien</td>
<td>ynod</td>
</tr>
<tr>
<td>Knoten+Elemente: 2D+3D Generierung</td>
<td>bloc</td>
</tr>
<tr>
<td>Knoten+Elemente: 2D Generierung</td>
<td>blox</td>
</tr>
<tr>
<td>Knoten+Elemente: 2D Interface Zone</td>
<td>blco</td>
</tr>
<tr>
<td>Knotenvektoren $\Xi^1$ für FEAP - ISOGEO</td>
<td>knv1</td>
</tr>
<tr>
<td>Knotenvektoren $\Xi^2$ für FEAP - ISOGEO</td>
<td>knv2</td>
</tr>
<tr>
<td>Koordinaten bei Generierung GMESH</td>
<td>geor</td>
</tr>
<tr>
<td>Knoten+Elemente: 2D+3D Generierung Dreiecke</td>
<td>trib</td>
</tr>
<tr>
<td>Knotenlasten</td>
<td>load</td>
</tr>
<tr>
<td>Knotenlasten (ständig)</td>
<td>loa0</td>
</tr>
<tr>
<td>Knotenlasten an einem Punkt</td>
<td>poin</td>
</tr>
<tr>
<td>Knotennummeroptimierung</td>
<td>opti</td>
</tr>
<tr>
<td>Knotennummernoptimierung</td>
<td>optn</td>
</tr>
<tr>
<td>Kontaktdataeingabe</td>
<td>icon</td>
</tr>
<tr>
<td>Koordinaten</td>
<td>coor</td>
</tr>
<tr>
<td>Aufgabe</td>
<td>Macro</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Koordinaten der Segmente bei Generierung</td>
<td>geom</td>
</tr>
<tr>
<td>Koordinaten: Imperfektionen</td>
<td>icor</td>
</tr>
<tr>
<td>Koordinaten: kugel → kart.</td>
<td>sphe</td>
</tr>
<tr>
<td>Koordinaten: polar → kart.</td>
<td>pola</td>
</tr>
<tr>
<td>Koordinaten: Rotation um Achse</td>
<td>rot</td>
</tr>
<tr>
<td>Koordinaten: Translation</td>
<td>tran</td>
</tr>
<tr>
<td>Koordinaten: spezielle Modifikationen</td>
<td>cmod</td>
</tr>
<tr>
<td>Koordinaten: Zufallsveränderung</td>
<td>rndm</td>
</tr>
<tr>
<td>Lasteingabe bei Fließlinien</td>
<td>yloa</td>
</tr>
<tr>
<td>Lasten auf Gebieten</td>
<td>aloa</td>
</tr>
<tr>
<td>Linienlasten</td>
<td>eloa</td>
</tr>
<tr>
<td>Linienlasten auf Segmenten bei Generierung</td>
<td>pres</td>
</tr>
<tr>
<td>Macro - Modus</td>
<td>macr</td>
</tr>
<tr>
<td>Materialdaten</td>
<td>mate</td>
</tr>
<tr>
<td>Oberflächen- bzw. Folgelasten</td>
<td>sloa</td>
</tr>
<tr>
<td>Randbedingungen an Knoten</td>
<td>boun</td>
</tr>
<tr>
<td>Randbedingungen an einem Punkt</td>
<td>poin</td>
</tr>
<tr>
<td>Randbedingungen an Segmenten bei Generierung</td>
<td>fixe</td>
</tr>
<tr>
<td>Randbedingungen bei Fließlinien</td>
<td>ybou</td>
</tr>
<tr>
<td>Randbedingungen fuer eine feste Koord.</td>
<td>ebou</td>
</tr>
<tr>
<td>Randbedingungen fuer eine Linie</td>
<td>edge</td>
</tr>
<tr>
<td>Randbedingungen in einer Region</td>
<td>vbou</td>
</tr>
<tr>
<td>Randbedingungen in einem Zylinderbereich</td>
<td>rbou</td>
</tr>
<tr>
<td>Randbedingungen: Werte fuer eine feste Koord.</td>
<td>edis</td>
</tr>
<tr>
<td>Reaktionskräfte bei mehreren Knoten</td>
<td>rsum</td>
</tr>
<tr>
<td>Regionen bei Generierung</td>
<td>regi</td>
</tr>
<tr>
<td>Segmente bei Generierung</td>
<td>segm</td>
</tr>
<tr>
<td>Temperaturen</td>
<td>temp</td>
</tr>
<tr>
<td>Temperaturen: 2D+ 3D Generierung</td>
<td>btem</td>
</tr>
<tr>
<td>Titelzeile + Generierungstyp bei Gen. CYLT</td>
<td>cylt</td>
</tr>
<tr>
<td>Titelzeile + Problemgrössen</td>
<td>feap</td>
</tr>
<tr>
<td>Titelzeile + Problemgrössen bei Gen. GMESH</td>
<td>gmesh</td>
</tr>
<tr>
<td>Titelzeile + Problemgrössen bei Gen. NEGE</td>
<td>nege</td>
</tr>
<tr>
<td>Unterteilung bei Generierung GMESH</td>
<td>ndvi</td>
</tr>
<tr>
<td>Verbindung von Freiheitsgraden</td>
<td>link</td>
</tr>
<tr>
<td>Verbindung von Knoten mit gl. Koordinaten</td>
<td>tie</td>
</tr>
<tr>
<td>Verzerrungs<code>lasten</code></td>
<td>epsq</td>
</tr>
<tr>
<td>Verschneidungen</td>
<td>isec</td>
</tr>
</tbody>
</table>
2.4 Aufbau FEAP–Eingabefile

<table>
<thead>
<tr>
<th>Feap oder Gmesh oder Nege</th>
<th>Kopfzeile</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>MESH–Macros</td>
</tr>
<tr>
<td>↓</td>
<td>end</td>
</tr>
<tr>
<td>&lt;opti&gt;</td>
<td>Eingabe, falls nötig</td>
</tr>
<tr>
<td>&lt;tie&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;macr&gt;</td>
<td>Optional: Eingabe von Macro-Befehlen, die im Batch-Modus automatisch abgearbeitet werden</td>
</tr>
<tr>
<td>↑</td>
<td>MACRO–Macros</td>
</tr>
<tr>
<td>↓</td>
<td>end</td>
</tr>
<tr>
<td>↑</td>
<td>Eingabewerte für MACRO–Macros in entsprechender Reihenfolge</td>
</tr>
<tr>
<td>↓</td>
<td>zugehörige Eingabedaten genaue Zeilenanzahl erforderlich!!</td>
</tr>
<tr>
<td>inte</td>
<td>Optional: interaktive Eingabe</td>
</tr>
<tr>
<td>stop</td>
<td>Ende Eingabe</td>
</tr>
</tbody>
</table>

2.5 Startoptionen für FEAP

2.5.1 Ausführung als Windowsprogramm

Ausführung rein interaktiv mit inte
oder
zunächst mit Macro-Befehlen macr und dann interaktiv mit inte

2.6 Anwendung im reinen Batchbetrieb

Start mit

```
feap -iIfile <-oofile -rRfile -sSfile -pPfile> -dIdev
```


Idev = 3 INTEL, Idev = 4 SALFORD.
2.7 Macros in detail

**FEAP**

```
feap [ title of problem for printouts, etc.]
numnp,numel,nummat,ndm,ndf,nen,nad,ndd
```

Each problem to be solved by FEAP must start with a single record which contains the characters `feap` as the first entry; the remainder of the record (columns 5–80) may be used to specify a problem title. The title will be printed with each “page” of output as the first line.

Immediately following the `feap` record, the `control` information describing characteristics of the problem to be solved must be given. The `control` information describes the characteristics of the finite element problem to be solved. The data entries are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numnp</td>
<td>total number of nodal points in the problem.</td>
</tr>
<tr>
<td>numel</td>
<td>total number of elements in the problem.</td>
</tr>
<tr>
<td>nummat</td>
<td>number of material property sets in the problem.</td>
</tr>
<tr>
<td>ndm</td>
<td>number of spatial coordinates needed to define mesh.</td>
</tr>
<tr>
<td>ndf</td>
<td>maximum number of degrees–of–freedom on any node.</td>
</tr>
<tr>
<td>nen</td>
<td>maximum number of nodes on any element.</td>
</tr>
<tr>
<td>nad</td>
<td>increases size of element arrays to ndf*nen + nad</td>
</tr>
<tr>
<td>ndd</td>
<td>maximum number of parameters for element properties</td>
</tr>
<tr>
<td></td>
<td>(default 50)</td>
</tr>
</tbody>
</table>

The number of spatial coordinates needed to define the finite element mesh (ndm) must be 1, 2, or 3. The maximum number of degrees–of–freedom on any node must be between 1 and 6. The maximum number of the other quantities is limited only by the size of the dynamically dimensioned array used to store all the data and solution parameters. This is generally quite large and, normally, should not be exceeded. If the error message that memory is exceeded appears the data should be checked to make sure that no errors exist which could cause large amounts of memory to solve the problem (e.g., if the error occurs when the `tang`, `utan`, or `cmas` solution macro statements are encountered, the profile of the matrix should be checked for very large column heights — appropriate renumbering of the mesh can often significantly reduce the storage required). If necessary, the main subprogram — ‘program feap’ can be recompiled with a larger dimension for the array m(*) in blank common.

For simplicity of input files it is allowed to start **FEAP** without defining the number of nodes (‘numnp’), the number of elements (‘numel’), and the number of materials (‘nummat’). In this case **FEAP** counts these values automatically. Results for ‘numnp’, ‘numel’ and ‘nummat’ are shown in the output file. For the correct calculation of ‘nummat’ it is necessary to to define each material with a separate mate-card. Otherwise ‘nummat’ is standard input.
Usage in FEAP - ISOGEO:
In FEAP - ISOGEO input files 'numnp' and 'numel' have to be given in all cases. 'numnp' is calculated by $n \times m$ for every patch and then summed up, 'numel' is determined by the sum of $(n - p) \times (m - q)$ of all patches. 'ndm' has to be set to 4 as control points are given in 4D-projective coordinates. For 2D cases the z-coordinate should be set to zero. 'nen' is the maximum number of control points having influence on one element. So all variables are chosen equivalent to the Finite Element Analysis counterparts.
ALOA

aloa
xmin,xmax,ymin,ymax,<zmin,zmax>
idof,etyp
qtyp,q1,q2,....
<etc., terminate with a blank record>

With aloan an external load \( q \) is set on all elements between \( \text{xmin}/\text{ymin},<\text{zmin}> \) \( \text{xmax}/\text{ymax}<\text{zmax} \). An element will be loaded for degree of freedom \( \text{idof} \), if the center of the element is within these bounds. \text{etyp} defines the element type. Elements should be defined isoparametrically.

<table>
<thead>
<tr>
<th>etyp</th>
<th>element</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4-node plate/shell element</td>
</tr>
<tr>
<td>9</td>
<td>9-node plate/shell element</td>
</tr>
<tr>
<td>8</td>
<td>8-node brick element, loaded on top ( (z = +\frac{\ell}{2}) )</td>
</tr>
<tr>
<td>-8</td>
<td>8-node brick element, loaded on bottom ( (z = -\frac{\ell}{2}) )</td>
</tr>
<tr>
<td>27</td>
<td>27-node brick element, loaded on top ( (z = +\frac{\ell}{2}) )</td>
</tr>
<tr>
<td>-27</td>
<td>27-node brick element, loaded on bottom ( (z = -\frac{\ell}{2}) )</td>
</tr>
</tbody>
</table>

\text{qtyp} defines the type of loading whereas \( q_1, q_2, .... \) are the associated load terms. Currently the following load types are implemented:

<table>
<thead>
<tr>
<th>qtyp</th>
<th>( q_1 )</th>
<th>( q_2 )</th>
<th>( q_3 )</th>
<th>( q_4 )</th>
<th>( q_5 )</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( q_0 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>constant load ( q_0 )</td>
</tr>
<tr>
<td>2</td>
<td>( q_0 )</td>
<td>( \ell_x )</td>
<td>( \ell_y )</td>
<td></td>
<td></td>
<td>cos-load ( q = q_0 \cdot \cos(\frac{\pi}{2\ell_x}x) \cdot \cos(\frac{\pi}{2\ell_y}y) )</td>
</tr>
</tbody>
</table>

Remarks

1. For the Cos–Load case the load \( q_0 \) is the maximum value at \( (0,0) \), \( \ell_x \) is the quadrant length of plate in x-direction and \( \ell_y \) the quadrant length of plate in y-direction.

2. Macro could not be used with more than one element types.

3. Brick elements only along with plates in x-y-plane.

4. aloan and reac lead to wrong results for associated dofs at boundaries.
The \texttt{angl} command is used to specify the angles for sloping nodal boundary conditions in two directions.

\begin{center}
\begin{tabular}{|l|}
\hline
\texttt{angl1} : first axis of rotated basis \hspace{1cm} (default=1) \\
\texttt{angl2} : second axis of rotated basis \hspace{1cm} (default=2) \\
\texttt{itrot} : 0/1 with/without rot. dofs 4-6 \hspace{1cm} (default=0) \\
\hline
\end{tabular}
\end{center}

For each node to be specified a record is entered with the following information:

\begin{center}
\begin{tabular}{|l|}
\hline
\texttt{node} : the number of the node to be specified. \\
\texttt{ngen} : the increment to the next node, if generation is used, otherwise 0. \\
\texttt{\alpha(node)} : value of angle in degrees that new 1-coord. makes with \texttt{x(1,node)}. \\
\hline
\end{tabular}
\end{center}

When generation is performed, the node number sequence will be \( (\text{node1}, \text{node1}+\text{ngen1}, \text{node1}+2*\text{ngen1}, \ldots, \text{node2}) \). The values for each angle generated will be a linear interpolation between \( \alpha(\text{node1}) \) and \( \alpha(\text{node2}) \). The degree–of–freedoms associated with the sloping boundary may differ from element to element as described in the 'ELMTnn' manuals.

The \texttt{angl} is positive from first axis to second axis which means counter clockwise.

\textbf{Remarks:}

1. Only one choice of directions \texttt{angl1}, \texttt{angl2} and \texttt{itrot} is possible.
2. A transformation will be done for dofs \texttt{angl1} and \texttt{angl2}.
3. In case of \texttt{ndf}=6 the same transformation is chosen for dofs \texttt{angl1+3} and \texttt{angl2+3} for \texttt{itrot}=0.
4. The associated dofs of the used element can be chosen via the macro \texttt{mate}, parameters idfg. As example the angles of the DKQ-plate element can be transformed via \texttt{mate} 1,7,2,3,1
5. Results are printed and plotted for nodes with \texttt{angl} \neq 0 in directions \texttt{angl1,angl2}. Thus mixed vectors occur.
6. Results which base on a smoothing procedure are plotted in global directions. Thus results for nodes with \texttt{angl} \neq 0 are transformed to global directions. Ex.: \texttt{disp, resi, eigv},...
The `back` command is used to specify a background mesh of triangular elements.

```
  nelm  - the number of the element to be specified.
  ix(1,nelm)  - node-1 number attached to element.
  ix(2,nelm)  - node-2 number attached to element.
  ix(3,nelm)  - node-3 number attached to element.

  node  - the number of the node to be specified.
  x(1,node)  - value of coordinate in 1-direction.
  x(2,node)  - value of coordinate in 2-direction.
  d(1,node)  - required element size.
  d(2,node)  - required stretching.
  d(3,node)  - 1-value of direction cosine of stretching direction.
  d(4,node)  - 2-value of direction cosine of stretching direction.
```

Remarks:

1. A background mesh with triangular elements will be defined to monitor the element density of the mesh. If `back` is used the parameter 'esiz' (used in `nege`) will not be used.

2. For mesh generation with NEGE the following order of statements is necessary: `nege`, `[neco]`, `[back]`, `geom`, `segm`, `regi`, `[pres]`, `[fixe]`, further FEAP-macros. Note: macros in brackets are optional.
Calculate the nodal base vectors at all nodes due to the chosen type 'n1'. For some fields the additional value 'x2-x4' are necessary. Actually the following base definitions are possible:

<table>
<thead>
<tr>
<th>n1</th>
<th>Free form for each global node</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Basis for each global node + averaging process, see element description</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n1</th>
<th>Free form for each element node</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Basis for each element node, see element description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Normal vector e3 for each element node, see element description, arbitrary vector t1 input, vectors e1 = t1/</td>
<td></td>
<td>t1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n1</th>
<th>Analytical solution for each global node</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Plate x-y plane e1 = e_x, e2 = e_y, e3 = e_z</td>
<td>y0</td>
<td>z0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Plate y-z plane e1 = e_y, e2 = e_z, e3 = e_x</td>
<td>x0</td>
<td>z0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Plate x-z plane e1 = e_x, e2 = e_z, e3 = -e_y</td>
<td>x0</td>
<td>y0</td>
<td>z0</td>
</tr>
<tr>
<td>6</td>
<td>Plate x-y plane polar</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Cylinder(x-axis) y = y0 + y, z = z0 + z</td>
<td>y0</td>
<td>z0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Cylinder(y-axis) x = x0 + x, z = z0 + z</td>
<td>x0</td>
<td>y0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Cylinder(z-axis) x = x0 + x, y = y0 + y, outside</td>
<td>x0</td>
<td>y0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Cylinder(z-axis) x = x0 + x, y = y0 + y, inside</td>
<td>x0</td>
<td>y0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Sphere x = x0 + x, y = y0 + y, z = z0 + z</td>
<td>x0</td>
<td>y0</td>
<td>z0</td>
</tr>
<tr>
<td>12</td>
<td>Hyperbola z = (4l_z/(l_x l_y)) xy</td>
<td>l_x</td>
<td>l_y</td>
<td>l_z</td>
</tr>
<tr>
<td>13</td>
<td>Rot. hyperboloid r = r0/c √c^2 + z^2</td>
<td>r0</td>
<td>c</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Rot. paraboloid r = r0 √(2z)</td>
<td>r0</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>15</td>
<td>Parabola z = a x^2</td>
<td></td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>16</td>
<td>Parabola z = a x^2</td>
<td></td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>17</td>
<td>Twisted beam</td>
<td></td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>

For these geometries the following base vectors are calculated:
<table>
<thead>
<tr>
<th>System</th>
<th>Base vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. plate x-y plane</td>
<td>$t_1 = 1, 0, 0$</td>
</tr>
<tr>
<td>4. plate y-z plane</td>
<td>$t_1 = 0, 1, 0$</td>
</tr>
<tr>
<td>5. plate x-z plane</td>
<td>$t_1 = 1, 0, 0$</td>
</tr>
<tr>
<td>6. cylinder (x-axis)</td>
<td>$t_1 = \cos \varphi, \sin \varphi, 0$</td>
</tr>
<tr>
<td>7. cylinder (y-axis)</td>
<td>$t_1 = \sin \varphi, 0, -\cos \varphi$</td>
</tr>
<tr>
<td>8. cylinder (z-axis)</td>
<td>$t_1 = -\sin \varphi, \cos \varphi, 0$</td>
</tr>
<tr>
<td>9. cylinder (z-axis) inside</td>
<td>$t_1 = \sin \varphi, -\cos \varphi, 0$</td>
</tr>
<tr>
<td>10. sphere (z-axis)</td>
<td>$t_1 = -\sin \varphi, \cos \varphi, 0$</td>
</tr>
<tr>
<td>11. hyper shell $z = axy$</td>
<td>$t_1 = 1, 0, ay/t$</td>
</tr>
<tr>
<td>12. rot. hyperboloid</td>
<td>$t_1 = -\sin \varphi, \cos \varphi, 0$</td>
</tr>
<tr>
<td>13. paraboloid</td>
<td>$t_1 = -\sin \varphi, \cos \varphi, 0$</td>
</tr>
<tr>
<td>14. paraboloid</td>
<td>$t_1 = -\sin \varphi, \cos \varphi, 0$</td>
</tr>
<tr>
<td>15. twisted beam</td>
<td>$t_1 = 1/t, 0, 2ax$</td>
</tr>
<tr>
<td>16. twisted beam</td>
<td>$t_1 = 1/t, -r \pi/a \sin \varphi, r \pi/a \cos \varphi$</td>
</tr>
</tbody>
</table>

The nodal base vectors can be plotted within the plot state under the macro base.
The **blico** command is used to generate a mesh for an interface zone and is very similar to the **bloc** macro.

The macro generates at first a 2D-mesh for quadrilaterals which are ‘at the bottom’ of the structure. In a second step a similar mesh is generated ‘at the top’ of the structure. These meshes have different nodes, the nodes themselves have the same coordinates. In between a third mesh of 8-node trilinear elements (mtyp=1, default) of 18-node biquadratic (Lagrange)/linear elements (mtyp=2) or of 16-node biquadratic (Serendipity)/linear elements (mtyp=3) of thickness h=0 is generated at the end.

![Diagram](image.png)

The patch of nodes/elements defined by **blico** is developed from a master element which is defined by an isoparametric 4–9 node mapping function in terms of the natural coordinates \( r \) and \( s \). The node numbers on the master element of each patch defined by **blico** are specified according to the following figure. The four corner nodes of the master element must be specified, the mid–point and central nodes are optional.

![Diagram](image2.png)

The spacing between the \( r \)–increments and \( s \)–increments may be varied by a proper specification of the mid–side and central nodes. Thus, it is possible to ‘concentrate’ nodes/elements into one corner of the patch generated by **blico**. The mid–nodes must lie within the central–half of the \( r \)– or \( s \)–directions to keep the isoparametric mapping single valued for all \((r, s)\) points.
The data parameters are defined as:

| nodes     | number of master nodes needed to define the patch. |
| r-inc     | number of nodal increments to be generated along r-direction of the patch. |
| s-inc     | number of nodal increments to be generated along s-direction of the patch. |
| node1     | number to be assigned to first generated node in patch |
| elmt1     | number to be assigned to first element generated in patch |
| mat1      | material number to be assigned to generated quadrilaterals at the bottom |
| mat2      | material number to be assigned to generated quadrilaterals at the top |
| mat3      | material number to be assigned to generated 8 node interface elements |
| mtyp      | type of interface elements: mtyp=1: 8-node, mtyp=2: 18-node, mtyp=3: 16-node |

r-inc, s-inc must be even numbers for mtyp=2,3!
**BLOC**

The **bloc** command acts on 2–D meshes for \( btype = 0...9,16,200...210 \), see A and on 3–D meshes for \( btype = 10...15,19,300...310 \), see B.

**A: 2–D mesh generation**

The **bloc** data input segment is used to generate a regular one or two dimensional patch of nodes. Alternatively, nodes together with 4–node quadrilateral elements may be generated for two dimensional patches or three dimensional surfaces.

The patch of nodes/elements defined by **bloc** is developed from a master element which is defined by an isoparametric 4–9 node mapping function in terms of the natural coordinates \( r \) and \( s \). The node numbers on the master element of each patch defined by **bloc** are specified according to the following figure. The four corner nodes of the master element must be specified, the mid–point and central nodes are optional.

```
  4  |  7  |
  1  |  2  |
  8  |  9  |
  5  |  6  |
```

The spacing between the \( r \)–increments and \( s \)–increments may be varied by a proper specification of the mid–side and central nodes. Thus, it is possible to `concentrate' nodes/elements into one corner of the patch generated by **bloc**. The mid–nodes must lie within the central–half of the \( r \)– or \( s \)–directions to keep the isoparametric mapping single valued for all \((r, s)\) points.

```
  4  |  10 |  16 |
  11 |  13 |  12 |
  15 |  14 |  17 |
```

Patches may be interconnected, in a restricted manner, by using the “\( r \)–skip” parameter judiciously. In addition, the **tie** macro may be used to `connect' any nodes which have the same coordinates.
The data parameters are defined as:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodes</td>
<td>number of master nodes needed to define the patch.</td>
</tr>
<tr>
<td>r-inc</td>
<td>number of nodal increments to be generated along r-direction of the patch.</td>
</tr>
<tr>
<td>s-inc</td>
<td>number of nodal increments to be generated along s-direction of the patch.</td>
</tr>
<tr>
<td>node1</td>
<td>number to be assigned to first generated node in patch (default = 1). First node is located at same location as master node 1. Last generated node (i.e., node1 - 1 + (r-inc * s-inc)) is located at same location as master node 3.</td>
</tr>
<tr>
<td>elmt1</td>
<td>number to be assigned to first element generated in patch; if zero no elements are generated (default = 0)</td>
</tr>
<tr>
<td>matl</td>
<td>material number to be assigned to all generated elements in patch (default = 1)</td>
</tr>
<tr>
<td>r-skip</td>
<td>number of nodes to skip between end of an r-line and start of next r-line (may be used to interconnect blocks side-by-side) (default = 1)</td>
</tr>
<tr>
<td>b-type</td>
<td>nodes and 4-node elements on patch; sequentially along consecutive r-lines</td>
</tr>
<tr>
<td></td>
<td>= 0</td>
</tr>
<tr>
<td></td>
<td>= 1</td>
</tr>
<tr>
<td></td>
<td>= 2</td>
</tr>
<tr>
<td></td>
<td>= 3</td>
</tr>
<tr>
<td></td>
<td>= 4</td>
</tr>
<tr>
<td></td>
<td>= 5</td>
</tr>
<tr>
<td></td>
<td>= 6</td>
</tr>
<tr>
<td></td>
<td>= 7</td>
</tr>
<tr>
<td></td>
<td>= 8</td>
</tr>
<tr>
<td></td>
<td>= 9</td>
</tr>
<tr>
<td></td>
<td>=16</td>
</tr>
<tr>
<td></td>
<td>=2xx</td>
</tr>
</tbody>
</table>
The **bloc** data input segment is used here to generate a regular 3 dimensional patch of nodes for 8/20/27/64/.../1331–node 3–D brick elements and 4–node tetrahedron elements. The 20 node brick element can be used with nen=20 or nen=21. In the latter case an additional unused node is generated.

The patch of nodes/elements defined by **bloc** is developed from a master element which is defined by an isoparametric 8/27 node mapping function in terms of the natural coordinates \( r, s \) and \( t \). The node numbers on the master element of each patch defined by **bloc** are specified according to the following figure.

The eight corner nodes of the master element at \( t = \pm 1 \) must be specified, the mid-point nodes and the central nodes are optional.

The spacing between the \( r, s, t \)–increments may be varied by a proper specification of the mid-side and central nodes. Thus, it is possible to ‘concentrate’ nodes/elements into one corner of the patch generated by **bloc**. The mid-nodes must lie within the central-half of the \( r, s, t \)–directions to keep the isoparametric mapping single valued for all \( (r, s, t) \) points.

Patches may be interconnected by the **tie** command to connect any nodes which have the same coordinates.

**Bug:** combination of nodes 9-12/13-16/18-21 with 23-26 leads to errors.

**Bug:** Hide does not work for tetrahedron.
**b-type = 10,12,13,14,15,19**

This option generates 8-/20-/21-/27- and 64- node brick elements.

The data parameters are defined as:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodes</td>
<td>number of master nodes needed to define the patch.</td>
</tr>
<tr>
<td>r-inc</td>
<td>number of nodal increments to be generated along r-direction of the patch.</td>
</tr>
<tr>
<td>s-inc</td>
<td>number of nodal increments to be generated along s-direction of the patch.</td>
</tr>
<tr>
<td>t-inc</td>
<td>number of nodal increments to be generated along t-direction of the patch.</td>
</tr>
<tr>
<td>node1</td>
<td>number to be assigned to first generated node in patch (default = 1). First node is located at same location as master node 1. Last generated node (i.e., node1 - 1 + (r-inc * s-inc * t-inc)) is located at same location as master node 3.</td>
</tr>
<tr>
<td>elmt1</td>
<td>number to be assigned to first element generated in patch; if zero no elements are generated (default = 0)</td>
</tr>
<tr>
<td>matl</td>
<td>material number to be assigned to all generated elements in patch (default = 1)</td>
</tr>
</tbody>
</table>

**b-type = 10** – 8-node brick elements

= 12 – 20-node brick elements, nen=21 (Interior node 17 generated but not used!), rinc, sinc and tinc must be even numbers

= 13 – 27-node brick elements, rinc, sinc and tinc must be even numbers

= 14 – 20-node brick elements, nen=20 (Interior node 17 generated but not used, continuous node numbering), rinc, sinc and tinc must be even numbers

= 15 – 18-node brick elements (biquadratic/linear, continuous node numbering, rinc, sinc must be even numbers, tinc=2

= 19 – 64-node brick elements, rinc, sinc and tinc must be multiples of three

=3xx – Arbitrary number of nodes. 3 in 3xx specifies a 3 dimensional mesh of bricks. The following number xx specifies the shape function order o=xx per direction. 301 = linear elements ... 310 = shapes of order 10, which is currently the maximum order. r-inc, s-inc, t-inc must be multiple of o.
The positions of the local nodes of the generated elements are illustrated in the following figures.
Institut für Baustatik, KIT
Cont. Mesh Macro Plot Mate3D Elem Nege Gmesh Cylt Isogeo
This option generates 4-node tetrahedron elements by subdividing one brick element into 6 tetrahedrons. Thus one has to specify six-times the number of elements \((r\text{-inc} \times s\text{-inc} \times t\text{-inc} \times 6)\). The data parameters are defined as:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodes</td>
<td>number of master nodes needed to define the patch.</td>
</tr>
<tr>
<td>r-inc</td>
<td>number of nodal increments to be generated along (r)-direction of the patch.</td>
</tr>
<tr>
<td>s-inc</td>
<td>number of nodal increments to be generated along (s)-direction of the patch.</td>
</tr>
<tr>
<td>t-inc</td>
<td>number of nodal increments to be generated along (t)-direction of the patch.</td>
</tr>
<tr>
<td>node1</td>
<td>number to be assigned to first generated node in patch (default = 1). First node is located at same location as master node 1. Last generated node (i.e., (node1 - 1 + (r\text{-inc} \times s\text{-inc} \times t\text{-inc}))) is located at same location as master node 3.</td>
</tr>
<tr>
<td>elmt1</td>
<td>number to be assigned to first element generated in patch; if zero no elements are generated (default = 0)</td>
</tr>
<tr>
<td>matl</td>
<td>material number to be assigned to all generated elements in patch (default = 1)</td>
</tr>
<tr>
<td>r-skip</td>
<td>number of nodes to skip between end of an (r)-line and start of next (r)-line (may be used to interconnect blocks side-by-side) (default = 1)</td>
</tr>
<tr>
<td>b-type</td>
<td>type of patch to be generated = 11</td>
</tr>
</tbody>
</table>
Additional informations about the bloc numbers 2xx and 3xx:

- **Node numbering**

  Node numbering is similar to 3D-64 nodes for the 2D- and 3D-case and is stated in general in the next figures.

  ![Node numbering diagram](image)

  The diagrams illustrate the node numbering for bloc 201 and bloc 301, showing how nodes are labeled with respect to the positions indicated by the equations for the nodes.
**BLOX**

**blox**

nodes, r-inc, s-inc, node1, [elmt1-, mat, r-skip, b-type]

1, x1, y1, z1 (only ndm coordinates required)

2, x2, y2, z2

etc., until 4
dx1, dx2, dx3, . . . r-times (16 per line!)
dy1, dy2, dy3, . . . s-times

mx1, mx2, mx3, . . . r-times

my1, my2, my3, . . . s-times

The **blox** data input segment is used to generate a nonregular 2D patch of nodes on a rectangular(!) mesh similar to **bloc**.

The patch of nodes/elements defined by **blox** is developed from a master element which is defined by an 4 node mapping function. The node numbers on the master element of each patch defined as in **bloc**. The data parameters are defined as:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodes</td>
<td>number of master nodes needed to define the patch.</td>
</tr>
<tr>
<td>r-inc</td>
<td>number of nodal increments to be generated along r-direction.</td>
</tr>
<tr>
<td>s-inc</td>
<td>number of nodal increments to be generated along s-direction.</td>
</tr>
<tr>
<td>node1</td>
<td>number to be assigned to first generated node.</td>
</tr>
<tr>
<td>elmt1</td>
<td>number to be assigned to first element generated in patch.</td>
</tr>
<tr>
<td>matl</td>
<td>dummy input</td>
</tr>
<tr>
<td>r-skip</td>
<td>number of nodes to skip between end of an r-line and start of next r-line (default = 1)</td>
</tr>
<tr>
<td>b-type</td>
<td>like <strong>bloc</strong></td>
</tr>
<tr>
<td>dx(i=1,r-inc)</td>
<td>increments in x-direction</td>
</tr>
<tr>
<td>dy(i=1,s-inc)</td>
<td>increments in y-direction</td>
</tr>
<tr>
<td>mx(i=1,r-inc)</td>
<td>material number for element dx</td>
</tr>
<tr>
<td>my(i=1,s-inc)</td>
<td>material number for element dy</td>
</tr>
</tbody>
</table>

The maximum of mx, my will be used for element dx, dy.
The `boun` command is used to specify the values for the boundary restraint conditions. For each node to be specified a record is entered with the following information:

<table>
<thead>
<tr>
<th>node</th>
<th>the number of the node to be specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>ngen</td>
<td>the increment to the next node, if generation is used, otherwise 0.</td>
</tr>
<tr>
<td>id(1,node)</td>
<td>value of boundary restraint for 1–dof for 'node'</td>
</tr>
<tr>
<td>id(2,node)</td>
<td>value of boundary restraint for 2–dof for 'node'.</td>
</tr>
<tr>
<td>etc. until to 'ndf' directions</td>
<td></td>
</tr>
</tbody>
</table>

The boundary restraint codes are interpreted as follows:

- $\text{id}(i,\text{node}) = 0$ i–dof has unknown displacement, $f(i,\text{node})$ will be an applied load.
- $\text{id}(i,\text{node}) \neq 0$ i–dof has known displacement, $f(i,\text{node})$ is the displacement value (default = 0.).

When generation is performed, the node number sequence will be (for node1–node2 sequence shown above):

- node1, node1+ngen1, node1+2*ngen1, ..., node2

The values for each boundary restraint will be as follows:

- id(i,node1) = 0 or positive $\rightarrow$ id(i,node1+ngen1) = 0
- id(i,node1) = negative $\rightarrow$ id(i,node1+ngen1) = -1

With this convention the value of '$\text{id}(i,node2)$' will be negative whenever the value of $\text{id}(i,node1)$ starts negative. Accordingly, it is necessary to assign a positive value for the restraint code to terminate a generation sequence (e.g., when it is no longer desired to set a dof to be restrained). Alternatively, an 'i–dof' may be eliminated for all nodes by using the generation sequence:

<table>
<thead>
<tr>
<th>Degree of Freedom</th>
</tr>
</thead>
<tbody>
<tr>
<td>node 1 ... i ... ndf</td>
</tr>
<tr>
<td>numnp 0 ... -1, +1, ...</td>
</tr>
</tbody>
</table>

Subsequent records may then be used to assign the values to the other degree–of–freedoms.

**Remarks:**

1. Up to 16 values are possible on each input record.
2. `boun` sets the boundary conditions whereas `ebou` adds boundary conditions to existing values.
3. Prescribed displacements are defined with the macro command `disp`.
4. Loads on bounded DOFS, introduced via the macro command `load`, are interpreted as prescribed displacements!!
The **btem** data input segment is used to generate temperatures for a regular two or three dimensional patch of nodes.

The patch of nodes/elements defined by **btem** is developed from a master element which is defined by an isoparametric 4–9 node mapping function in terms of the natural coordinates $r$, $s$ and $t$. The node numbers on the master element of each patch defined by **btem** are specified according to Fig. 24.25 on page 748 of *THE FINITE ELEMENT METHOD, 3RD. ED.*, by O. C. Zienkiewicz. The corner nodes of the master element must be specified, the mid–point and central node are optional. The node numbers on the master element of each patch are specified according to the following setup:

### 2-D-generation

<table>
<thead>
<tr>
<th>Coordinate</th>
<th>corner nodes</th>
<th>mid nodes</th>
<th>center node</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = 0$</td>
<td>1 2 3 4</td>
<td>5 6 7 8</td>
<td>9</td>
</tr>
</tbody>
</table>

### 3-D-generation

<table>
<thead>
<tr>
<th>Coordinate</th>
<th>corner nodes</th>
<th>mid nodes</th>
<th>center node</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = -1$</td>
<td>1 2 3 4</td>
<td>13 14 15 16</td>
<td>17</td>
</tr>
<tr>
<td>$t = 0$</td>
<td>9 10 11 12</td>
<td>23 24 25 26</td>
<td>27</td>
</tr>
<tr>
<td>$t = +1$</td>
<td>5 6 7 8</td>
<td>18 19 20 21</td>
<td>22</td>
</tr>
</tbody>
</table>

The spacing between the $r$, $s$ and $t$–increments may be varied by a proper specification of the mid–side and central nodes. Thus, it is possible to ‘concentrate’ nodes/elements into one corner of the patch generated by **btem**. The mid–nodes must lie within the central–half of the $r$, $s$ or $t$–directions to keep the isoparametric mapping single valued for all $(r, s, t)$ points.

Patches may be interconnected, in a restricted manner, by using the “$r$–skip” parameter judiciously. In addition, the **tie** macro may be used to ‘connect’ any nodes which have the same coordinates. The data parameters are defined as:
<table>
<thead>
<tr>
<th>nodes</th>
<th>number of master nodes needed to define the patch.</th>
</tr>
</thead>
<tbody>
<tr>
<td>r-inc</td>
<td>number of nodal increments to be generated along r-direction of the patch.</td>
</tr>
<tr>
<td>s-inc</td>
<td>number of nodal increments to be generated along s-direction of the patch.</td>
</tr>
<tr>
<td>t-inc</td>
<td>number of nodal increments to be generated along s-direction of the patch (default = 1).</td>
</tr>
<tr>
<td>node1</td>
<td>number to be assigned to first generated node in patch (default = 1). First node is located at same location as master node 1. Last generated node (i.e., node1 - 1 + (r-inc * s-inc)) is located at same location as master node 3.</td>
</tr>
<tr>
<td>r-skip</td>
<td>number of nodes to skip between end of an r-line and start of next r-line (may be used to interconnect blocks side-by-side) (default = 1)</td>
</tr>
<tr>
<td>tempi</td>
<td>temperature at node i</td>
</tr>
</tbody>
</table>
**CMOD**

**cmd**
```cmod```
```type,node1,node2,parameters```
```
<terminate with blank record>
```

The **cmod** command may be used to modify coordinates which have been defined before by **coor**, **bloc** etc. Coordinates from node1 to node2 are treated.

Up to now the following types are possible

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter</th>
<th>Result</th>
<th>Formula</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>r</td>
<td>sphere</td>
<td>( z = \sqrt{r^2 - x^2 - y^2} )</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>r,f</td>
<td>parabola</td>
<td>( z = -\frac{f}{r^2} (x^2 + y^2) + f )</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>hypar</td>
<td>( z = a xy )</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>a,c</td>
<td>hyperboloid</td>
<td>( r = \frac{a}{c} \sqrt{c^2 + z^2} )</td>
<td>polar coor.</td>
</tr>
<tr>
<td>5</td>
<td>r,a</td>
<td>hyperboloid</td>
<td>( z = r(\sqrt{1 + x^2/a^2} - 1) )</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>r,a</td>
<td>sphere</td>
<td>( z = z + \sqrt{r * r - y * y - a} )</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>a</td>
<td>twisted beam</td>
<td>( y = y \cos \varphi, z = y \sin \varphi, \varphi = \pi/(2a) ) ( x )</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>r,m</td>
<td>sphere-3 blocs</td>
<td>correct ( x,y,z ) to real spherical coordinates</td>
<td>use GMESH for meshing \n3 blocs with ( m \times m ) elements</td>
</tr>
</tbody>
</table>

Be careful with the use of this macro. Due to the modification of the coordinates macros like **eloa**, **edge** or **tie** may not work.
COOR

The `coor` command is used to specify the values for nodal coordinates. For each node to be specified a record is entered with the following information:

<table>
<thead>
<tr>
<th>node</th>
<th>the number of the node to be specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>ngen</td>
<td>the increment to the next node, if generation is used, otherwise 0.</td>
</tr>
<tr>
<td>x(1,node)</td>
<td>value of coordinate in 1–direction for ’node’</td>
</tr>
<tr>
<td>x(2,node)</td>
<td>value of coordinate in 2–direction for ’node’, etc. to ’ndm’ directions</td>
</tr>
</tbody>
</table>

When generation is performed, the node number sequence will be (for node1–node2 sequence shown above):

node1, node1+ngen1, node1+2*ngen1, .... , node2

The values for each coordinate will be a linear interpolation between x(node1) and x(node2).

Input of node numbers is not sequential, gaps in numbering are possible.

Usage in FEAP - ISOGEO:

The `coor` command works as in standard feap. Control points B have to be given in unweighted 4D-projective coordinates (x, y, z, w). Generation may not be used, gaps in numbering are not allowed. Input of node numbers is advised to be sequential. The enumeration has to follow a fixed scheme as the sequence of the control points defines the form of the mesh. There is no further possibility to build kind of an ’element-node’-relation. This is fixed in the ordering of the control points. The generation of NURBS is usually done with a 3D-modelling software. Export into an IGES-textfile gives the correct order of control points. These IGES-files have to be converted by a preprocessor to fit the FEAP - ISOGEO input convention. `coor` has to stand before macro `nmpq`. Not more than 10 digits for the base may be used, the total length of one line may not exceed 80 characters. The exponent can be started with e or d.

Enumeration scheme:
```plaintext
int=0
for patch=1:n_patch
  for j=1:m(patch)
    for i=1:n(patch)
      int = int + 1
      CP(int)=CP(i,j,patch)
    end
  end
end
```
Description of a curve.

To use the **curve**-option the complete mesh must first be defined. **curv** must be inserted before the **end** command for the mesh definition.

- **n1** the unique curve number (may be any integer number)
  (the maximum number of curves is 80)
- **n2** type of the desired curve (see below)
- **n3** parameter to describe the intention of the curve:
  e.g. mesh refinement and mesh smoothing
  - **n3=15** curved surface elements
  - **n3=16** curved boundary function for plane surfaces
  - **n3=17** curved boundary function for curved surfaces
  - **n3=18** linear boundary function for curved surfaces
  - **n3=20+ndf** functions, which define boundary conditions
  - **n3=30+ndf** functions, which define load conditions
- **p1** to **p8** - up to 8 parameters needed to specify the curve (see below)

The possible values for **n2** are:

- **1** to **9** for 2-dimensions
- **10** to **49** for 3-dimensions
- **50** to **99** user defined curves (**SR cvuser**)

- **n2 = 1** Linear function defined by two points
  - p1 = x1, p2 = y1, p3 = x2, p4 = y2
- **n2 = 2** Linear function defined by one point and the slope
  - p1 = x1, p2 = y1, p3 = slope
- **n2 = 3** half circle defined by midpoint (x1/y1) and radius
  - p1 = radius
  - p2 = +1 or -1 for upper or lower part of the circle
  - p3 = x1, p4 = y1
- **n2 = 30** Linear function defined by two points
  - p1 = x1, p2 = y1, p3 = z1,
  - p4 = x2, p5 = y2, p6 = z2.
- **n2 = 31** sphere with radius r and mid-point x1,y1,z1
  - p1 = radius, p2 = x1, p3 = y1, p4 = z1
- **n2 = 32** plane function defined by \((x_0 \cdot x + y_0 \cdot y + z_0 \cdot z + c_0) = 0\)
  - p1 = x0, p2 = y0, p3 = z0, p4 = c0
Each problem to be solved by FEAP using the yield-line mesh generator CYLT must start with a single record which contains the characters cylt as the first entry. The remainder of the record (columns 5-80) may be used to specify a problem title. Immediately following the cylt record, the ‘control’ information describing characteristics of the problem to be solved must be given. The ‘control’ information describes the characteristics of the problem to be solved. The data entries are:

\[
\begin{array}{l}
typ \quad 0 = \text{yield-line mesh} \\
\quad 1 = \text{1st step of triangulation} \\
\quad 2 = \text{2nd step of triangulation} \\
\quad \text{etc.}
\end{array}
\]

Remarks:

1. It is recommended to test the data input with ‘typ’ = 0. The achieved yield-line mesh will not be able to produce calculation results (therefore use ‘typ’ = 1). Depending on the geometry it is possible that several yield-line mesh solutions are found. The user has to choose the considered mesh number after the generation. The number does not depend on the most likely solution; it may be possible that different numbers content the same mesh, see CYLT-manual.

2. ‘typ’ > 1 yields higher triangulated meshes which will not be useful for a yield-line calculation performed by FEAP. The possibility of higher triangulation is optional to other programs.

3. A FEAP-input file will be generated called ifeap.tmp (always).

4. For mesh generation with CYLT the following order of statements is necessary: cylt, [ycon], ynod, yedg, further FEAP-macros (including ybou).
   Note: macros in brackets are optional.

5. For further information see the CYLT-manual.
The `debu` macro command can be used to debug the program code.

- `n1 = 0` - set the debug option to false (default value)
- `n1 = 1` - set the debug option to true
The **disp** command is used to specify the values for nodal prescribed displacements (forced conditions). For each node to be specified a record is entered with the following information:

<table>
<thead>
<tr>
<th>node</th>
<th>the number of the node to be specified.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ngen</td>
<td>the increment to the next node, if generation is used, otherwise 0.</td>
</tr>
<tr>
<td>d(1,node)</td>
<td>value of displ for 1–dof for 'node'</td>
</tr>
<tr>
<td>d(2,node)</td>
<td>value of displ for 2–dof for 'node'</td>
</tr>
<tr>
<td>etc., to 'ndf' directions</td>
<td></td>
</tr>
</tbody>
</table>

When generation is performed, the node number sequence will be (for node1–node2 sequence shown above):

    node1, node1+ngen1, node1+2*ngen1, ..., node2

The values for each displacement will be a linear interpolation between d(node1) and d(node2). Up to 16 values are possible on each input record.

A displacement is prescribed when together with the **disp** command also a boundary condition is set for the same degree of freedom with **boun** or **ebou**.
EANG

eang
angl1, angl2, itrot
i–dir, xi–value, angle
<etc., terminate with a blank record>

eang
define a local base system along a line. The angles may be set along any set of nodes which has a constant value of the 'i–direction' coordinate (e.g., 1–direction (or x), 2–direction (or y), etc.). The following data are necessary:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>angl1</td>
<td>first axis of rotated basis (default=1)</td>
</tr>
<tr>
<td>angl2</td>
<td>second axis of rotated basis (default=2)</td>
</tr>
<tr>
<td>itrot</td>
<td>0/1 with/without rot. dofs 4-6 (default=0)</td>
</tr>
<tr>
<td>i–idir</td>
<td>the direction of the coordinate (i.e., 1 = x, 2 = y, etc. to ndm) to be searched.</td>
</tr>
<tr>
<td>xi–value</td>
<td>the value of the i–direction coordinate to be used during the search (a tolerance of about 1/1000 of the mesh size is used during the search, any coordinate within the tolerance is assumed to have the specified value).</td>
</tr>
<tr>
<td>angle</td>
<td>value of angle new 1-coordinate makes with x(1,node).</td>
</tr>
</tbody>
</table>

Remarks:

1. Only one choice of directions angl1, angl2 and itrot is possible.
2. A transformation will be done for dofs angl1 and angl2.
3. In case of ndf=6 the same transformation is chosen for dofs angl1+3 and angl2+3 for itrot=0.
4. The associated dofs of the used element can be chosen via the macro mate, parameters idfg. As example the angles of the DKQ-plate element can be transformed via mate
   1,7,2,3,1
5. Results are printed and plotted for nodes with angl ≠ 0 in directions angl1,angl2. Thus mixed vectors occur.
6. Results which base on a smoothing procedure are plotted in global directions. Thus results for nodes with angl ≠ 0 are transformed to global directions. Ex.: disp, resi, eigv,...
**EBOU**

```
ebou,<gap>
<vgap, only in case of ebou,gap>
i–dir,xi–value,(ibc(j),j=1,ndf)
<etc., terminate with a blank record>
```

The boundary restraint conditions may be set along any set of nodes which has a constant value of the 'i–direction' coordinate (e.g., 1–direction (or x), 2–direction (or y), etc.). The data to be supplied during the definition of the mesh (or in macro execution using the mesh command) consists of:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;vgap&gt;</td>
<td>Input only in case of ebou,gap: user defined absolute tolerance for search.</td>
</tr>
<tr>
<td>i–idir</td>
<td>the direction of the coordinate (i.e., 1 = x, 2 = y, etc. to ndm) to be searched.</td>
</tr>
<tr>
<td>xi–value</td>
<td>the value of the i–direction coordinate to be used during the search (a tolerance of about 1/1000·√numnp of the mesh size is used during the search, any coordinate within the tolerance is assumed to have the specified value). In case of problems use ebou,gap</td>
</tr>
<tr>
<td>ibc(1)</td>
<td>restraint conditions for all nodes with the value of the search (0 = active dof).</td>
</tr>
<tr>
<td>ibc(2)</td>
<td>restraint conditions for all nodes with the value of the search (0 = active dof).</td>
</tr>
<tr>
<td>...</td>
<td>restraint conditions for all nodes with the value of the search (0 = active dof).</td>
</tr>
<tr>
<td>ibc(ndf)</td>
<td>&gt; 0 or &lt; 0 denotes a fixed dof).</td>
</tr>
</tbody>
</table>

Remarks:

1. Up to 16 values and up to 80 characters are possible on each input record.
2. **boun** sets the boundary conditions whereas **ebou** adds boundary conditions to existing values.
3. For the specification of prescribed displacements within a calculation, see macro commands **edis** or **disp**.
**EDGE**

```plaintext
edge
a_1, a_2, (a_3), b_1, b_2, (b_3), c_1, c_2, (c_3), tol, add, rad
(IBC(j), j=1, ndf)
<etc., terminate with a blank record>
```

- The boundary restraint conditions may be set on a line between points on a straight line, on a parabola or on a circle.
- The straight line is defined by \( P_a(a_1, a_2, (a_3)) \), \( P_b(b_1, b_2, (b_3)) \) and \( P_c(0, 0, (0)) \).
- The parabola is defined by \( P_a(a_1, a_2, (a_3)) \) (begin), \( P_b(b_1, b_2, (b_3)) \) (end) and the third point \( P_c(c_1, c_2, (c_3)) \).
- The circle is defined by \( P_a(a_1, a_2, (a_3)) \) (begin), \( P_b(b_1, b_2, (b_3)) \) (end), the center point \( P_c(c_1, c_2, (c_3)) \) and a nonzero value 'rad' (radius of circle). Circles are only possible in the 1-2-plane. Thus differences in 3-direction are ignored. Angles to the points \( P_a \) and \( P_b \) are measured from the 1-axis counterclockwise. Angles are allowed only in the range: \( 0^\circ \leq \varphi \leq 360^\circ \). Thus a circle between for example \( 300^\circ \) and \( 450^\circ \) has to be described by two circles: \( 300^\circ \leq \varphi \leq 360^\circ \) and \( 0^\circ \leq \varphi \leq 90^\circ \).
- **edge** adds the boundary conditions for add = 0 and set the boundary conditions for add = 1.
- 'Tol' is a value to define the convergence radius around the line. With the default value 'tol' = 1000 usually the associated nodes are found. In special situations (fine discretization) a higher value may be necessary to find only the nodes exact on the given line.
- Proof if all nodes are correct found with `boun` in the plot mode.
- 'Rad' is the radius of the circle. Input for other types 'Rad' = 0.
- The boundary data are defined by:

```
  ibc(1)
  ibc(2)  restraint conditions for all nodes with the
  ...  value of the search (0 = active dof).
  ibc(ndf) > 0 or < 0 denotes a fixed dof).
```

**Remarks:**

1. Up to 16 values and up to 80 characters are possible on each input record.
EDIS

edis
i–dir,xi–value,(dbc(j),j=1,ndf)
<etc., terminate with a blank record>

Prescribed values for boundary restraint conditions may be set along any set of nodes which has a constant value of the 'i–direction' coordinate (e.g., 1–direction (or x), 2–direction (or y), etc.). The data to be supplied during the definition of the mesh (or in macro execution using the mesh command) consists of:

<table>
<thead>
<tr>
<th>i–idir</th>
<th>the direction of the coordinate (i.e., 1 = x, 2 = y, etc. to ndm) to be searched.</th>
</tr>
</thead>
<tbody>
<tr>
<td>xi–value</td>
<td>the value of the i–direction coordinate to be used during the search (a tolerance of about 1/1000 of the mesh size is used during the search, any coordinate within the tolerance is assumed to have the specified value).</td>
</tr>
<tr>
<td>dbc(1)</td>
<td>prescribed values for restraint conditions for all nodes</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>dbc(ndf)</td>
<td></td>
</tr>
</tbody>
</table>

Remarks:

1. Up to 16 values and up to 80 characters are possible. Only one input record could be used at the moment, thus the constraint is ndf ≤ 14.

2. The boundary conditions have to be set with appropriate macro commands, typically ebou. Otherwise these values are interpreted as loads.

3. Control of input is possible, if ebou is inactive. Then values of edis could be seen as loads.
The **elem** command is used to specify the values of nodal numbers which are attached to each element. For each element to be specified, a record is entered with the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nelm</td>
<td>the number of the element to be specified.</td>
</tr>
<tr>
<td>matl</td>
<td>the material number for the element, this will determine the element type.</td>
</tr>
<tr>
<td>ix(i,nelm)</td>
<td>node–1 number attached to element.</td>
</tr>
<tr>
<td>ix(2,nelm)</td>
<td>node–2 number attached to element.</td>
</tr>
<tr>
<td>etc., to 'nen' nodes.</td>
<td></td>
</tr>
<tr>
<td>ngen</td>
<td>value to increment each node–i value when generation is used (default = 1).</td>
</tr>
</tbody>
</table>

When generation is performed, the element number sequence will be in increments of 1 from ‘nelm1’ to ‘nelm2’; the nodes which are generated for each intermediate element will be as follows:

\[
ix(i,\text{nelm}+1) = ix(i,\text{nelm}) + \text{ngen1}
\]

except

\[
x(i,\text{nelm}+1) = 0 \text{ whenever } ix(i,\text{nelm}) = 0
\]

The program will assume that any zero value of an ix(i,elm) indicates that no node is attached at that point. This option is especially useful when isoparametric elements with midside nodes form a transition to elements without midside nodes. In this case it is often necessary to specify an element whose sides have mixed order. For example:

```
4 7 3
 o------o----o  has: ix = 1,2,3,4,0,0,7,8,9
 |     |   where the 0’s denote the unused nodes.
8 o 9 o | 1
 |    |
```

In case ‘numel’ is defined in the input it is not necessary to describe the last element number. A blank record will cause the program to generate elements up to ‘numel’ using the node increment specified on the last input record.

Input of element numbers is sequential! Free input of element numbers is possible with macro **elfr**.
The `elfr` command is used to specify the values of nodal numbers which are attached to each element. For each element to be specified, a record is entered with the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>nelm</code></td>
<td>the number of the element to be specified.</td>
</tr>
<tr>
<td><code>matl</code></td>
<td>the material number for the element, this will determine the element type.</td>
</tr>
<tr>
<td><code>ix(1,nelm)</code></td>
<td>node–1 number attached to element.</td>
</tr>
<tr>
<td><code>ix(2,nelm)</code></td>
<td>node–2 number attached to element.</td>
</tr>
<tr>
<td>etc., to <code>nen</code></td>
<td>etc., to 'nen' nodes.</td>
</tr>
</tbody>
</table>

No generation is possible. In this case use macro `elem`. Input of element numbers is not sequential, gaps in numbering are possible.
The `el3b` command is identical to the `elem` command. The only difference is that the node-generation does not act on the third node which is used for 3-D beams to specify the orientation of local axis.
Loads may be set on a line between points. Here, a straight line, a parabola or a circle are possible up to now.

- The straight line is defined by \( P_a(a_1, a_2, (a_3)) \) and \( P_b(b_1, b_2, (b_3)) \) and \( P_c(0, 0, (0)) \).
- The parabola is defined by \( P_a(a_1, a_2, (a_3)) \) (begin), \( P_b(b_1, b_2, (b_3)) \)(end) and the third point \( P_c(c_1, c_2, (c_3)) \).
- The circle is defined by \( P_a(a_1, a_2, (a_3)) \) (begin), \( P_b(b_1, b_2, (b_3)) \)(end), the center point \( P_c(c_1, c_2, (c_3)) \) and a nonzero value 'rad' (radius of circle). Circles are only possible for x-y-systems (2D). Angles to the points \( P_a \) and \( P_b \) are measured from the x-axis counterclockwise. Angles are allowed only in the range: \( 0^\circ \leq \varphi \leq 360^\circ \). Thus a circle between for example \( 300^\circ \) and \( 450^\circ \) has to be described by two circles: \( 300^\circ \leq \varphi \leq 360^\circ \) and \( 0^\circ \leq \varphi \leq 90^\circ \).

- 'Tol' is a value to define the convergence radius around the line. With the default value 'tol' = 1000 usually the associated nodes are found. In special situations (fine discretization) a higher value may be necessary to find only the nodes exact on the given line.

- 'bou': For bou = 0 (default) external loads are calculated only if no boundary condition is set. On the other hand, for bou = 1, the given load is set directly on the node as a single load! Use for this purpose load case 1! Thus, if boundary conditions are set, the load is assumed as given displacement.

- Proof if all nodes are correct loaded with
  - \# load (nodes and load vectors)
  - \# load,,,,-1 (only nodes marked) in the plot mode.

- eloa adds the loads to the nodal force vector.

- 'Rad' is the radius of the circle. Input for other types 'Rad' = 0.

- 'etyp' is the parameter to define on which type of elements loads are generated:
  - # etyp = 1: linear elements (2 nodes on boundary)
  - # etyp = 2: quadratic elements (3 nodes on boundary)

- 'ltyp' is the parameter to define the special loading parameter due to the following table

<table>
<thead>
<tr>
<th>load case</th>
<th>ltyp</th>
<th>d(1)</th>
<th>d(2)</th>
<th>d(3)</th>
<th>d(4)</th>
<th>d(5)</th>
<th>d(6)</th>
<th>d(7)</th>
<th>d(8)</th>
<th>d(9)</th>
<th>d(10)</th>
<th>d(11)</th>
<th>d(12)</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant loads</td>
<td>1</td>
<td>( q_{lx} )</td>
<td>( q_{px} )</td>
<td>( q_{oy} )</td>
<td>( q_{py} )</td>
<td>( q_{oz} )</td>
<td>( q_{pz} )</td>
<td>( m_{ox} )</td>
<td>( m_{px} )</td>
<td>( m_{oy} )</td>
<td>( m_{py} )</td>
<td>( m_{oz} )</td>
<td>( m_{pz} )</td>
</tr>
<tr>
<td>linear loads</td>
<td>2</td>
<td>( q_{lx} )</td>
<td>( q_{bx} )</td>
<td>( q_{ax} )</td>
<td>( q_{ay} )</td>
<td>( q_{az} )</td>
<td>( q_{0x} )</td>
<td>( q_{0y} )</td>
<td>( q_{0z} )</td>
<td>( q_{0z} )</td>
<td>( q_{0p} )</td>
<td>( q_{0z} )</td>
<td>( q_{0z} )</td>
</tr>
</tbody>
</table>

- Loads acts in global 1 – 3 direction on specified length
\[ ip = 0 \quad ds = \sqrt{(dx^2 + dy^2 + dz^2)} \] element length e.g. load case dead load
\[ ip = 1 \quad ds = dx \] e.g. load case snow
\[ ip = 2 \quad ds = dy \]
\[ ip = 3 \quad ds = dz \]
\[ ip = 12 \quad ds = \sqrt{(dx^2 + dy^2)} \]
\[ ip = 13 \quad ds = \sqrt{(dx^2 + dz^2)} \]
\[ ip = 23 \quad ds = \sqrt{(dy^2 + dz^2)} \]

- constant loads are defined as
  \[ q_x(s) = q_{0x} \quad m_x(s) = m_{0x} \]
  \[ q_y(s) = q_{0y} \quad m_y(s) = m_{0y} \]
  \[ q_z(s) = q_{0z} \quad m_z(s) = m_{0z} \]

- linear loads are defined as
  \[ q_x(s) = q_{ax} + \frac{s}{l_{ab}} \cdot (q_{bx} - q_{ax}) \] with \( l_{ab} = ds \) specified above.
  \[ q_y(s) = q_{ay} + \frac{s}{l_{ab}} \cdot (q_{by} - q_{ay}) \]
  \[ q_z(s) = q_{az} + \frac{s}{l_{ab}} \cdot (q_{bz} - q_{az}) \]

- Remarks:
  1.) Do not use this macro for Bernoulli-type elements.
  2.) The macro can be used in the axisymmetric case with \( q_{axi} = q \cdot 2\pi \cdot r \).
  3.) Proof that the loads acts correctly on quadratic elements.
  4.) The intermediate node for quadratic elements must on a straight line and the MIDpoint.
  5.) Force input may be used as displacements if boundary conditions are set. Thus, these points could not have loads! If loads are generated from \texttt{eloa}, delete them by using the macro \texttt{load}.
  6.) Up to now use macro only for straight lines.
  7.) Use the parameter 'bou' for given displacements.
  8.) \texttt{eloa} and \texttt{react} lead to wrong results for associated dofs at boundaries.
The last command in the mesh input must be **end**. Then special actions like searching for double nodes with **tie** or linking conditions with **link** are done. It follows the execution of the program, which is controlled by **macro** commands. The execution of these commands can be done in batch-modus **macro** or in interactive modus **inte**. In the first case the commands are defined in the input-file after the **macro** command, in the second case the commands are defined interactively.

**stop** is the final command in the input file, which stops the execution.
EPSQ

\texttt{epsq}

\texttt{iswm, nss, natyp, fx, fy}

\texttt{( epsq(i), i=1, nss)}

Input of transfer data

1. line

- \texttt{iswm} control switch parameter
- \texttt{nss} no. of strain parameter: 6 (3D), 8 (Shell), 10 (Shell ext.), 6 (3D-beam), 15 (3D-beam ext.)
- \texttt{natyp} 1 (3D), 2 (shell), 3 (shell ext.), 4 (3D-beam), 5 (3D-beam ext.)
- \texttt{fx} factor in $\kappa_x = \frac{1}{1 + f_x(\ell/h)^2}$
- \texttt{fy} factor in $\kappa_y = \frac{1}{1 + f_y(\ell/h)^2}$

2. line

Input of strain array for

- 3D \texttt{nss=6} \quad \textbf{E} = [\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, 2\varepsilon_{12}, 2\varepsilon_{13}, 2\varepsilon_{23}]^T
- inextensible shell \texttt{nss=8} \quad \textbf{E} = [\varepsilon_{11}, \varepsilon_{22}, 2\varepsilon_{12}, \kappa_{11}, \kappa_{22}, 2\varepsilon_{13}, 2\varepsilon_{23}]^T
- extensible shell \texttt{nss=10} \quad \textbf{E} = [\varepsilon_{11}, \varepsilon_{22}, 2\varepsilon_{12}, \kappa_{11}, \kappa_{22}, 2\varepsilon_{13}, 2\varepsilon_{23}, \varepsilon_{0}^{33}, \varepsilon_{33}]^T
- inextensible beam \texttt{nss=6} \quad \textbf{E} = [\varepsilon_{11}, 2\varepsilon_{12}, 2\varepsilon_{13}, \vartheta, \kappa_{11}, \kappa_{22}]^T
- extensible beam \texttt{nss=15} \quad \textbf{E} = [\varepsilon_{11}, 2\varepsilon_{12}, 2\varepsilon_{13}, \vartheta, \kappa_{11}, \kappa_{22}, \varepsilon_{0}^{32}, \varepsilon_{0}^{22}, \varepsilon_{1}^{22}, \varepsilon_{2}^{22}, \varepsilon_{33}, \varepsilon_{33}, \varepsilon_{23}, \varepsilon_{23}, \varepsilon_{33}]^T

Strain data are input for the calculation of prescribed displacements $\textbf{V} = \textbf{A} \textbf{E}$ within macro \texttt{epsq}.
**FAST**

```plaintext
fast
Name of input file
```

- Input data are stored before using an input file called Ifile via macro `fast` on a binary file called Ifile.sys.
- Produce a new input file like this

```
fast
Name of Input file: Ifile
(including path, without `.sys')
end

<macr>
↑ MACRO - Macros
↓
end
```

Optional: macros for automatic calculation Input

- Use this input file
The `fixe` command is used to specify the values for the boundary restraint conditions at segments. For each segment to be specified a record is entered with the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>segm</code></td>
<td>the number of segment to be specified</td>
</tr>
<tr>
<td><code>id(1,segm)</code></td>
<td>value of boundary restraint for 1–dof for 'segm'</td>
</tr>
<tr>
<td><code>id(2,segm)</code></td>
<td>value of boundary restraint for 2–dof for 'segm'</td>
</tr>
<tr>
<td><code>etc.</code></td>
<td>etc. until to 'ndf' directions</td>
</tr>
</tbody>
</table>

The boundary restraint codes are interpreted as follows:

- `id(i,segm) = 0`  i–dof has unknown displacement,
- `id(i,segm) = 1`  i–dof has known displacement 0.

**Remarks:**

1. For mesh generation with NEGE the following order of statements is necessary: `nege`, `[neco]`, `[back]`, `geom`, `segm`, `regi`, `[pres]`, `[fixe]`, further FEAP-macros. Note: macros in brackets are optional.

2. Compare the macro `boun`.
This macro is up to now not documented.
The `gcor` command is used to specify the values for nodal coordinates for the generation program GMESH. The only difference to `coor` is that no generation increment can be used. For each node to be specified a record is entered with the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>the number of the node to be specified</td>
</tr>
<tr>
<td>x(1,node)</td>
<td>value of coordinate in 1-direction for ‘node’</td>
</tr>
<tr>
<td>x(2,node)</td>
<td>value of coordinate in 2-direction for ‘node’</td>
</tr>
<tr>
<td>&lt;x(3,node)&gt;</td>
<td>value of coordinate in 3-direction for ‘node’</td>
</tr>
<tr>
<td>&lt;ityp&gt;</td>
<td>0=cart., 1=pola(1-2), 2=pola(2-3), 3=pola(1-3), 4=sphe</td>
</tr>
</tbody>
</table>

Input could be given in different coordinate systems, see ‘ityp’.

Output is always in cartesian coordinates.
The `gele` command is used to specify the values of nodal numbers which are attached to each element for program GMESH. The only differences to `elem` are that no generation is available and always 8 nodes are used. For each element to be specified, a record is entered with the following information:

<table>
<thead>
<tr>
<th>nelm</th>
<th>– the number of the element to be specified.</th>
</tr>
</thead>
<tbody>
<tr>
<td>matl</td>
<td>– the material number for the element, this will determine the element type.</td>
</tr>
<tr>
<td>ix(1,nelm)</td>
<td>– node–1 number attached to element.</td>
</tr>
<tr>
<td>ix(2,nelm)</td>
<td>– node–2 number attached to element, etc., to '8' nodes.</td>
</tr>
</tbody>
</table>

Node numbering of bloc is in contrast to `elem` of FEAP: corner-node, mid-side-node, corner-node, ... etc. anti-clockwise and is shown below:

```
    7  6  5
   o-----o-----o
  |     |     |
8 o     o  4
  |     |     |
    o-----o-----o
  1  2  3
```
GEOM

The `geom` command is used to specify the values for nodal coordinates for segment nodes. For each node to be specified a record is entered with the following information:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>the number of the node to be specified</td>
</tr>
<tr>
<td>x(1,node)</td>
<td>value of coordinate in 1–direction for ‘node’</td>
</tr>
<tr>
<td>x(2,node)</td>
<td>value of coordinate in 2–direction for ‘node’</td>
</tr>
</tbody>
</table>

Remark:

For mesh generation with NEGE the following order of statements is necessary: `nege`, `[neco]`, `[back]`, `geom`, `segm`, `regi`, `[pres]`, `[fixe]`, further FEAP-macros. Note: macros in brackets are optional.
GMESH

```
gmesh [ title of problem for printouts, etc.]
node,ndf,ndm,iopt,ityp,iprin
```

Each problem to be solved by FEAP and using the mesh generator GMESH must start with a single record which contains the characters `gmesh` as the first entry; the remainder of the record (columns 5-80) may be used to specify a problem title. The title will be printed with each “page” of output as the first line. Immediately following the `gmesh` record, the ‘control’ information describing characteristics of the problem to be solved must be given. The ‘control’ information describes the characteristics of the finite element problem to be solved. The data entries are:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>node</code></td>
<td>number of nodes per element.</td>
</tr>
<tr>
<td></td>
<td>3 node triangle, 4/8 node quadrilateral.</td>
</tr>
<tr>
<td><code>ndf</code></td>
<td>maximum number of degrees-of-freedom on any node.</td>
</tr>
<tr>
<td><code>ndm</code></td>
<td>maximum number of dimensions.</td>
</tr>
<tr>
<td><code>iopt</code></td>
<td>0/1 without/with node numbering optimization.</td>
</tr>
<tr>
<td><code>ityp</code></td>
<td>0 = generate mesh, 1 = check input data.</td>
</tr>
<tr>
<td><code>iprin</code></td>
<td>0 = no action, 1 = print additional information.</td>
</tr>
</tbody>
</table>

Remarks:

1. A FEAP-input file will be generated called ifeap.tmp (always).
2. Control information of generation can be found in the file GMESH.log.
3. For mesh generation with GMESH the following order of statements is necessary:
4. `gmesh`, `[neco]`, `gcor`, `gele`, further FEAP-macros.
   Note: macros in brackets are optional.
5. For further information see the GMESH-manual.
The **icon** macro is used to define surfaces which may interact through contact during the analysis of solids. It is necessary to specify:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>nsl</code></td>
<td>the total number of potential contact surfaces.</td>
</tr>
<tr>
<td><code>nsntl</code></td>
<td>the total number of nodes on all “slave” surfaces.</td>
</tr>
<tr>
<td><code>nmntl</code></td>
<td>the total number of nodes on all “master” surfaces.</td>
</tr>
<tr>
<td><code>naxi</code></td>
<td>flag (0 = plane, 1 = axisymmetric problem.)</td>
</tr>
</tbody>
</table>

When looking along a potential contact surface the “slave” surface is on the left side and the “master” surface is on the right side. Nodes defined on surfaces must always be specified with the “master” surface on the right as the sequence is input. For each contact surface (i = 1 to nsl), the following slideline control information is required:
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nsln</td>
<td>number of slave nodes per slide line</td>
</tr>
<tr>
<td>nmln</td>
<td>number of master nodes per slide line</td>
</tr>
</tbody>
</table>
| islt     | slide–line type:  
|          | 1 = sliding only;  
|          | 2 = slide–line tied;  
|          | 3 = slide + void;  
|          | 4 = friction + void. |
| ifla     | flag for contact algorithm  
|          | 1 = 1-pass algorithm  
|          | 0 = 2-pass algorithm  
|          | (change role of master and slave surface during 2\textsuperscript{nd} pass) |
| sfacn    | stiffness scaling factor normal (penalty) |
| sfact    | stiffness scaling factor tangential (penalty) |
| ityp     | type of solution method:  
|          | 0 = Penalty;  
|          | 1 = Augmented Lagrangian; simultaneous iteration  
|          | (implemented for frictionless contact only);  
|          | 2 = Augmented Lagrangian, nested iteration,  
|          | symmetrized treatment of friction;  
|          | 3 = Augmented Lagrangian, nested iteration,  
|          | non-symmetric treatment of friction. |
| itan     | type of solution tangent matrix:  
|          | 0 = Consistent;  
|          | 1 = Nonsymmetric contribution to tangent omitted for frictional problems (not ordinarily recommended);  
|          | 2 = Linearized. |
After the above slideline control information, the following contact node data is required:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>beta</td>
<td>friction coefficient on this surface</td>
</tr>
<tr>
<td>na</td>
<td>local number of slave node on this surface</td>
</tr>
<tr>
<td>nsvn(na)</td>
<td>global (mesh) node number associated with na.</td>
</tr>
<tr>
<td>msrn(na)</td>
<td>global (mesh) node number associated with na.</td>
</tr>
</tbody>
</table>

..... repeat until all nodes are input (or generated).

When specifying nodes on a master or slave surface the local nodes must be input in an increasing sequence (i.e., 1, 2, 5, ...). If a node is missing in the sequence the global node will be generated according to the following relationship:

\[ n_{svn}(i+1) = (n_{svn}(nb) - n_{svn}(na))/(nb - na) + n_{svn}(i) \]

It is also necessary to indicate on the macro program level with the macro **cont** that a contact problem is being solved. If a solution is desired without using the contact option it is necessary to reset the contact solution using **cont**, off. It is also possible to reset the value of the penalty parameters during execution using the **cont** macro.

**Further remarks:**

The augmented Lagrangian solution schemes available are one solution to difficulties posed by the penalty method; i.e., potential ill-conditioning on one hand and underpenalization of constraints on the other. The simultaneous technique (option 1) performs the augmentations during the equilibrium iterations, and has the drawback that it does not preserve quadratic rates of convergence. The nested techniques (options 2 and 3) are identical in the frictionless case, but differ in their treatments of friction. The symmetrized option only enforces the Coulomb condition in the augmentation phase, while the nonsymmetric option enforces it both in the solution phase and in the augmentation phase. The term ‘nested’ refers to the manner in which augmentations are performed; within a time step, they are done after equilibrium has been enforced using the current Lagrange multipliers. That is, solutions (with the associated Newton iterations) and augmentations are done recursively and completely separately until convergence of the multipliers is achieved. In the nested schemes, the augmentations must be requested by the user at the time they are desired by the use of the CAUG macro; no such request need be made in the simultaneous treatment.

**References**

Details can be found in Wriggers, Simo (1985) [30] and Simo, Wriggers, Schweizerhof, Taylor (1986) [24].
ICOR

\texttt{icor}
\begin{verbatim}
  (xperf(i),i=1,ndm),(ximperf(i),i=1,ndm)
<etc., terminate with blank record>
\end{verbatim}

The \texttt{icor} command is used to modify perfect coordinates due to (e.g. measured) imperfections. This macro should be used at the end of the input file. Due to the modification of coordinates macros like \texttt{eloa} or \texttt{edge} may not work.
**IMPF**

```plaintext
impf
α
node1, ngen1, (impf(i,node1),i=1,ndm)
node2, ngen2, (impf(i,node2),i=1,ndm)
<etc., terminate with blank record>
```

The `impf` command is used to modify the values for nodal coordinates. The handling is identical to the macro `coor`.

The coordinates are modified due to

\[ \mathbf{x} = \mathbf{x} + \alpha \cdot \text{impf} \]

Be careful with the use of this macro. Due to the modification of the coordinates macros like `eloa`, `edge` or `tie` may not work. Thus use for example `eloa` before `impf`.

The vector \( \mathbf{u} = \alpha \cdot \text{impf} \) can be plotted using `dimp` directly after start of FEAP. Later `dimp` plots \( \mathbf{u} + \mathbf{u} \).
The last command in the mesh input must be `end`. Then special actions like searching for double nodes with `tie` or linking conditions with `link` are done. It follows the execution of the program, which is controlled by `macro` commands. The execution of these commands can be done in batch-modus `macro` or in interactive modus `inte`. In the first case the commands are defined in the input-file after the `macro` command, in the second case the commands are defined interactively.

`stop` is the final command in the input file, which stops the execution.
ISEC

ise
c
$a_1, a_2, a_3, b_1, b_2, b_3, tol$
<etc., terminate with a blank record>

- This macro sets dof6 = 1 for shells with 5 dofs and release this dof (dof6 = 0) at intersections.
- The intersection line is straight and defined by $P_a(a_1, a_2, a_3)$, and $P_b(b_1, b_2, b_3)$.
- ’Tol’ is a value to define the convergence radius around the line. With the default value ’tol’ = 1000 usually the associated nodes are found. For fine discretizations a higher value may be necessary to find only the nodes exact on the given line.
- Proof if all nodes are correct found with isec in the plot mode.
The **jint** macro is used to input a series of elements which define the path along which the $J$–integral has to be evaluated. It is necessary to specify:

<table>
<thead>
<tr>
<th>njint</th>
<th>the total number of elements along the path.</th>
</tr>
</thead>
<tbody>
<tr>
<td>nel1–nel8</td>
<td>the element numbers (8 per record).</td>
</tr>
</tbody>
</table>

When looking along the path along which the $J$–integral has to be computed the series of elements have to be put in "anti-clock-wise".
knv1 defines the knot vector in direction of parameter \( \Xi^1 \). After the keyword knv1 every line contains one knot. The total number of lines after knv1 must be

$$\sum_{i=1}^{n_{\text{patch}}} n_i + p_i + 1$$

due to NURBS conventions (as open knot vectors have to be used). The line with the entry

$$\Xi^1_i (\xi^1_j)$$

contains the \( j \)-th knot of the \( i \)-th patch in \( \Xi^1 \)-direction. The rules regarding digits and exponents are the same as for macro coor. knv1 and knv2 have to be placed after nmpq.
**KNV2**

```
knv2
Ξ^2_0 (ξ^2_1)
Ξ^2_1 (ξ^2_2)
...
Ξ^2_{m_i+q_i+1} (ξ^2_{m_i+q_i+1})
...
Ξ^2_{n_patch} (ξ^2_{m_i+q_i+1})
```

`knv2` defines the knot vector in direction of parameter Ξ^2. After the keyword `knv2` every line contains one knot. The total number of lines after `knv2` must be

\[ \sum_{i=1}^{n\text{\_patch}} m_i + q_i + 1 \]

due to NURBS conventions (as open knot vectors have to be used). The line with the entry

\[ Ξ^2_i (ξ^2_j) \]

contains the j-th knot of the i-th patch in Ξ^2-direction. The rules regarding digits and exponents are the same as for macro `coor`. `knv1` and `knv2` have to be placed after `nmpq`. 
LINK

**link**

1. `node1,node2,inc,(idl(i),i=1,ndf)`
   or
2. `node1,idm, x0 , (idl(i),i=1,ndf),eps`
   or
3. `node_is,node_je,inc_i,node_js,node_je,inc_j,(idl(i),i=1,ndf)`
   or
4. `idm, x1 , x2, (idl(i),i=1,ndf),eps`
   or
5. `idmx, x1, x2, idmy, y1, y2,(idl(i),i=1,ndf),eps`
   or
6. `idmx, x1, x2,(idl(i),i=1,ndf),eps`
   or
7. `lx1, lx2, lx3,(idl(i),i=1,ndf)`
   <terminate with a blank record>

A mesh may be generated in FEAP in which it is desired to let the values at more than one node share the same ‘displacement’ unknown. For example, in repeating structures the value of the dependent variable will be the same at each repeating interval. In a finite element model it is necessary to specify the repeating condition by ‘linking’ the degree-of-freedoms at these nodes to the same unknown in the equations. The **link** command is used for this purpose.

To use the **link** option the complete mesh must first be defined. After the **end** command for the mesh definition and before the **macro**, **batch** or **interactive** command for defining a solution algorithm, the use of a **link** statement together with the list of affected nodes and degree-of-freedoms will cause the program to search for all conditions that are to be connected together. A tolerance is defined to find the nodes. If this is not successful a user defined value ‘eps’ can be used, default ‘eps=0’.

The input data are interpreted as follows:

**Type 1: link nodes from node 1 to node 2 to master node 1**

| node1       | first node in a sequence to be linked = master node |
| node2       | last node in a sequence to be linked               |
| inc         | increment to generate intermediate nodes affected. |
| idl(1)      | linking condition, 0 = link, 1 = do not link this dof. |
| idl(2)      | linking condition, 0 = link, 1 = do not link this dof. |

etc. for ‘ndf’ degree of freedoms
Type 2: link nodes along const. coordinate to master node 1

- **node1** – master node to which other nodes are linked
- **idm** – direction of coordinate (i.e., 1=x, 2=y, 3=z)
- **x0** – the value of the idm-direction coordinate
- **idl( )** – see above
- **eps** – user defined search tolerance

Type 3: link nodes **j** to nodes **i** (master), where nodes **i** and **j** are out of sequences.

Thus, type 3 is an automatic repetition of type 1 for given sequences of nodes.

- **node_is** – first node in a sequence of master nodes
- **node_ie** – last node in a sequence of master nodes
- **inc_i** – increment to generate intermediate nodes affected.
  (default = node_ie–node_is)
- **node_js** – first node in a sequence to be linked with node_is
- **node_je** – last node in a sequence to be linked with node_ie
- **inc_j** – increment to generate intermediate nodes affected.
  (default = node_je–node_js)
- **idl(1)** – linking condition, 0 = link, 1 = do not link this dof.
- **idl(2)** – linking condition, 0 = link, 1 = do not link this dof.
  etc. for ‘ndf’ degree of freedoms

Type 4: link nodes along coordinate x2 to nodes along coordinate x1

- **idm** – direction of coordinate (i.e., 1=x, 2=y, 3=z)
- **x1** – the value x1 of the idm-direction coordinate
- **x2** – the value x2 of the idm-direction coordinate
- **idl( )** – see above
- **eps** – see above
Thus for idm=1 nodes \((x2,yi,zi)\) are linked to nodes \((x1,yi,zi)\) for same \(yi,zi\).

**Type 5: link nodes along line \(x2,y2\) to nodes along line \(x1,y1\)**

<table>
<thead>
<tr>
<th>idmx</th>
<th>direction of coordinate (i.e., 1=x,2=y,3=z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>the value (x1) of the idmx-direction coordinate</td>
</tr>
<tr>
<td>x2</td>
<td>the value (x2) of the idmx-direction coordinate</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>idmy</th>
<th>direction of coordinate (i.e., 1=x,2=y,3=z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>y1</td>
<td>the value (y1) of the idmy-direction coordinate</td>
</tr>
<tr>
<td>y2</td>
<td>the value (y2) of the idmy-direction coordinate</td>
</tr>
</tbody>
</table>

\(idl(\ )\) – see above

\(eps\) – see above

Thus for idmx=1, idmy=2 nodes \((x2,y2,zi)\) are linked to nodes \((x1,y1,zi)\) for same \(zi\).

**Type 6: link nodes along coordinate \(x2\) to nodes along coordinate \(x1\) cross-wise**

\(y2(x2)=-y1(x1)\), **only for \(x,y\)-direction!**

The origin of the coordinate system must be at least \((0,0,z)\).

<table>
<thead>
<tr>
<th>idm</th>
<th>direction of coordinate (i.e., 1=x,2=y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>the value (x1) of the idm-direction coordinate</td>
</tr>
<tr>
<td>x2</td>
<td>the value (x2) of the idm-direction coordinate</td>
</tr>
</tbody>
</table>

\(idl(\ )\) – see above

\(eps\) – see above

Thus for idm=1 nodes \((x2,+yi,zi)\) are linked to nodes \((x1,-yi,zi)\) for same \(yi,zi\).

This macro is only used for FE\(^2\) on the RVE.

**Type 7: link border nodes of a RVE with length \(lx1, lx2, lx3\).** The origin of the coordinate system must be \((0,0,0)\). All border nodes of a plane \([x1, x2, x3]\) are linked to a node \([-lx1/2, -lx2/2, x3]\).

<table>
<thead>
<tr>
<th>lx1</th>
<th>length (lx1) of the RVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>lx2</td>
<td>length (lx2) of the RVE</td>
</tr>
<tr>
<td>lx3</td>
<td>length (lx3) of the RVE</td>
</tr>
</tbody>
</table>

\(idl(\ )\) – see above

This macro is only used for FE\(^2\) on the RVE.
Remarks:

1. Termination of input occurs with a blank record.

2. Whenever it is desired to only connect 'node1' to 'node2', 'inc' need not be specified (it may be blank or zero).

3. Be extremely careful using this macro, because input errors could be seen very difficult! A graphical control with link is strongly recommended!

4. Be extremely careful using this macro with multiple records due to possible dependencies.

5. Circle links (like node1 → node2 and then node2 → node1) are identified. Otherwise necessary dofs are deleted and FEAP do not return after a tang-macro.

6. In case loads should be specified for the linked nodes, only one node is specified with the resulting load of all nodes. Output for reac and resi is not correct at linked nodes!

7. In case of boundary conditions should be specified for the linked nodes, only the b.c. for node1 has to be specified.

8. In case of boundary conditions with prescribed displacements link should not be used! Use boun, ebou etc. for setting the b.c. and disp or edis for setting the prescribed values for all nodes.

9. link could be used together with tie. Thus it is allowed to use the 'original' node numbers of tied nodes.

10. Linking conditions could be set for a DOF of a node only once!! If this rule is violated an error message occur in FEAP. The desired linking condition is not executed! Nevertheless linking conditions for different DOFs of a node are possible in more than one call of link.

11. Up to 16 values are possible on each input record.

12. Input of 'eps' is only possible, if one record is sufficient for whole input.
load
node1, ngen1, (f(i,node1),i=1,ndf)
node2, ngen2, (f(i,node2),i=1,ndf)
<etc., terminate with blank record>

The **load** command is used to specify the values for nodal boundary loads. For each node to be specified a record is entered with the following information:

<table>
<thead>
<tr>
<th>node</th>
<th>the number of the node to be specified.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ngen</td>
<td>the increment to the next node, if generation is used, otherwise 0.</td>
</tr>
<tr>
<td>f(1,node)</td>
<td>value of load for 1-dof for 'node'</td>
</tr>
<tr>
<td>f(2,node)</td>
<td>value of load for 2-dof for 'node'</td>
</tr>
<tr>
<td>etc., to 'ndf' directions</td>
<td></td>
</tr>
</tbody>
</table>

When generation is performed, the node number sequence will be (for node1–node2 sequence shown above):

```
node1, node1+ngen1, node1+2*ngen1, ..., node2
```

The values for each load will be a linear interpolation between f(node1) and f(node2).

**Remarks:**

1. Up to 16 values are possible on each input record.
2. Loads on bounded DOFS, introduced via the macro command **boun** (or similar macros), are interpreted as prescribed displacements!!
3. The **load**-macro is identical to the old **force**-macro.
The `loa0` command is used to specify the values for nodal loads which are not time dependent. For each node to be specified a record is entered with the following information:

<table>
<thead>
<tr>
<th>node</th>
<th>the number of the node to be specified.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ngen</td>
<td>the increment to the next node, if generation is used, otherwise 0.</td>
</tr>
<tr>
<td>f(1,node)</td>
<td>value of load for 1–dof for ’node’</td>
</tr>
<tr>
<td>f(2,node)</td>
<td>value of load for 2–dof for ’node’</td>
</tr>
<tr>
<td></td>
<td>etc., to ’ndf’ directions</td>
</tr>
</tbody>
</table>

When generation is performed, the node number sequence will be (for node1–node2 sequence shown above):

`node1, node1+ngen1, node1+2*ngen1, .... , node2`

The values for each load will be a linear interpolation between f(node1) and f(node2).

Up to 16 values are possible on each input record.

On macro level these values could be set via the macro `newf`. 
The **mate** command is used to specify the parameters for each material set in the analysis, as well as to specify the particular element type associated with the properties.

Optional it is possible to change the position of degrees of freedoms in the problem. Thus the idfg-array define the global dofnumbers for the used element.

**Example.** A plate element (e.g. iel=7) with 3 dofs ($w, \varphi_x, \varphi_y$) can be combined with a 3D-beam element (e.g. iel=11) with 6 dofs ($u, v, w, \varphi_x, \varphi_y, \varphi_z$) in the following way:

- Plate local number: 1 ($w$) 2 ($\varphi_x$) 3 ($\varphi_y$)
- Global number: 3 4 5

**FEAP input:**

```
mate
1,11
Beam specific data ......
2,7, 3,4,5
Plate specific data ..... 
```

The specific parameters to be input are described in each element manual (e.g., see manual for 'elmt01' for the parameters associated with iel = 1', etc.).
The solution algorithm used by FEAP to solve problems is defined by a “macro statement program”. By properly specifying the macro program a very wide range of applications may be addressed — including both linear and nonlinear, as well as, steady state and transient applications.

The description for the macro statements may be obtained from the manual entry for each command, (e.g., use manual TANG to obtain all options and actions for the tang command).
The mesh macro commands \texttt{mesn} \((n = 1,...,5)\) are currently not implemented. These commands refer to subroutine calls UMESHn in SR pmesh. They can be used by a programmer who wants to add new input macro features to FEAP.
**NDVI**

The **ndvi** command is used to specify the values of subdivisions of all blocs defined for the mesh generation with GMESH. For each bloc to be specified, two records are entered with the following information:

```

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nelm</td>
<td>the number of the bloc to be specified.</td>
</tr>
<tr>
<td>ndvix1</td>
<td>number of subdivisions in 1-direction.</td>
</tr>
<tr>
<td>ndviy1</td>
<td>number of subdivisions in 2-direction.</td>
</tr>
<tr>
<td>dx1(i)</td>
<td>ndvix1*subdivisions in 1-direction.</td>
</tr>
<tr>
<td>dy1(i)</td>
<td>ndvix1*subdivisions in 2-direction.</td>
</tr>
</tbody>
</table>
```

Remarks:
1) At connection lines of different blocs the same subdivision has to be used.
2) The input of values dx1,dy1 etc. is only necessary for a non-constant division.
   Then:
3) Values can be chosen arbitrary. The subdivision bases on relative results from the input values.
4) Up to 16 values are possible on each input record.
Each problem to be solved by **FEAP** and using the mesh generator **NEGE** must start with a single record which contains the characters **nege** as the first entry; the remainder of the record (columns 5-80) may be used to specify a problem title. The title will be printed with each “page” of output as the first line. Immediately following the **nege** record, the ‘control’ information describing characteristics of the problem to be solved must be given. The ‘control’ information describes the characteristics of the finite element problem to be solved. The data entries are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| ntyp      | - 1 = background mesh (control).  
- 2 = mesh from segments (control).  
- 3 = mesh from regions (control).  
- 4 = generates FE mesh. |
| node      | number of nodes per element.  
- 3/6 node triangle, 4/8/9 node quadrilateral. |
| ndf       | maximum number of degrees-of-freedom on any node.  
- (max for NEGE is 3!). |
| ncorr     | 0 = region with straight boundarys,  
- 1 = region with curved boundarys. |
| esiz      | approximate element length in diagonal direction within the surrounded rectangle, 10-0.5 ...(rough-fine). |

**Remarks:**

1. It will be recommended to test data input with **ntyp** = (1),2 to 4. 'Ntyp' = (1),2,3 is only used to control the input data for mesh generation. In these cases the plot-options **node**, **elem**, **mesh**, **load**, **boun** are still working.

2. 'Ndm' must be 2 (number of spatial coordinates).

3. A FEAP-input file will be generated called ifeap.tmp (always).

4. Control information of generation can be found in the file NEGE.log.

5. For mesh generation with NEGE the following order of statements is necessary: **nege**, [**neco**], [**back**], **geom**, **segm**, **regi**, [**pres**], [**fixe**], further FEAP-macros. Note: macros in brackets are optional.

6. For further information see the **NEGE**-manual.
This macro is similar to the macro `para`. All definitions which are defined by `neco` hold for the mesh generation and FEAP.

**Remark:**
For mesh generation with NEGE the following order of statements is necessary: `nege`, `[neco]`, `[back]`, `geom`, `segm`, `regi`, `[pres]`, `[fixe]`, further FEAP-macros. Note: macros in brackets are optional.
The \texttt{nmpq} command is used to specify the shape of the mesh and the order of the basis functions. In the first line the number of patches ‘\texttt{n\_patch}’ has to be stated. As the control points are entered via a linear list, the dimensions of the control point ‘matrices’ have to be given. For all patches the following constants have to be given. \( n \) is the number of control points in \( \Xi^1 \)-direction, \( m \) in \( \Xi^2 \)-direction. \( p \) and \( q \) are the corresponding orders. ‘\texttt{mat}’ gives the element type (at the moment only element 60 is available, which is derived from element 5). Every line except the first line creates one patch. \texttt{nmpq} has to appear after \texttt{coor} and before \texttt{knv1} and \texttt{knv2}.

<table>
<thead>
<tr>
<th>\texttt{n_patch}</th>
<th>the number of patches</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{n}</td>
<td>the number of control points in ( \Xi^1 )-direction</td>
</tr>
<tr>
<td>\texttt{m}</td>
<td>the number of control points in ( \Xi^2 )-direction</td>
</tr>
<tr>
<td>\texttt{p}</td>
<td>the order of basis functions in ( \Xi^1 )-direction</td>
</tr>
<tr>
<td>\texttt{q}</td>
<td>the order of basis functions in ( \Xi^2 )-direction</td>
</tr>
<tr>
<td>\texttt{mat}</td>
<td>element type</td>
</tr>
</tbody>
</table>

\texttt{nmpq n\_patch n1, m1, p1, q1, mat1 n2, m2, p2, q2, mat2 <etc., terminate with blank record>}

\( \texttt{coor} \)
**NOPA**

**nopa**

*nopa* allows all input data only numerical. Thus parameters are not allowed in input data. The opposite command is *pars* which allows all input data as expressions.
The use of the `nopr` macro command will discontinue the output of all subsequent mesh data (except for material data printed in each element). The use of `prin` will cause the output of mesh information to again be reported. The default value is `prin` at start of each problem execution.
A mesh may be generated in **FEAP** without looking on a good node numbering which leads to small differences for the node numbers. This may be essential for the bandwidth of the stiffness matrix and therefore for the necessary memory and for the solution time of solving equations.

Thus two optimization procedures are available which lead to an internal renumbering of the nodes.

To use the **opti** option the complete mesh must first be defined. After the **end** command for the mesh definition and before the **macro** or **interactive** command for defining a solution algorithm, **opti** is used to optimize the input data.

Optimization procedures:

- **opti** - bases on Minimum front criteria (Hoit-Wilson)
- **optn** - bases on the 'Cuthill-McKee Algorithm'

Note that the optimization procedure bases on the chosen basic node numbering. Furthermore it is not clear which algorithm leads to better results, thus try both ones and compare the results shown in the first **FEAP**-output.
The `page` command may be used to specify the 'character' which produces a page feed on a line printer for the printout of the results. In 'character', 4 single characters may be defined.
### PARA

Data input specifications in FEAP may be in the form of numerical constants, parameters, or expressions. Constants are conventional forms for specifying input data. Parameters are single character lower case letters to which values are assigned. Expressions are combinations of constants and parameters and functions.

Here, the `para` macro command may be used to assign values to letter parameters.

A letter parameter is defined immediately following the `para` macro (several may follow terminating with a blank record) according to the following:

\[
x = \text{input data}
\]

where 'x' may be any of the single letters (a-z), any group of two letters (aa-zz), or any letter and a numeral (a0-z9) followed by the equal sign. All alphabetic input characters are automatically converted to lower case, hence there are 962 unique parameters permitted at any one time.

The input data may be:

- numbers (floating point numbers can contain an 'e' or a 'd' exponent format),
- previously defined letter constants,
- expressions.

Expressions can be calculated from numbers, parameters and functions. The expression is processed left to right and can contain one set of parentheses to force groupings.

The following arithmetic operations +, −, *, / or ^ are allowed.

The following functions may appear in an expression, a statement, or a parameter definition:

\[
\begin{align*}
\sin & \quad \sin d & \quad \text{asin} & \quad \text{asind} & \quad \text{sinh} & \quad \text{abs} & \quad \text{int} \\
\cos & \quad \cos d & \quad \text{acos} & \quad \text{acosd} & \quad \text{cosh} & \quad \text{exp} & \quad \text{log} & \quad \text{sqrt} \\
\tan & \quad \tan d & \quad \text{atan} & \quad \text{atan2} & \quad \text{atanh} & \quad \text{inc} & \quad \text{dec}
\end{align*}
\]

The trigonometric and inverse trigonometric functions which end in d involve values of angles in degrees; whereas, the ones without involve values in radians. Each function has one argument which is contained between parenthesis (which counts as the one level of depth). The argument may be an expression but may not contain any parenthesis or additional functions.

The function inc(val+) increases a value by val+, reset to 0 is given by val+ = 0, whereas the function dec(val-) increases a value by val-, reset to 0 is given by val- = 0.

A hierarchical evaluation is performed according to the rules defined below

<table>
<thead>
<tr>
<th>Order</th>
<th>Operation</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Parenthetical expressions</td>
<td>(...)</td>
</tr>
<tr>
<td>2.</td>
<td>Functions</td>
<td>-</td>
</tr>
<tr>
<td>3.</td>
<td>Exponentiation</td>
<td>^</td>
</tr>
<tr>
<td>4.</td>
<td>Multiplication or Division</td>
<td>* or /</td>
</tr>
<tr>
<td>5.</td>
<td>Addition or Subtraction</td>
<td>+ or -</td>
</tr>
</tbody>
</table>

Examples:

\[
\begin{align*}
a &= 3. \\
b &= 14/3.45 \\
c &= f + 1.03e-4*a/b \\
p &= 4 * \text{atan}(1) \\
c1 &= a + 3.23/bb \\
i &= 4 * \cos(a+b)
\end{align*}
\]
Remarks

- The macro `para` replaces the macro `cons`.
- The macro `para` can be used also as macro command `para`.
- Note that the expressions do not have more than 75 characters and contain not more than 32 entries (+,-,numbers,parameter) for a correct input!
- Note that internal computations are all performed in double precision arithmetic (e.g., as REAL*8 variables).
- At the present time only one level of parenthesis may appear in any expression.
- Solve the above problems by a repetition of the definitions.
  Example: `q1 = tan(1./(3.+aa))` is not a legal expression at the present time. It should be replaced by the pair of statements
  
  \[
  q1 = 1./(3.+aa) \\
  q1 = \tan(q1)
  \]
- As a consequence of these repetition parameters may have their values redefined as many times as needed by using the `para` data command followed by other commands and data using the values of assigned parameters.
- In interactive mode of execution, the current set of parameters may be output by entering `list` while in parameter input mode. After listing input of additional parameters may be continued.
- It is possible to use expressions containing the parameters in any input mode which uses the utility input routine `dinput`. Users who write their own programs should try to use this routine to perform all inputs – `dinput` will input from either the input file or from the keyboard, depending on which is the active input unit.

Consequences for all FEAP input data

The most powerful form of data input in FEAP is the use of parameters as described above. Once defined by `para` they can be used at any place of the data input.

Furthermore expressions containing of constants, parameters and functions are allowed at any place of input.

Accordingly, a load may be assigned as

```
load 1.,a/12. + 3. 
```

is permitted. Another example are the node and element numbers. After defining `n,e,m,x,y,p` by `para` the following input is possible

```
bloc  
4,4,4.n,e,m  
1.0.+x,0.+y  
2.5.+x,0.+y  
3.5.+x,5.+y  
4.0.+x,5.+y 
coor 
```
n+25,0,5.5+x,2.5+y

**elem**
e+16,p,n+14,n+15

to input a block of nodes and elements. Note that **here** the expressions do not have more than 10 characters to have a correct input!
pars allows all input data as expressions. Thus parameters could be used in input data. pars is set automatically by para. The opposite command is nopa which allows all input data only numerical.
POIN

\begin{verbatim}
poin
  ( xp(i),i=1,ndm),<fact>*
  ( fp(i),i=1,ndf)
  (bcp(i),i=1,ndf)
<etc., terminate with a blank record>
\end{verbatim}

At a point with coordinates xp loads and boundary conditions can be set with \texttt{poin}.
FEAP is searching for points within a certain circle around the point with coordinates xp.
In case of a very fine discretization, it may occur that more than one point will be found. In that case a warning occur during the data input process of FEAP. With a value of 'fact' (default=1) larger than 1 the radius of the circle can be reduced.

Remarks:

- * Input for a point with \texttt{xp=0}: fact have to be set explicitly!
- Loads fp and bc-conditions bcp are 'added'.
- Nodes have to be defined before.
- Be careful in case of tied nodes to avoid multiple loads, see macro \texttt{tie}.

Example: For a load F at coordinate \texttt{(x,y,z)} \texttt{poin} has found 3 nodes A,B,C. \texttt{tie} set the sum of loads (=3F) on each node and skip nodes B,C. Thus, as a wrong result node A has a load of 3F!
As a solution the macro \texttt{poin} has to be used with a load F/(no. of nodes to be tied).
- Input possible only for ndf $\leq$ 16.
The **pola** command may be used to convert any coordinates \((r, \theta)\) which have been specified in polar (or cylindrical) form, to cartesian coordinates \(x_i, x_j\). The conversion is performed using the table below where 'ntype' denotes the coordinates \(i\) and \(j\) for which the conversion takes place. 'ntype = 0' is the default value for two-dimensional meshes.

<table>
<thead>
<tr>
<th>ntype</th>
<th>(x_1 - x_2)-plane</th>
<th>(x_1 - x_3)-plane</th>
<th>(x_2 - x_3)-plane</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(x(1,\text{node}))</td>
<td>(x(1,\text{node}))</td>
<td>(x(2,\text{node}))</td>
</tr>
<tr>
<td>(r)</td>
<td>(x(2,\text{node}))</td>
<td>(x(3,\text{node}))</td>
<td>(x(3,\text{node}))</td>
</tr>
<tr>
<td>(\theta)</td>
<td>(x_{01} + r \cdot \cos(\theta))</td>
<td>(x_{01} + r \cdot \cos(\theta))</td>
<td>(x_{02} + r \cdot \sin(\theta))</td>
</tr>
<tr>
<td>(x(1,\text{node}))</td>
<td>(x_{02} + r \cdot \sin(\theta))</td>
<td>(x_{03} + r \cdot \sin(\theta))</td>
<td>(x_{03} + r \cdot \sin(\theta))</td>
</tr>
</tbody>
</table>

As can be seen from the equations in the table above, \(x_{0i}\) and \(x_{0j}\) are the cartesian coordinates of the origin of the polar coordinate system.

A sequence of nodes may be converted by specifying non-zero values for 'node1', 'node2', and 'inc'. The sequence generated will be:

```
node1, node1+inc, node1+2*inc, ... , node2
```

Several records may follow the **pola** command. Execution terminates with a blank record.
PRES

The **pres** command is used to specify the values for loads at segments. For each segment to be specified a record is entered with the following information:

<table>
<thead>
<tr>
<th>segm</th>
<th>the number of segment to be specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1(1,segm)</td>
<td>value of load for 1–dof at 'node 1'</td>
</tr>
<tr>
<td>p1(2,segm)</td>
<td>value of load for 2–dof at 'node 1'</td>
</tr>
<tr>
<td></td>
<td>etc. until to 'ndf' directions</td>
</tr>
<tr>
<td>p2(1,segm)</td>
<td>value of load for 1–dof at 'node 1'</td>
</tr>
<tr>
<td>p2(2,segm)</td>
<td>value of load for 2–dof at 'node 2'</td>
</tr>
<tr>
<td></td>
<td>etc. until to 'ndf' directions</td>
</tr>
</tbody>
</table>

Remarks:

1. A trapezoidal load is defined from the nodal values.
2. Loads are acting in the direction of defined dofs of freedom.
3. The reference length is the real length of the segment.
4. The load case warping is defined by p1(1,1)=−999.
5. For mesh generation with NEGE the following order of statements is necessary: **nege**, **[neco]**, **[back]**, **geom**, **segm**, **regi**, **[pres]**, **[fixe]**, further FEAP-macros. Note: macros in brackets are optional.
PRIN

prin

The use of the prin macro will cause the description of all information produced during MESH description to be written on the output file. The use of nopr will discontinue the output of mesh information (except for data printed in elements). The default value is prin.
With \texttt{qloa} up to 10 load parameters are set which act on the specified elements associated with material number \( ma \). Elements are generated from element \( a \) until element \( b \) with increment \( incr \). This input could be be repeated until all elements are defined. Each element could be loaded more than once (if this makes sense!).

This macro could be used only \textbf{after} the definition of elements and materials! The definition and use of the load parameters depend on the used element and can be found in the element description:

\begin{verbatim}
  elmt01 : -
  elmt02 : ma, q1, q2, n1, n2, if ol, T_u, T_o
  elmt03 : ma, q1, q2, n1, n2
  elmt04 : ma, qzL, qyL, qzL, qzG, qyG, qzG
  elmt05 : ma, b1, b2
  elmt06 : -
  elmt07 : ma, q, \Delta T
  elmt08 : -
  elmt09 : ma, b1, b2, b3, bx, by, bz
  elmt10 : -
  elmt11 : ma, qz, qy, qz
  elmt12 : -
  elmt13 : ma, q
  elmt14 : -
  elmt15 : ma, yp, zp, qx, qy, qz
  elmt16 : -
  elmt17 : ma, q
  elmt18 : ma, q
  elmt19 : ma, q
  elmt20 : ma, q1, q2
  elmt21 : ma, qz, qy, qz, \Delta T
  elmt22 : ma, qz, qy
  elmt30 : ma, q1, q2, q3, ltyp
  elmt45 : ma, qz, qy, qz,
\end{verbatim}

For details please check the element library.

An associated load vector will be calculated on element level (\texttt{isw=22}). In nonlinear cases the actual load factor \texttt{prop} is used. Nonlinear effects like e.g. dependency on displacements or temperature (modification of stiffness matrix) are possible if coded on element level. In such cases the macro \texttt{sloa} could also be used.
RBOU

rbou

\( r_{\text{min}}, r_{\text{max}}, \phi_{\text{min}}, \phi_{\text{max}}, < z_{\text{min}}, z_{\text{max}} >, (ibc(j), j=1, ndf) \)

<etc., terminate with a blank record>

This macro is used if coordinates are given in cartesian directions. Otherwise use macro vbou
The boundary restraint conditions may be set in a two/three dimensional region which is defined by

\[
\begin{array}{c}
 r_{\text{min}} \leq r \leq r_{\text{max}} \\
 \phi_{\text{min}} \leq \phi \leq \phi_{\text{max}} \\
 < z_{\text{min}} \leq z \leq z_{\text{max}} >
\end{array}
\]

The data to be supplied during the definition of the mesh (or in macro execution using the 'mesh' command) consists of:

\[
\begin{array}{l}
 r_{\text{min}} \quad - \quad \text{minimum value of radius } r = \sqrt{x^2 + y^2} \\
 r_{\text{max}} \quad - \quad \text{maximum value of radius } r = \sqrt{x^2 + y^2} \\
 \phi_{\text{min}} \quad - \quad \text{minimum value of angle } \phi \text{ in degree with } \phi_{\text{min}} \geq 0 \\
 \phi_{\text{max}} \quad - \quad \text{maximum value of angle } \phi \text{ in degree with } \phi_{\text{max}} \geq 0 \\
 < z_{\text{min}} \quad - \quad \text{minimum value of coordinate } 3=z > \\
 < z_{\text{max}} \quad - \quad \text{maximum value of coordinate } 3=z > \\
 ibc(1) \\
 ibc(2) \quad \text{restraint conditions for all nodes with the} \\
 \ldots \quad \text{value of the search (0 = active dof).} \\
 ibc(ndf) \quad > 0 \text{ or } < 0 \text{ denotes a fixed dof).}
\end{array}
\]

Remarks:

1. Up to 16 values and up to 80 characters are possible on each input record.

2. boun sets the boundary conditions whereas ebou, vbou, gbou and rbou adds boundary conditions to existing values.

3. There is no tolerance in the defined block values. Thus nodes defined by block may lay not in the block. In this case modify the max and min values.

For the specification of prescribed displacements within a calculation, see macro command disp.
RNDM

rndm
val, ntype
node1, ngen1
node2, ngen2
<etc., terminate with blank record>

The **rndm** command is used to modify the values for nodal coordinates with a random parameter \((0 < \text{rndm} < 1)\) times a given value.

\[ X = X + 2 \times (\text{rndm}-0.5) \times \text{val} \]

For each node to be specified a record is entered with the following information:

<table>
<thead>
<tr>
<th>val</th>
<th>value to be added on coordinates by function random</th>
</tr>
</thead>
</table>
| ntype | - 1 - produce every time different random numbers (def.)  
  | - 2 - produce every time same random numbers |
| node1 | the number of first node to be specified |
| node2 | the number of last node to be specified |
| ngen | the increment to the next node, if generation is used, otherwise 0 |

When generation is performed, the node number sequence will be (for node1-node2 sequence shown above):

node1, node1+ngen1, node1+2*ngen1, ..., node2

Be careful with the use of this macro. Due to the modification of the coordinates macros like **eloa**, **edge** or **tie** may not work. Thus use for example **eloa** before **rndm**.
RSUM

The `rsum` command is used to specify the values for a sum of nodal reactions for the degree of freedom 'dofrsum'. **Alternatively** 2 options are possible

- **rstyp=1**(default)
  For each node to be specified a record is entered with the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>node1</td>
<td>the first number of node to be specified</td>
</tr>
<tr>
<td>node2</td>
<td>the last number of node to be specified</td>
</tr>
<tr>
<td>ngen</td>
<td>the increment to the next node</td>
</tr>
</tbody>
</table>

  node2 and ngen1 are optional.

  When generation is performed, the node number sequence will be (for node1-node2 sequence shown above):

  node1, node1+ngen1, node1+2*ngen1, .... , node2

  The intermediate nodes will be a linear interpolation between 'node1' and 'node2'.

- **rstyp=2**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>i-dir</td>
<td>the direction of the coordinate (i.e., 1 = x, 2 = y, etc. to ndm) to be searched.</td>
</tr>
<tr>
<td>xi-value</td>
<td>the value of the i-direction coordinate to be used during the search (a tolerance of about $1/1000 \cdot \sqrt{numnp}$ of the mesh size is used during the search.)</td>
</tr>
</tbody>
</table>

A graphical control is possible using the macro `rsum` in Plot-modus.

The resulting reaction force R is always printed using the macro `reac`. When using the macro `tplo` this resulting force R is plotted versus displacement or time in case of reac or reat.
The `regi` command is used to specify the segments which are attached to each region. For each region to be specified, a record is entered with the following information:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>regi1</code></td>
<td>the number of region to be specified.</td>
</tr>
<tr>
<td><code>nregi1</code></td>
<td>the number of segments for the region to be specified.</td>
</tr>
<tr>
<td><code>segm1(1)</code></td>
<td>segment 1 attached to region regi.</td>
</tr>
<tr>
<td><code>segm1(2)</code></td>
<td>segment 2 attached to region regi.</td>
</tr>
</tbody>
</table>

Remarks:

1. Each region has its own material. If a segment is used against its defined direction, this segment has to be marked by a Minus sign. Each segment must occur for the first time positive! Thus, modify the segment definition if necessary.

2. Regions are defined as a number of sections along its boundaries. **Outside**–boundaries have to be defined **counterclockwise**, whereas **Inside**–boundaries have to be defined **clockwise**!

3. There must not be a direct connection between segments.

4. Up to 16 values are possible on each input record, then use continuation line.

5. For mesh generation with NEGE the following order of statements is necessary: `nege` , `[neco]`, `[back]`, `geom`, `segm`, `regi`, `[pres]`, `[fixe]`, further FEAP-macros. Note: macros in brackets are optional.
ROT

\texttt{rot node1,node2,inc,iax,ang}<\text{terminate with blank record}>

The \texttt{rot} command may be used to change cartesian coordinates by a defined rotation \texttt{ang(deg)} around axis \texttt{iax}.

\[ x(i, node) = T(iax, ang) \cdot x(i, node) \]

A sequence of nodes may be converted by specifying non-zero values for \texttt{node1}, \texttt{node2}, and \texttt{inc}. The sequence generated will be:

\[ \text{node1, node1+inc, node1+2*inc, \ldots, node2} \]

Several records may follow the \texttt{rot} command. Execution terminates with a blank record.

A translation of coordinates is possible with \texttt{tran}. 
The **segm** command is used to specify the values of nodal numbers which are attached to each segment. For each segment to be specified, a record is entered with the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>segm</strong></td>
<td>the number of the segment to be specified.</td>
</tr>
<tr>
<td><strong>ix(1,segm)</strong></td>
<td>node-1 number attached to segment.</td>
</tr>
<tr>
<td><strong>ix(2,segm)</strong></td>
<td>node-2 number attached to segment.</td>
</tr>
</tbody>
</table>

only for curved segments:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>mtyp</strong></td>
<td>2 = circle (midpoint ( P_x, P_y ), radius ( R = R_x )).</td>
</tr>
<tr>
<td><strong>g(1,segm)</strong></td>
<td>( P_x ).</td>
</tr>
<tr>
<td><strong>g(2,segm)</strong></td>
<td>( P_y ).</td>
</tr>
<tr>
<td><strong>g(3,segm)</strong></td>
<td>( R_x ).</td>
</tr>
<tr>
<td><strong>g(4,segm)</strong></td>
<td>( R_y ) (not used for mtyp=3).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>mtyp</strong></td>
<td>6 = circle like 2, but automatically finding of coordinates of end points of the segment and of midpoint of the circle. Assume for this purpose in <strong>geom</strong> that both nodes have the coordinates of the intersection of the segments before and after the circle.</td>
</tr>
<tr>
<td><strong>g(1,segm)</strong></td>
<td>( P_1 ) node number on segment before circle.</td>
</tr>
<tr>
<td><strong>g(2,segm)</strong></td>
<td>( P_2 ) node number on segment after circle.</td>
</tr>
<tr>
<td><strong>g(3,segm)</strong></td>
<td>( R ) radius of circle.</td>
</tr>
</tbody>
</table>

**Remarks:**

1. Each segment has a ‘direction’ from node 1 to node 2. This has to be taken into account in the definition of regions. See the macro **regi**.

2. For mesh generation with NEGE the following order of statements is necessary: **nege**, **[neco]**, **[back]**, **geom**, **segm**, **regi**, **[pres]**, **[fixe]**, further FEAP-macros. Note: macros in brackets are optional.
SLOA

The `sloa` command is used to specify the values for surface loading quantities. Only traction quantities are considered (e.g., no surface displacement distributions may be specified by `sloa`). The nodal values for the loads are determined by each element subprogram (i.e., in `ELMTnn` with the isw = 7. Here for example the associated load vector and the load stiffness matrix have to be computed).

Datas are specified as follows:

<table>
<thead>
<tr>
<th>iel</th>
<th>element subprogram which generates surface loads (only one routine may be given for a problem). ⇒ FEAP element number</th>
</tr>
</thead>
<tbody>
<tr>
<td>ns</td>
<td>number of nodes on surface of element.</td>
</tr>
<tr>
<td>nv</td>
<td>number of parameters defining distributed loading.</td>
</tr>
<tr>
<td>nl</td>
<td>loading type (generally only one type is currently included in elements and 'nl' is ignored – default may be 0).</td>
</tr>
</tbody>
</table>

Next input for all loaded elements

<table>
<thead>
<tr>
<th>ixl(i)</th>
<th>list of nodes on element surface.</th>
</tr>
</thead>
<tbody>
<tr>
<td>p(i)</td>
<td>list of parameters defining loading.</td>
</tr>
<tr>
<td>ngen</td>
<td>increment for node generation (default 0).</td>
</tr>
<tr>
<td>lgen</td>
<td>number of repetitions (default 0).</td>
</tr>
</tbody>
</table>

A maximum of 8 items can appear on each record. If more than 8 items are required continue on the next record.
Choose an appropriate solver for FEAP solutions by defining 'n0'. For the iterative solvers input of 'n1'-'n4' is possible. Otherwise the default values are be used. These values could also be changed using the macros pbeg and pgmr in macro-mode.

<table>
<thead>
<tr>
<th>n0</th>
<th>solver</th>
<th>storage technique</th>
<th>n1</th>
<th>n2</th>
<th>n3</th>
<th>n4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>sym/un sym.</td>
<td>standard profile [default]</td>
<td>columnheight</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>symmetric</td>
<td>sparse matrix (SM)(-min.deg)</td>
<td>CSR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>symmetric</td>
<td>sparse matrix (SM)(+min.deg)</td>
<td>CSR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>sym/un symn.</td>
<td>SUPER-LU (sequential)</td>
<td>CSR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>sym/un sym.</td>
<td>PARDISO direct (parallel)</td>
<td>CSR</td>
<td>isym</td>
<td>nproc</td>
<td>icgs</td>
</tr>
<tr>
<td>5</td>
<td>symmetric</td>
<td>HSL MA86 (parallel)</td>
<td>CSR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>sym/un sym.</td>
<td>PBCG</td>
<td>CSR</td>
<td>niter</td>
<td>tol</td>
<td>itol</td>
</tr>
<tr>
<td>7</td>
<td>sym/un sym.</td>
<td>PGMRES</td>
<td>CSR</td>
<td>niter</td>
<td>tol</td>
<td>im</td>
</tr>
<tr>
<td>8</td>
<td>sym/un sym.</td>
<td>PGMRES</td>
<td>CSR</td>
<td>niter</td>
<td>tol</td>
<td>im</td>
</tr>
<tr>
<td>9</td>
<td>symmetric</td>
<td>PARDISO ML (sequential)</td>
<td>CSR</td>
<td>niter</td>
<td>tol</td>
<td>impro</td>
</tr>
<tr>
<td>10</td>
<td>Simplex optimization solver</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Solver 0** is the standard solver and used for small problems (small neq) and allows symmetric and un-symmetric stiffness matrices based on a Gaussian elimination with a \( L^TDL \) decomposition. A column-height technique is used, thus the efficiency of the solver depends strongly on the nodal and element numbering. The speed of the solution process can be improved using a node number optimization with opti-macro.

Input on MESH-level is nothing or **solv-0**

The other solvers are advantageous to solve large systems of equations and do not depend on node and element numbering.

For **symmetric problems** the sparse matrix solver (SM) (n0=1,2), the Pardiso solver (n0=4) (only INTEL) or the HSL MA86 solver (n0=5) (only INTEL) are preferable and very fast. Here a \( L^TDL \) decomposition is used. When storing the stiffness matrix only non zero entries are taken into account. This is called a compressed sparse row format (CSR). In addition the Pardiso and HSL MA86 solvers are running parallel. The Pardiso solver deals furthermore with **unsymmetric problems**.

For very large systems direct solvers could not compete with iterative solvers. For this purpose two solvers are implemented. These solvers do not need a triangular decomposition, which is the most time consuming part in a solution process of a large system of equations. As a consequence the system of equations is not solved directly but in a set of iterations. The number of iterations depends strongly on the preconditioning (prco) and can be in the range of 10 to 2000. Furthermore - to be efficient - a CSR sparse storage procedure is used while calculating the stiffness matrix. These solvers could be used for **symmetric and un-symmetric** matrices without any time differences!

- **Solver 1** is the direct sparse matrix solver (SM), which could be used for symmetric problems. Reordering is without minimum degree. Typically here solver 2 is used.

Input on MESH-level is **solv-1**
**Solver 2** is the direct sparse matrix solver (SM), which could be used for symmetric problems. Reordering is with minimum degree.

Input on MESH-level is `solv-2`

**Solver 3** is the direct SUPER-LU solver. This is a symmetric and unsymmetric sparse matrix solver and base on a LU decomposition and is available only for the SALFORD-Version.

Input on MESH-level is `solv-3`

**Solver 4** is the PARDISO direct parallel solver. This is a symmetric and unsymmetric sparse matrix solver. The solver is available only for the INTEL-Version.

Input on MESH-level is `solv-4` and

- The parameter `n1` = `isym` (Def = 1) sets the type of matrix: 1 = symmetric, 2 = unsymmetric.
- The parameter `n2` = `nproc` (Def = all, 1 = sequential) sets the number of processors.
- The parameter `n4` = `icgs` (Def = 0) defines computation of \( K \). For `icgs=0` always LU computation is done. Otherwise the computation of LU is replaced by CGS iteration, if there are nearly no changes, see PARDISO-parameter `iparm(4)`. This is faster but may lead sometimes to convergence problems.

**Solver 5** is the HSL MA86 direct parallel solver. This is a symmetric sparse matrix solver. Reordering is based on METIS-package. The solver is available only for the INTEL-Version.

Input on MESH-level is `solv-5`

**Solver 6** is based on the Pre-conditioned bi-Conjugated Gradient method (PBCG).

Input on MESH-level is `solv-6` and

- The parameter `n1` = `niter` (Def = 150) denotes the maximum numbers of iterations. Typical values are in the range of 50-1000.
- The parameter `n2` = `tol` (Def = \( 1 \cdot 10^{-08} \)) sets the tolerance value.
- The parameter `n3` = `itol` (Def = 1) sets the type of tolerance criterion (1-4).
- The parameter `n4` = `isym` (Def = 1) sets the type of matrix: 1 = symmetric, 2 = unsymmetric.

Input on MACRO-level is once `prco`. Here three versions of preconditioning are available. Input on MACRO-level can be `pbcg,n1,n2,n3` to reset parameters `n1`, `n2`, `n3`.

**Solver 7** is based on the Pre-conditioned Generalized Minimum Residual Method (PGMRES) and should be more robust but needs more work space for the iterative solution.

Input on MESH-level is `solv-7` and

- The parameter `n1` = `niter` (Def = 150) denotes the maximum numbers of iterations. Typical values are in the range of 10-500.
- The parameter `n2` = `tol` (Def = \( 1 \cdot 10^{-08} \)) sets the tolerance value.
- The parameter `n3` = `im` (Def = 50) sets the number of Krylov iteration vectors. Typical values are in the range of 50-100.
The parameter 'n4' = 'isym' (Def = 1) sets the type of matrix: 1 = symmetric, 2 = unsymmetric.
Parameters 'n1', 'n2' can be reset with the macro `pgmr,n1,n2`.

Input on MACRO-level is once `prco`. Here three versions of preconditioning are available.
Input on MACRO-level can be `pgmr,n1,n2` to reset parameters 'n1', 'n2'.

- **Solver 8** is an alternative version of the PGMRES solver.
- **Solver 9** is the PARDISO ML (multi-level) solver. This is a symmetric iterative sequential solver. The solver package for the LU decomposition use its own storage management. For storage efficiency the CSR-storage technique is used. The solver is available only for the INTEL-Version and PARDISO-Basel.

Input on MESH-level is `solv-9` and

- The parameter 'n1' = 'niter' (Def = 300) denotes the maximum numbers of iterations. Typical values are in the range of 50-1000.
- The parameter 'n2' = 'tol' (Def = $1 \cdot 10^{-06}$) sets the tolerance value.
- The parameter 'n3' = 'impro' (Def = 25) sets the maximum numbers of iterations without improvement. Then the solver will stop with error 101.

- **Solver 10** is an optimization solver based on the SIMPLEX-algorithm.

- **Remarks:**
  1. Do not use `opti` together with SM/SUPERLU/PARDISO/PBCG/PGMRES-solvers.
  2. Solvers 1,2,5,9 need symmetric matrices, solvers 0,3,4,6,7,8 allow also un-symmetric matrices.
  3. Recommendations: use for symmetric small problems solver 0 with `opti` or for large systems solvers 2,3,4,5 and for un-symmetric problems solver 0 with `opti` or for large systems solvers 3,4. In case of 3d-structures use always for large problems the iterative solvers 6,7,8,9 or the parallel solvers 4,5.
  4. solvers 1-9 not allowed with `cont`, `ext`ended system,
  5. calculation of $\text{det} K_T$ and negative diagonals

<table>
<thead>
<tr>
<th>n0</th>
<th>solver</th>
<th>$\text{det} K_T$</th>
<th>neg. diags</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>sym/unsym.</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>1</td>
<td>symmetric</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>2</td>
<td>symmetric</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>3</td>
<td>sym/unsym. SUPER-LU (sequential)</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>4</td>
<td>sym/unsym. PARDISO (parallel)</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td>5</td>
<td>sym/unsym. HSL MA86 (parallel)</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>6</td>
<td>sym/unsym. PBCG (iterative)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>sym/unsym. PGMRES (iterative)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>sym/unsym. PGMRES (iterative)</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

8. Infos HSL: [http://www.hsl.rl.ac.uk/](http://www.hsl.rl.ac.uk/)
The `sphe` command may be used to convert any coordinates \((r, \theta, \beta)\) which have been specified in spherical form, to cartesian coordinates \((x_1, x_2, x_3)\). The conversion is performed using the following relations:

<table>
<thead>
<tr>
<th>(r)</th>
<th>(x(1,\text{node}))</th>
<th>input value of radius</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\theta)</td>
<td>(x(2,\text{node}))</td>
<td>input value of angle in degrees</td>
</tr>
<tr>
<td>(\beta)</td>
<td>(x(3,\text{node}))</td>
<td>input value of angle in degrees</td>
</tr>
</tbody>
</table>

\[
x(1, \text{node}) = x_0 + r \cdot \cos(\theta) \cdot \sin(\beta) \\
x(2, \text{node}) = x_0 + r \cdot \sin(\theta) \cdot \sin(\beta) \\
x(3, \text{node}) = x_0 + r \cdot \cos(\theta)
\]

Here \(x_0\), \(x_0\), and \(x_0\) are the cartesian coordinates of the origin of the spherical coordinate system.

A sequence of nodes may be converted by specifying non-zero values for `node1`, `node2`, and `inc`. The sequence generated will be:

```
node1, node1+inc, node1+2*inc, ..., node2
```

Several records may follow the `sphe` command. Execution terminates with a blank record.
The last command in the mesh input must be **end**. Then special actions like searching for double nodes with **tie** or linking conditions with **link** are done. It follows the execution of the program, which is controlled by **macro** commands. The execution of these commands can be done in batch-modus **macro** or in interactive modus **inte**. In the first case the commands are defined in the input-file after the **macro** command, in the second case the commands are defined interactively.

**stop** is the final command in the input file, which stops the execution.
temp
node1, ngen1, t(node1)
node2, ngen2, t(node2)
<etc., terminate with blank record>

The **temp** command is used to specify the values for nodal temperatures. For each node to be specified a record is entered with the following information:

<table>
<thead>
<tr>
<th>node</th>
<th>the number of the node to be specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>ngen</td>
<td>the increment to the next node, if generation is used, otherwise 0.</td>
</tr>
<tr>
<td>t(node)</td>
<td>value of temperature for 'node'.</td>
</tr>
</tbody>
</table>

When generation is performed, the node number sequence will be (for node1–node2 sequence shown above):

node1, node1+ngen1, node1+2*ngen1, ..., node2

The values for each temperature will be a linear interpolation between t(node1) and t(node2).
TIE

tie, <tol>
tie,all <tol>
tie,node,<tol>,n1,n2
tie,mate,<tol>,n1,n2
tie,dir,<tol>,n1,n2
tie,line,<tol> and continuation line: x1,y2,<z1>,x2,y2,<z2>
tie,mlin,<tol>,n1,n2 and continuation line: x1,y2,<z1>,x2,y2,<z2>
tie,prin

tie,nopr

A mesh may be generated in FEAP in which are more than one node with the same coordinates. The tie command may be used to “connect” these nodes so that the same values of the solution will be produced at all nodes with the same initial coordinates. Repetitions of the command are allowed to define e.g. the connection of nodes on different lines.

Special options are possible to connect only nodes on a line, nodes in a direction etc.

With 'tol' a user defined tolerance for tie could be set. The tolerance is defined as $|x_{\text{max}} - x_{\text{min}}| \cdot 'tol$.

The default value for 'tol' is $10^{-3} \sqrt{\text{numnp}}$.

With tie,node, the tie command acts only on nodes between node numbers 'n1' and 'n2'.

With tie,mate, the tie command acts only on nodes of elements with material numbers 'n1' and 'n2'.

With tie,dir, the tie command acts only on nodes which have the coordinate 'n2' in direction 'n1'.

With tie,line, the tie command acts on a line between points 1 and 2. Input data for the line must follow directly on a continuation line with: x1,y2,<z1>,x2,y2,<z2>.

With tie,mlin, the tie command acts on a line between points 1 and 2 but only for materials 'n1' and 'n2'. Input data for the line must follow directly on a continuation line with: x1,y2,<z1>,x2,y2,<z2>.

With tie,prin, all tie output active. The macro acts like prin.

With tie,nopr, all tie output is suppressed. The macro acts like nopr.

Comment: For the summary of input data – as a final result of the meshing procedure – on the screen it is necessary that prin is activated.

To use the tie option the complete mesh must be defined first. After the end command for the mesh definition and before the macro or interactive command for defining a solution algorithm, the use of a tie statement will cause the program to search for all coordinates that are to be connected together.

Remarks:

1. When nodes are connected any specified, restrained boundary condition (and value) will be assigned to all interconnected nodes. Thus, it is only necessary to specify restrained boundary conditions (and values) for one of the nodes.

2. If tied nodes are loaded then the sum of all loads is set on all nodes. This can occur e.g. for aloa and poin.

   $F_i = F_1, \quad F_k = F_2 \rightarrow \text{loads are added and set:} \quad F_i = F_1 + F_2, \quad F_k = F_1 + F_2$

3. Tied nodes are printed when starting FEAP. A further control is possible in plot modus with macro tie.
TRAN

\textbf{TRAN}
\begin{verbatim}
tran
node1,node2,inc,dx1, dx2, dx3
<br terminate with blank record>
\end{verbatim}

The \texttt{tran} command may be used to change cartesian coordinates by a defined translation.

\[ x(i, node) = x(i, node) - dx_i \]

A sequence of nodes may be converted by specifying non-zero values for node1, node2, and inc. The sequence generated will be:

\begin{verbatim}
node1, node1+inc, node1+2*inc, ... , node2
\end{verbatim}

Several records may follow the \texttt{tran} command. Execution terminates with a blank record.

A rotation around a certain axis is possible with \texttt{rot}. 
The `trib` data input segment is used to generate a regular triangular patch consisting of triangular elements for two dimensional patches or three dimensional surfaces.

The patch of nodes/elements defined by `trib` is developed from a master triangle which is defined by an isoparametric 3-node mapping function in terms of the natural coordinates.

The spacing between the `r`-increments is equal.

Patches may be interconnected. This can be done by using the `tie` macro to 'connect' any nodes which have the same coordinates. The data parameters are defined as:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodes</td>
<td>number of master nodes needed to define the patch.</td>
</tr>
<tr>
<td><code>r-inc</code></td>
<td>number of nodal increments to be generated along <code>r</code>-direction and <code>s</code>-direction of the patch.</td>
</tr>
<tr>
<td><code>node1</code></td>
<td>number to be assigned to first generated node in patch (default = 1). First node is located at same location as master node 1. Last generated node (i.e., <code>node1 - 1 + (r-inc * (r-inc +1)/2)</code>) is located at same location as master node 3.</td>
</tr>
<tr>
<td><code>elmt1</code></td>
<td>number to be assigned to first element generated in patch; if zero no elements are generated (default = 0)</td>
</tr>
<tr>
<td><code>matl</code></td>
<td>material number to be assigned to all generated elements in patch (default = 1)</td>
</tr>
</tbody>
</table>

The generated number of elements is: `numel = r^2`. The generated number of nodes is: `numnp = (r + 1) * (r + 2)/2`. 
VANG

**VANG**

**vang**

`vang`, `angl1`, `angl2`, `itrot`, `inpc`

`x_{1min}, x_{1max}, x_{2min}, x_{2max}, < x_{3min}, x_{3max} >`

<etc., terminate with a blank record>

**VANG** define a local base system in a region. Angles may be set for the following directions:

- `angl1` – Number of first axis of rotated basis (default=1)
- `angl2` – Number of second axis of rotated basis (default=2)
- `itrot` – 0/1 with/without rot. dofs 4-6 (default=0)
- `inpc` – 1=cartesian input calculate $\phi$ (default=1)
- `inpc` – 2=cartesian input calculate $\phi - 90$
- `inpc` – 3=polar input calculate $\phi$
- `inpc` – 4=polar input calculate $\phi - 90$

in a two/three dimensional region which is defined by:

\[
\begin{align*}
&x_{1min} \leq x_1 \leq x_{1max} \\
&x_{2min} \leq x_2 \leq x_{2max} \\
&< x_{3min} \leq x_3 \leq x_{3max} >
\end{align*}
\]

**Remarks:**

1. Only one choice of axis `angl1`, `angl2` and `itrot` is possible.
2. A transformation will be done for dofs `angl1` and `angl2`.
3. In case of ndf=6 the same transformation is chosen for dofs `angl1+3` and `angl2+3` for `itrot=0`.
4. The associated dofs of the used element can be chosen via the macro `mate`, parameters `idfg`. As example the angles of the DKQ-plate element can be transformed via `mate 1,7,2,3,1`
5. Results are printed and plotted for nodes with `angl \neq 0` in directions `angl1,angl2`. Thus mixed vectors occur.
6. Results which base on a smoothing procedure are plotted in global directions. Thus results for nodes with `angl \neq 0` are transformed to global directions. Ex.: `disp, resi, eigv, ...`
The boundary restraint conditions may be set in a two/three dimensional region which is defined by

\[
\begin{align*}
&x_{1\text{min}} \leq x_1 \leq x_{1\text{max}} \\
&x_{2\text{min}} \leq x_2 \leq x_{2\text{max}} \\
&x_{3\text{min}} \leq x_3 \leq x_{3\text{max}} 
\end{align*}
\]

The data to be supplied during the definition of the mesh (or in macro execution using the 'mesh' command) consists of:

\[
\begin{align*}
&x_{1\text{min}} &- &\text{minimum value of coordinate 1} \\
&x_{1\text{max}} &- &\text{maximum value of coordinate 1} \\
&x_{2\text{min}} &- &\text{minimum value of coordinate 2} \\
&x_{2\text{max}} &- &\text{maximum value of coordinate 2} \\
&x_{3\text{min}} &- &\text{minimum value of coordinate 3 } > \\
&x_{3\text{max}} &- &\text{maximum value of coordinate 3 } > \\
&\text{ibc(1)} &- &\text{restraint conditions for all nodes with the} \\
&\text{ibc(2)} &- &\text{value of the search (0 = active dof).} \\
&\text{ibc(ndf)} &- &> 0 \text{ or } < 0 \text{ denotes a fixed dof).}
\end{align*}
\]

Remarks:

1. Up to 16 values and up to 80 characters are possible on each input record.
2. \textbf{boun} sets the boundary conditions whereas \textbf{ebou} and \textbf{vbou} adds boundary conditions to existing values.
3. There is no tolerance in the defined block values. Thus nodes defined by block may lay not in the block. In this case modify the max and min values.
4. Coordinates $x_i$ may be defined in any coordinate system.
**YBOU**

| ybou | $a_1, a_2, (a_3), b_1, b_2, (b_3), bc, m_{pl,l}, m_{pl,u}$ | <etc., terminate with a blank record> |

**Ybou** defines boundary constraints in FEAP for yield line calculations.

- Each boundary restraint condition is defined by a line between $P_a(a_1, a_2, (a_3))$ and $P_b(b_1, b_2, (b_3))$.
- 'bc' defines the boundary condition of the plate's edge. 'bc' = 0 signifies a free edge, 'bc' = 1 is a supported edge.
- 'm_{pl,l}' is the lower plastic moment, 'm_{pl,u}' is the upper plastic moment at the considered edge. I.e. the caused tension cracks will appear at the upper side of the plate when 'm_{pl,u}' is meant.

**Remarks:**

1. Additionally the usual degrees of freedom belonging to the boundary have to be given by the regular FEAP command **edge**. In this way the degrees of freedom for the optimization search will be directed.

2. Even when a free edge exists ('bc' = 0), plastic moments can be defined. In this way it is possible to map symmetrical problems being cut along the axis of symmetry.
This macro is used for mesh generation with CYLT and is similar to the macro para. All definitions which are defined by ycon hold for the mesh generation and FEAP.

**Remark:**
For mesh generation with CYLT the following order of statements is necessary: cylt, [ycon], ynod, yedg, further FEAP-macros (including ybou).
Note: macros in brackets are optional.
YEDG

**yedg**

```
yedg
edge1, node1, node2, ang
<etc., terminate with blank record>
```

The `yedg` command is used within mesh generation with CYLT to specify the values of nodal numbers which are attached to each edge of a yield-line plate. For each edge to be specified, a record is entered with the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>edge1</td>
<td>the number of the edge to be specified.</td>
</tr>
<tr>
<td>node1</td>
<td>node–1 number attached to edge.</td>
</tr>
<tr>
<td>node2</td>
<td>node–2 number attached to edge.</td>
</tr>
<tr>
<td>ang</td>
<td>rotation of plane:</td>
</tr>
<tr>
<td></td>
<td>’ang’ = 1 – angle is 45° for supported edges</td>
</tr>
<tr>
<td></td>
<td>’ang’ = 0 – angle is 90° for free edges</td>
</tr>
</tbody>
</table>

Remarks:

1. The angles defined by ’ang’ specify the rotation of the planes defining the lines of intersection and ruling the yield-lines, see CYLT-manual.
The **yloa** command is used to specify the values of loads on triangular yield-line elements for points, lines and areas.

- The first line per entry defines the nodes of the load, e.g. \(P_a(a_1, a_2), P_b(b_1, b_2)\) and \(P_c(c_1, c_2)\) defining a triangular area.
- The second line gives the extrapolated loads for each node.

**Remarks:**

1. The current definition is a preliminary version. The loads have to be extrapolated by the user.

2. As zero loads are not allowed, FEAP is able to check the zero load entries in the second line and defines the type of load:
   - \(P_a\) with \(q_a\) defines a single load,
   - \(P_a, P_b\) with \(q_a, q_b\) defines a linear load,
   - \(P_a, P_b, P_c\) with \(q_a, q_b, q_c\) defines an area load.

3. It must be assured that the triangulated yield-line mesh provides a node at the required load point.
YNOD

ynod
node1, (x(i,node1),i=1,2)
node2, (x(i,node2),i=1,2)
<etc., terminate with blank record>

The **ynod** command is used within mesh generation with CYLT to specify the values of nodal coordinates for edge nodes of yield-lines (using **yedg**). For each node to be specified a record is entered with the following information:

<table>
<thead>
<tr>
<th>node</th>
<th>the number of the node to be specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>x(1,node)</td>
<td>value of coordinate in 1–direction for ’node’</td>
</tr>
<tr>
<td>x(2,node)</td>
<td>value of coordinate in 2–direction for ’node’</td>
</tr>
</tbody>
</table>
Chapter 3

Macro Commands

3.1 Available Macros

The following entries \texttt{xxxx,yyyy,v1,v2,v3} are available for the control of the calculation: (where \texttt{xxxx} is selected from the following list):

\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|}
\hline
acce & arcl & augm & auto & back & bfgs & chec & cmas & cont & conv  \\
\hline
copy & crit & curv & damp & data & debu & dets & dibc & disp & dt  \\
\hline
dplo & eigi & eigk & eig1 & else & endi & epsq & erro & exit  \\
\hline
ext & fast & feas & form & four & fsol & fsum & geom & help & hist  \\
\hline
iden & if & iiimp & init & jint & lamb & lan & line & lmas & loop  \\
\hline
macn & man & mate & mesh & newf & next & nonl & nopr & para & parv  \\
\hline
pau & pbcg & pgmr & plot & pola & post & prin & proco & proc & prop  \\
\hline
quit & reac & read & reme & rest & rhin & rinp & save & show & sigq  \\
\hline
smoo & solv & splo & stre & subs & summ & tang & tec & time & tol  \\
\hline
\texttt{tplo} & \texttt{trans} & \texttt{ueig} & \texttt{umas} & \texttt{updh} & \texttt{utan} & \texttt{velo} & \texttt{writ} & \texttt{yang} & \texttt{yevo}  \\
\hline
ygra & ymsh & ytab & ytry &  \\
\hline
\end{tabular}

The values \texttt{v1,v2,v3} can be predefined parameters. This can be done e.g. on input-level via \texttt{para} or on macro-level via \texttt{para}. Furthermore calculations for \texttt{v1,v2,v3} are possible, each up to 15 characters. For details see the description how to define \texttt{parameter}.

A short overview on commands is given in section 3.2, see macro \textbf{Index} in FEAP, whereas possible actions can be found in section 3.3, see macro \textbf{Action} in FEAP.

The solution algorithm used by FEAP to solve problems is defined by a “macro statement program”. By properly specifying the macro program a very wide range of applications may be addressed – including both linear and nonlinear, as well as, steady state and transient applications.

The description for the macro statements may be obtained from the manual entry for each command, (e.g., use manual \texttt{tang} to obtain all options and actions for the \texttt{tang} command).
### 3.2 Overview on commands

<table>
<thead>
<tr>
<th>Makro</th>
<th>Aufgabe</th>
</tr>
</thead>
<tbody>
<tr>
<td>acce</td>
<td>Beschleunigungen</td>
</tr>
<tr>
<td>arcl</td>
<td>Bogenlängenverfahren</td>
</tr>
<tr>
<td>augm</td>
<td>Augmented Lagrange Verfahren</td>
</tr>
<tr>
<td>auto</td>
<td>Zeitschrittsteuerung automatisch</td>
</tr>
<tr>
<td>back</td>
<td>Zeitschritt rückgängig machen</td>
</tr>
<tr>
<td>bfgs</td>
<td>BFGS - Verfahren</td>
</tr>
<tr>
<td>chec</td>
<td>System - Test</td>
</tr>
<tr>
<td>cmas</td>
<td>Massenmatrix - konsistent</td>
</tr>
<tr>
<td>cont</td>
<td>Kontakt</td>
</tr>
<tr>
<td>conv</td>
<td>Konvergenzabfrage</td>
</tr>
<tr>
<td>copy</td>
<td>Kopiere Versch.vектор in Lastvektor</td>
</tr>
<tr>
<td>crit</td>
<td>Schädigung CFK</td>
</tr>
<tr>
<td>curv</td>
<td>Kurven für Lasten und Lagerungen</td>
</tr>
<tr>
<td>dmp</td>
<td>Dämpfungsmatrix</td>
</tr>
<tr>
<td>data</td>
<td>Aenderung von Daten</td>
</tr>
<tr>
<td>debu</td>
<td>Debug - Option</td>
</tr>
<tr>
<td>detk</td>
<td>Determinante der Steifikeitsmatrix</td>
</tr>
<tr>
<td>disp</td>
<td>Verschiebungsausgabe</td>
</tr>
<tr>
<td>dt</td>
<td>Zeitschritt setzen</td>
</tr>
<tr>
<td>dplo</td>
<td>Setze Linie für Ausgabe</td>
</tr>
<tr>
<td>eigi</td>
<td>Eigenvektor (inverse Iteration)</td>
</tr>
<tr>
<td>eigk</td>
<td>Eigenvektor (alle mit RSG)</td>
</tr>
<tr>
<td>eigl</td>
<td>Alternative IF/ELSE/ENDIF (Batch)</td>
</tr>
<tr>
<td>end</td>
<td>Ende Makrokommandos (Batch)</td>
</tr>
<tr>
<td>endi</td>
<td>Ende IF/ELSE/ENDIF (Batch)</td>
</tr>
<tr>
<td>epsq</td>
<td>Eingeprägte Verschiebungen aus E</td>
</tr>
<tr>
<td>erro</td>
<td>Fehlerberechnung</td>
</tr>
<tr>
<td>exit</td>
<td>Ende Makrokommandos (interaktiv)</td>
</tr>
<tr>
<td>ext</td>
<td>Extended System</td>
</tr>
<tr>
<td>fast</td>
<td>Schreibe Systemdaten in Binär-File</td>
</tr>
<tr>
<td>feas</td>
<td>Eigenvektorberechnung FEAST</td>
</tr>
<tr>
<td>form</td>
<td>Lastvektor</td>
</tr>
<tr>
<td>four</td>
<td>Fourierreihe</td>
</tr>
<tr>
<td>fsol</td>
<td>Fourierlösung</td>
</tr>
<tr>
<td>fsum</td>
<td>Fouriersummation</td>
</tr>
<tr>
<td>geom</td>
<td>Geometrische Matrix</td>
</tr>
<tr>
<td>help</td>
<td>Hilfe</td>
</tr>
<tr>
<td>hist</td>
<td>History, Ablauf der Makros</td>
</tr>
<tr>
<td>iden</td>
<td>Einheitsmatrix</td>
</tr>
<tr>
<td>if</td>
<td>Alternative IF/ELSE/ENDIF (Batch)</td>
</tr>
<tr>
<td>iimp</td>
<td>Felder auf Elementebene</td>
</tr>
<tr>
<td>init</td>
<td>Dynamische Rechnung: Anfangswerte</td>
</tr>
<tr>
<td>jint</td>
<td>Materielle Kräfte, J - Integral</td>
</tr>
<tr>
<td>lamb</td>
<td>Beullastfaktor</td>
</tr>
<tr>
<td>lan</td>
<td>Eigenvektorberechnung LANCZOS</td>
</tr>
<tr>
<td>line</td>
<td>Lineare Berechnung</td>
</tr>
<tr>
<td>lmas</td>
<td>Massenmatrix - Lumped</td>
</tr>
<tr>
<td>loop</td>
<td>Schleifen - Anfang</td>
</tr>
<tr>
<td>macn</td>
<td>vom Anwender definierte Macros</td>
</tr>
<tr>
<td>man</td>
<td>Manual</td>
</tr>
<tr>
<td>mesh</td>
<td>Netzänderung</td>
</tr>
<tr>
<td>mate</td>
<td>Materialänderung</td>
</tr>
<tr>
<td>newf</td>
<td>Laständerung</td>
</tr>
<tr>
<td>next</td>
<td>Schleifen - Ende</td>
</tr>
<tr>
<td>nonl</td>
<td>Nichtlineare Berechnung</td>
</tr>
<tr>
<td>nopr</td>
<td>Datenausgabe deaktivieren</td>
</tr>
<tr>
<td>para</td>
<td>Definition von Konstanen</td>
</tr>
<tr>
<td>parv</td>
<td>Postprozessor PARAView</td>
</tr>
<tr>
<td>paus</td>
<td>Pause</td>
</tr>
<tr>
<td>pbcg</td>
<td>PBCG - Verfahren</td>
</tr>
<tr>
<td>pgmr</td>
<td>PGMRES - Verfahren</td>
</tr>
<tr>
<td>plot</td>
<td>Graphik</td>
</tr>
<tr>
<td>pola</td>
<td>Polarkoordinaten</td>
</tr>
<tr>
<td>post</td>
<td>Postprozessor</td>
</tr>
<tr>
<td>prin</td>
<td>Datenausgabe aktivieren</td>
</tr>
<tr>
<td>prco</td>
<td>Vorkonditionierung</td>
</tr>
<tr>
<td>proc</td>
<td>Prozedur erstellen, abrufen</td>
</tr>
<tr>
<td>prop</td>
<td>Last - Zeit - Funktionen</td>
</tr>
<tr>
<td>quit</td>
<td>Programm - Abruch</td>
</tr>
<tr>
<td>reac</td>
<td>Reaktionskräfte</td>
</tr>
<tr>
<td>read</td>
<td>Datensatz lesen</td>
</tr>
<tr>
<td>reme</td>
<td>Netzverfeinerung</td>
</tr>
<tr>
<td>rest</td>
<td>Restart</td>
</tr>
<tr>
<td>rhin</td>
<td>Postprozessor RHINo</td>
</tr>
<tr>
<td>rinp</td>
<td>Neuberechnung</td>
</tr>
<tr>
<td>save</td>
<td>Datensicherung</td>
</tr>
<tr>
<td>show</td>
<td>Zustandsinformationen</td>
</tr>
<tr>
<td>sigq</td>
<td>S, C für eingeprägte Verschiebungen aus E</td>
</tr>
<tr>
<td>smoo</td>
<td>Netzglättung</td>
</tr>
<tr>
<td>solv</td>
<td>Lösung</td>
</tr>
<tr>
<td>splo</td>
<td>Setze Linie für Ausgabe</td>
</tr>
<tr>
<td>stre</td>
<td>Spannungsauflage</td>
</tr>
<tr>
<td>subs</td>
<td>Eigenvektorberechnung SUBSPACE</td>
</tr>
<tr>
<td>summ</td>
<td>Ueberlagerung von Versch./Sp.</td>
</tr>
<tr>
<td>Makro</td>
<td>Aufgabe</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>tang</td>
<td>Tang. Steifigkeitsmatrix (Symm.)</td>
</tr>
<tr>
<td>tec</td>
<td>Postprozessor TECPLOT</td>
</tr>
<tr>
<td>time</td>
<td>Zeit ändern</td>
</tr>
<tr>
<td>tol</td>
<td>Toleranz des Iterations - Fehlers</td>
</tr>
<tr>
<td>tplo</td>
<td>Kurvendiagramme</td>
</tr>
<tr>
<td>trans</td>
<td>Zeitabhängige Verfahren Anfangswerte</td>
</tr>
<tr>
<td>ueig</td>
<td></td>
</tr>
<tr>
<td>umas</td>
<td>Massenmatrix - unsymmetrisch</td>
</tr>
<tr>
<td>utan</td>
<td>Tang. Steifigkeitsmatrix - unsymmetrisch</td>
</tr>
<tr>
<td>u pdh</td>
<td>Update $H_2 \rightarrow H_1$ for convergence</td>
</tr>
<tr>
<td>velo</td>
<td>Geschwindigkeiten</td>
</tr>
<tr>
<td>writ</td>
<td>Datensatz schreiben</td>
</tr>
<tr>
<td>yang</td>
<td>Knickwinkelausgabe Flieslinien</td>
</tr>
<tr>
<td>yevo</td>
<td>Evolutionsstrategie Flieslinien</td>
</tr>
<tr>
<td>ygra</td>
<td>Gradientenermittlung Flieslinien</td>
</tr>
<tr>
<td>ymsxh</td>
<td>Netz aktualisieren Flieslinien</td>
</tr>
<tr>
<td>ytab</td>
<td>Simplex-Tableau lösen Flieslinien</td>
</tr>
<tr>
<td>ytry</td>
<td>direkte Suche Flieslinien</td>
</tr>
</tbody>
</table>
## 3.3 Overview on actions

<table>
<thead>
<tr>
<th>Aufgabe</th>
<th>Makro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hilfe</td>
<td>help</td>
</tr>
<tr>
<td>Anderung von Daten</td>
<td>data</td>
</tr>
<tr>
<td>Anwender - Macros</td>
<td>macn</td>
</tr>
<tr>
<td>Augmented Lagrange Verfahren - Update</td>
<td>augm</td>
</tr>
<tr>
<td>Beschleunigungen</td>
<td>acce</td>
</tr>
<tr>
<td>Beullastfaktor</td>
<td>lamb</td>
</tr>
<tr>
<td>BFGS - Verfahren</td>
<td>bfgs</td>
</tr>
<tr>
<td>Bogenlängenverfahren</td>
<td>arcl</td>
</tr>
<tr>
<td>Dämpfungsmatrix</td>
<td>damp</td>
</tr>
<tr>
<td>Datenausgabe aktivieren</td>
<td>prin</td>
</tr>
<tr>
<td>Datenausgabe deaktivieren</td>
<td>nopr</td>
</tr>
<tr>
<td>Datensatz lesen</td>
<td>read</td>
</tr>
<tr>
<td>Datensatz schreiben</td>
<td>writ</td>
</tr>
<tr>
<td>Datensicherung</td>
<td>save</td>
</tr>
<tr>
<td>Debug - Option</td>
<td>debu</td>
</tr>
<tr>
<td>Definition von Konstanten</td>
<td>para</td>
</tr>
<tr>
<td>Determinante der Steifigkeitsmatrix</td>
<td>detk</td>
</tr>
<tr>
<td>direkte Suche Fließlinien</td>
<td>ytry</td>
</tr>
<tr>
<td>Dynamische Rechnung: Anfangswerte</td>
<td>init</td>
</tr>
<tr>
<td>Eigenvektorberechnung SUBSPACE</td>
<td>subs</td>
</tr>
<tr>
<td>Eigenvektorberechnung LANCZOS</td>
<td>lan</td>
</tr>
<tr>
<td>Eigenvektorberechnung FEAST</td>
<td>feas</td>
</tr>
<tr>
<td>Eigenvektorberechnung RSG</td>
<td>eigl</td>
</tr>
<tr>
<td>Eigenvektor (inverse Iteration)</td>
<td>eigi</td>
</tr>
<tr>
<td>Einheitsmatrix</td>
<td>iden</td>
</tr>
<tr>
<td>eingeprägte Verschiebungen aus E</td>
<td>epsq</td>
</tr>
<tr>
<td>S, C für eingeprägte Verschiebungen aus E</td>
<td>sigq</td>
</tr>
<tr>
<td>Ende Makrokommandos (Batch)</td>
<td>end</td>
</tr>
<tr>
<td>Ende Makrokommandos (interaktiv)</td>
<td>exit</td>
</tr>
<tr>
<td>Evolutionsstrategie Fließlinien</td>
<td>yevo</td>
</tr>
<tr>
<td>Extended System</td>
<td>ext</td>
</tr>
<tr>
<td>Fehlerberechnung</td>
<td>erro</td>
</tr>
<tr>
<td>Felder auf Elementebene</td>
<td>iimp</td>
</tr>
<tr>
<td>Fourierlösung</td>
<td>fsol</td>
</tr>
<tr>
<td>Fourierreihe</td>
<td>four</td>
</tr>
<tr>
<td>Fouriersummation</td>
<td>fsum</td>
</tr>
<tr>
<td>Geometrische Matrix</td>
<td>geom</td>
</tr>
<tr>
<td>Geschwindigkeiten</td>
<td>velo</td>
</tr>
<tr>
<td>Gradientenermittlung Fließlinien</td>
<td>ygra</td>
</tr>
<tr>
<td>Graphik</td>
<td>plot</td>
</tr>
<tr>
<td>Aufgabe</td>
<td>Makro</td>
</tr>
<tr>
<td>IF/ELSE/ENDIF (Batch)</td>
<td>if</td>
</tr>
<tr>
<td>IF/ELSE/ENDIF (Batch)</td>
<td>else</td>
</tr>
<tr>
<td>IF/ELSE/ENDIF (Batch)</td>
<td>endi</td>
</tr>
<tr>
<td>J - Integral, Materielle Kräfte</td>
<td>jint</td>
</tr>
<tr>
<td>Knickwinkelausgabe Fließlinien</td>
<td>yang</td>
</tr>
<tr>
<td>Kontakt</td>
<td>cont</td>
</tr>
<tr>
<td>Konvergenzabfrage</td>
<td>conv</td>
</tr>
<tr>
<td>Kopiere Vers.vektor in Lastvektor</td>
<td>copy</td>
</tr>
<tr>
<td>Kurvendiagramme</td>
<td>tplo</td>
</tr>
<tr>
<td>Kurven für Lasten und Lagerungen</td>
<td>curv</td>
</tr>
<tr>
<td>Laständerung</td>
<td>newf</td>
</tr>
<tr>
<td>Lastvektor</td>
<td>form</td>
</tr>
<tr>
<td>Last - Zeit - Funktionen</td>
<td>prop</td>
</tr>
<tr>
<td>Lineare Berechnung</td>
<td>line</td>
</tr>
<tr>
<td>Linie für Ausgabe</td>
<td>dplo</td>
</tr>
<tr>
<td>Linie für Ausgabe</td>
<td>splo</td>
</tr>
<tr>
<td>Lösung</td>
<td>solv</td>
</tr>
<tr>
<td>Massenmatrix - konsistent</td>
<td>cmas</td>
</tr>
<tr>
<td>Massenmatrix - lumped</td>
<td>lmas</td>
</tr>
<tr>
<td>Massenmatrix - unsymmetrisch</td>
<td>umas</td>
</tr>
<tr>
<td>Manual</td>
<td>man</td>
</tr>
<tr>
<td>Materialänderung</td>
<td>mate</td>
</tr>
<tr>
<td>Netz aktualisieren Fließlinien</td>
<td>ymsh</td>
</tr>
<tr>
<td>Netzänderung</td>
<td>mesh</td>
</tr>
<tr>
<td>Netzglättung</td>
<td>smoo</td>
</tr>
<tr>
<td>Netzverfeinerung</td>
<td>reme</td>
</tr>
<tr>
<td>Nichtlineare Berechnung</td>
<td>nonl</td>
</tr>
<tr>
<td>Pause</td>
<td>paus</td>
</tr>
<tr>
<td>PBCG - Verfahren</td>
<td>pbcg</td>
</tr>
<tr>
<td>PGMRES - Verfahren</td>
<td>pgmr</td>
</tr>
<tr>
<td>Polarkoordinaten</td>
<td>pola</td>
</tr>
<tr>
<td>Postprozessor</td>
<td>post</td>
</tr>
<tr>
<td>Postprozessor PARAVIEW</td>
<td>parv</td>
</tr>
<tr>
<td>Postprozessor RHINO</td>
<td>rhin</td>
</tr>
<tr>
<td>Postprozessor TECPLLOT</td>
<td>tec</td>
</tr>
<tr>
<td>Programm - Abbruch</td>
<td>quit</td>
</tr>
<tr>
<td>Prozedur erstellen, abrufen</td>
<td>proc</td>
</tr>
<tr>
<td>Reaktionskräfte</td>
<td>reac</td>
</tr>
<tr>
<td>Restart</td>
<td>rest</td>
</tr>
<tr>
<td>Schädigung CFK</td>
<td>crit</td>
</tr>
<tr>
<td>Schleifen - Anfang</td>
<td>loop</td>
</tr>
<tr>
<td>Schleifen - Ende</td>
<td>next</td>
</tr>
<tr>
<td>Schreibe Systemdaten in Binär-File</td>
<td>fast</td>
</tr>
<tr>
<td>Simplex-Tableau lösen Fliesslinien</td>
<td>ytab</td>
</tr>
<tr>
<td>Spannungsaugabe</td>
<td>stre</td>
</tr>
<tr>
<td>System - Test</td>
<td>chec</td>
</tr>
<tr>
<td>Tang. Steifigkeitsmatrix (Symm.)</td>
<td>tang</td>
</tr>
<tr>
<td>Tang. Steifigkeitsmatrix (Unsymm.)</td>
<td>utan</td>
</tr>
<tr>
<td>Toleranz des Iterations - Fehlers</td>
<td>tol</td>
</tr>
<tr>
<td>Ueberlagerung von Versch./Spann.</td>
<td>summ</td>
</tr>
<tr>
<td>Verschiebungsausgabe</td>
<td>disp</td>
</tr>
<tr>
<td>Vorkonditionierung</td>
<td>prco</td>
</tr>
<tr>
<td>Update $H_2 \to H_1$</td>
<td>updH</td>
</tr>
<tr>
<td>Zeit ändern</td>
<td>time</td>
</tr>
<tr>
<td>Zeitabhängige Verfahren Anfangswerte</td>
<td>trans</td>
</tr>
<tr>
<td>Zeitschritt setzen</td>
<td>dt</td>
</tr>
<tr>
<td>Zeitschritt rückgängig machen</td>
<td>back</td>
</tr>
<tr>
<td>Zeitschrittsteuerung automatisch</td>
<td>auto</td>
</tr>
<tr>
<td>Zustandsinformationen</td>
<td>show</td>
</tr>
</tbody>
</table>
3.4 Macros in detail

ACCE

\texttt{acce,\langle n1,n2,n3\rangle}
\texttt{acce,all}

The macro command \texttt{acce} may be used to print the current values of the “acceleration” vector as follows:

1. Using the macro command:
   \texttt{acce,n1,n2,n3}
   prints out the current acceleration vector for nodes ‘n1’ to ‘n2’ at increments of ‘n3’ (default = 1). If ‘n2’ is not specified only the value of node ‘n1’ is output. If both ‘n1’ and ‘n2’ are not specified only the first nodal acceleration is reported.

2. If the macro command is specified as:
   \texttt{acce,all}
   prints all nodal quantities.

In order to output a solution vector it is first necessary to specify macro commands to compute the desired values, i.e., a dynamic analysis.
The `arcl` macro computes an arclength solution with the following options:

<table>
<thead>
<tr>
<th><code>xxxx</code></th>
<th>n1</th>
<th>n2</th>
<th>n3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 5</td>
<td>0 or 1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>modi</td>
<td>n1</td>
<td>n2</td>
<td>n3</td>
</tr>
<tr>
<td>add</td>
<td>e1</td>
<td>τ</td>
<td></td>
</tr>
<tr>
<td>impf</td>
<td>e1</td>
<td>ξ</td>
<td></td>
</tr>
<tr>
<td>on</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>off</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>check</td>
<td>e1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>step</td>
<td>Id [6]</td>
<td>mono[1]</td>
<td>rm [0.5]</td>
</tr>
</tbody>
</table>

With `arcl`, `n1`, `n2`, the arclength method is initialized. 'n1' options are defined as follows:

| n1 = 0 | Normal plane, modified newton solution |
| n1 = 1 | Updated normal plane, modified newton solution |
| n1 = 2 | Normal plane, full newton solution |
| n1 = 3 | Updated normal plane, full newton solution |
| n1 = 4 | Displacement control, modified newton solution |
| n1 = 5 | Displacement control, full newton solution |
| n2 = 0 | Use current values for arclength and load direction (Initial default is calculated by first solution step). |
| n2 = 1 | Change current values for arclength and load direction |

With `arcl,modi` or `arcl,,1` the iteration parameters could be modified interactive, due to the chosen version.

With `arcl,modi,n1,n2,n3` the iteration parameters could be modified directly (displacement control: n1=node, n2=dof, n3=value, else: n1=ds n2=direction).

With `arcl,add` a scaled eigenvector is added. 'e1' denotes the number of the eigenvector to be included and τ is a scaling factor such that

\[ u \leftarrow u + \frac{|u|}{|\phi_{e1}|} \tau \phi_{e1} \]

where u is the current solution and \( \phi_{e1} \) is the \( e1^{th} \) eigenvector.

With `arcl,impf` an eigenvector is added in the sense of an imperfection. 'e1' denotes the number of the eigenvector to be included and ξ is the maximum amplitude such that
\( \mathbf{u} \leftarrow \mathbf{u} + \xi \phi_{e1} \)

where \( \mathbf{u} \) is the current solution and \( \phi_{e1} \) is the \( e1^{th} \) eigenvector. Often \( \xi \) is chosen as thickness 'h'.
Remarks:

- `arcl` has to be called once at the beginning of each series of macro commands, when a nonlinear problem is to be solved using this method. With this call all flags will be set to perform an arc length solution. It should be noted that you can reset this flags to continue e.g. with pure newton method when you type `arcl_off`. The command `arcl_on` then switches back to the arc length procedure.

- `arcl` requires at this stage of implementation that the time increment `dt` is set to 1.0. This is done automatically. Thus no input of `dt` is necessary.

- For full and modified Newton methods the standard procedures for one step are:

  ```plaintext
  time
time
tang,,1
loop,,N
next
```

- For the calculation of load deflection curves we need to specify proportional load with `prop`. However you may use the default values of `prop`, since the actual load level is computed by `arcl`.

- If you like to perform a branch–switching you must calculate the eigenvectors associated with the bifurcation load first (You may use a shift on the tangent, see `tang`.

- The command `arcl_chec` tells you whether the stability point is a limit point or a bifurcation point. In case of a bifurcation point the value returned by `arcl_chec` should be zero up to a tolerance).

  \[
  \phi^T P = \begin{cases} 
  0 & \text{Bifurcation point} \\
  \neq 0 & \text{Limit point}
  \end{cases}
  \]

- The branch–switching is initiated by the command `arcl_add, n1, n2` which adds the \( n1^{th} \) eigenvector \( \phi_{n1} \) to the current displacement field as shown above. \( n2 \) is the scaling factor \( \tau \). In case \( n2 \) is zero a scaling factor is automatically computed using the formula

  \[
  \tau = 100 \frac{u^T \phi_{n1}}{\sqrt{(u^T u)(\phi_{n1}^T \phi_{n1})}} + 1
  \]

- Note, that the command `arcl_add` can be used generally. Thus one has not to specify `arcl_kfl` before applying this command. This means that the command can be used also with e.g. standard load control analysis.

- After the addition of an eigenvector \( \phi_{n1} \) to the displacement field \( u \) a new equilibrium state has to be computed on the secondary branch. This is simply obtained by the following macro series

  ```plaintext
  loop,,N
tang,,1
next
```

- Within arclength solution techniques the load increments are modified automatically. Modifications can be initialized by `arcl_step`. The current increment size is

  \[
  ds_0 = \alpha * ds_0(i - 1)
  \]
where $ds_0(i-1)$ is the increment size of the previous time step.

The modification factor $\alpha$ is determined by

$$\alpha = (Id/I(i-1))^{sp}$$

where $Id (=n1)$ is a user defined number of iterations within a time step to achieve convergence, and $I(i-1)$ is the number of iterations needed in the previous time step. The exponent $sp$ is set to 0.5 [recommended $0.5 \leq sp \leq 1.0$]. A large value for $Id (=n1)$ will result in larger step sizes.

Within the iteration a monotone test is performed using the ratio of the incremental displacement norms of two subsequent iteration steps.

$$\theta_k = \frac{\|\Delta v(i+1)\|}{\|\Delta v(i)\|}.$$ 

If this ratio is larger than an user defined number $mono (=n2)$ the time step is restarted with a reduced $ds_0$ with the reduction factor

$$\alpha = \sqrt{rm \cdot n2 \cdot \theta_k}.$$ 

The number $rm$ is a safety factor ($rm$ approx. 0.5) For certain problems (e.g. stability points) it might be necessary to use an large number for $n2$ ($n2=3$ to 5) in order to prevent very small step sizes.

The step control can be turned off by the macro sequence

```
arcloff
arclon
```

References

The theoretical background is described e.g. in Simo, Wriggers, Schweizerhof, Taylor (1986) [24] or Wagner, Wriggers (1998) [31].
The macro command `augm` updates the Lagrange parameter in a nested augmented iteration on element level using (isw = 10). This feature is element depending. It can be used e.g. for incompressible rubber materials to improve incompressibility or for contact formulations.
Using the macro command `auto` leads to a variable length of the time increment within a dynamic analysis. Currently this macro is implemented for the time stepping algorithms Newm, HHT and Alpha.

The necessary parameter are defined as follows:

<table>
<thead>
<tr>
<th>parameter</th>
<th>abbreviation</th>
<th>description</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>htol</td>
<td>T</td>
<td>tolerance value for residual</td>
<td>0.1</td>
</tr>
<tr>
<td>dtmax</td>
<td>dtmax</td>
<td>maximum desired time step length</td>
<td>dt</td>
</tr>
<tr>
<td>dtdo</td>
<td>DA</td>
<td>reduction factor</td>
<td>0.85</td>
</tr>
<tr>
<td>dtup1</td>
<td>DG</td>
<td>increasing factor 1</td>
<td>0.80</td>
</tr>
<tr>
<td>dtup2</td>
<td>DM</td>
<td>increasing factor 2</td>
<td>1.25</td>
</tr>
<tr>
<td>eta</td>
<td>η</td>
<td>upper limiting factor</td>
<td>1.1</td>
</tr>
<tr>
<td>xsi</td>
<td>ξ</td>
<td>accuracy parameter</td>
<td>0.005</td>
</tr>
<tr>
<td>gamma</td>
<td>γ</td>
<td>lower limiting factor</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**Theory:** The algorithm is based on the so called "Half-step-residual".

**General Remarks:**

The macro has to be initialized with `auto, init`.

The tolerance htol is typically $0.1 \cdot P_{\text{max}}$. Here $P_{\text{max}}$ is a maximum load value and could be calculated with `auto, load`.

With `auto, para,<dtdo,dtup1,dtup2>` the default defined parameters could be changed.

In a dynamic analysis the length of the next time increment will be checked by adding the `auto` command at the end of each converged time step.

For small errors the next time step will be started with an unchanged or (conservatively) increased time increment, whereas for larger errors the time step has to be repeated with a reduced time increment.
A more detailed description could be found in the theory part of this manual.

With `auto`, `impl`, `<eta,xsi,gamma>` an a priori time step length calculation for the IMPL-EX scheme (damage/plasticity) is performed according to

\[
\Delta t_{n+1}^2 \leq \xi \alpha^{ref} \Delta t_n^2 \min \left\{ \frac{1}{\| \Delta \lambda_n(x) - \frac{\Delta \lambda_n}{\Delta \lambda_{n-1}} \Delta \lambda_{n-1}(x) \|} \right\}
\]

with \( \alpha^{ref} = \frac{\alpha_0}{\sqrt{E}} = q_0 \) in case of damage and \( \alpha^{ref} = \frac{\alpha_0}{E} \) (uniaxial elastic strain) in case of elasto-plastic models. The increment of the internal variable is \( \Delta \lambda_n \) at time step \( n \). The accuracy parameter \( \xi \) directly affects the next time step size. While the limiting parameter \( \eta \) affects the next time step size in case of an increasing step size according to

\[
\Delta t_{n+1} \leq \eta \Delta t_n, \quad \Delta t_{n+1} \leq \Delta t_{max}
\]

with \( \Delta t_{max} \) as the manually chosen value by the macro `dt`, `\Delta t`.

The last parameter \( \gamma \) performing a back step if the calculated time step size \( \Delta t_{n+1} \leq \gamma \Delta t_n \).

**Usage:**

Only the above given 'time steps'-box is necessary.

**References:**

Details on the theoretical and numerical background can be found in Oliver et.al. (2008) [15].
BACK

back,,<dtnew>

The use of the back macro command will decrement the current time by dt, the current time increment. In addition, the previous value of the proportional loading will be recomputed, if necessary. The value of the current time and proportional loading are reported in the output (or to the screen). The back macro also will recompute the dynamic state at the old time for every time integration of the equations of motion, as well as, restore the stress data base for any elements with non–linear constitutive equations which require variables other than the displacement state to compute a solution.

Time steps before the current time step could not be reached. Thus only ONE back is allowed.

As an option, it is possible to specify a new time increment for integrations to be continued. The value of ’dtnew’ is then used to perform the updates on the solutions in the same way as if the command dt,,dtnew were given. See manual on dt macro command for additional details.

Thus back,,dtnew is equal to the sequence
# back
# dt,,dtnew

Remark:

- In case of an active macro tplo reaction forces have to be recalculated via reac,,all for a correct plot of reaction forces.
The macro command `beta` is obsolete. Please use `trans`. 
The `bfgs` macro can be used to solve a nonlinear solutions step iteratively. The iterative procedure is based on a (BFGS) quasi Newton method which applies to problems with symmetric tangent matrices. The method involves updating the initial tangent matrix and provides a secant approximation to the tangent matrix. The convergence behavior is superlinear.

Remarks:

1. Before executing the `bfgs` macro the initial tangent matrix must be computed via the `tang` macro.
2. The parameter ’n1’ denotes the maximum number of iteration vectors (default and absolute maximum= 25).
3. The parameter ’v2’ is the line search tolerance and may be chosen between 0.5 and 0.9. The default 0.8 will generally produce good performance.
**CHEC**

The **chec** macro command requests a check of the mesh consistency. It is necessary for elements to have checking capability for the “isw = 2” option in order for **chec** to report results. Typical tests may include jacobian tests at nodes, tests on node sequencing, etc.
The macro command `cmas` is used to compute a consistent (i.e., a non-diagonal) “mass” matrix. Each element computes a contribution to the consistent mass in the array ‘s’ when ‘isw’ is 5 and ‘intyp’ eq.1. A consistent mass or a lumped mass (see macro command `lmas`) may be used for transient solutions computed using the Newmark method (see macro command `trans`). Both may also be used for eigencomputations (see macro command `subs`).
The `cont` macro is used to activate the contact logic during macro execution. It is necessary to describe the surfaces which may come into contact during the analysis when specifying the mesh data (i.e., see `icon` in the FEAP MESH USERS MANUAL).

The contact logic may be skipped during execution of a macro program (even though the contact surfaces are defined using `icon`) by:

1. never specifying a `cont` macro instruction, or
2. by specifying the `off` option in the second field.

The program will automatically readjust the profile of the equations during each computation of a 'tangent' matrix to account for the current configuration of contacting surfaces.

During execution it is possible to reset the values of the penalty parameters on any contact surface, 'n', to the values of 'pen-n' and 'pen-t' in normal and tangential direction. This permits the adjustment of the penalty parameter from a smaller to larger value during iterations. For problems in which large deformations occur the convergence to a solution may lead to a large number of iterations when large penalty parameters are involved. On the other hand, the use of a lower penalty parameter may result in unacceptable large penetrations across the contact surface. In these situations, it is recommended that the penalty parameter be adjusted to larger values during the iteration process in each load.

**References**

The theoretical background is described e.g. in Simo, Wriggers, Schweizerhof, Taylor (1986) [24]. Information on choosing the right penalty parameter is given in Nour-Omid, Wriggers (1987) [13].
The macro command `conv` may be used to check the convergence of a solution. If no convergence is achieved within a user defined number of equations the calculation is automatically set back to the original values via the back-macro without time. For a new time step the value of dt is reduced to 'dtnew = \( \xi \cdot dt \)', with \( 0 < \xi < 1 \). If no value of \( \xi \) is set, the program asks the value in interactive mode. With the macro command `conv,stop` the current loop is terminated.

Example:

```
# loop,,20
# time
# loop,iter,10
# tang,,1
# next
# conv,,0.5
# next
```

Within 20 time steps a nonlinear calculation is performed. 'conv,,0.5' leads to a recalculation with half length time step, if no convergence is achieved within 10 increments.

**Remark:** For arclength method `conv` do not set 'dt' but the constant arclength ds0.

With the macro command `conv` it is possible to do a **bisection method** to find a limit point. Hereby the load step is only reduced if no convergence is achieved within a user defined number of equations. A consecutive bisection can be done in the following manner using to 2 blocks.

```
block 1    block 2
# loop,,20  # loop,,20
# time      # time
# loop,iter,10  # loop,iter,10
# tang,,1     # tang,,1
# next       # next
# conv,,0.5  # conv,,1
# next       # next
```

The first block is used to follow the nonlinear solution path until the first divergence occurs. Then with the 2nd block the limit point can be found. For this purpose the step-length 'dt' is reduced in each time step. 'No convergence' is here defined as 'no solution within 10 iterations'.
When using the standard Newton-Raphson scheme, then the macro conv can be used to dynamically adjust the load/time step increment \( dt \). Two possibilities are implemented. The first one is analogue to the implementation in Abaqus. The step size is chosen based on the convergence behavior of the previous step.

<table>
<thead>
<tr>
<th>initialization</th>
<th>time steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>( dt, dt )</td>
<td>( loop_n )</td>
</tr>
<tr>
<td>( conv, aini )</td>
<td>( time )</td>
</tr>
<tr>
<td></td>
<td>( loop_m )</td>
</tr>
<tr>
<td></td>
<td>( tang_1 )</td>
</tr>
<tr>
<td></td>
<td>( next(m) )</td>
</tr>
<tr>
<td></td>
<td>( conv, auto )</td>
</tr>
<tr>
<td></td>
<td>( next(n) )</td>
</tr>
</tbody>
</table>

To use this scheme for dynamic time/load step size, it is necessary to initialize the scheme with \( conv, aini, < itinc, itdec, dtmin > \). With itinc the scheme increases \( dt \) to \( \xi_{inc} \cdot dt \) if the number of iterations of the previous step was smaller than itinc. Vice versa the scheme decreases \( dt \) to \( \xi_{dec} \cdot dt \) if the previous step needed more than itdec iterations.

To check the conditions it is necessary to perform \( conv, auto, < \xi_{dec}, \xi_{inc} > \) after each iteration loop. If the Newton-Raphson scheme do not converge and \( dt \) gets smaller than dtmin, the scheme tries the step again with \( dt_{max} = 2 \cdot dt \) (dt set by the \( dt \) macro). If it is still unsuccessful and dtmin is reached again, the load step loop will be terminated. The default parameters are:

\( conv, aini, 4,10,dt/1000 \)
\( conv, auto, 0.75,1.5 \)

Reference:

The second method is using \( conv,aunr \). Here no initialization is necessary.

<table>
<thead>
<tr>
<th>time steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>( loop_n )</td>
</tr>
<tr>
<td>( time )</td>
</tr>
<tr>
<td>( loop_m )</td>
</tr>
<tr>
<td>( tang_1 )</td>
</tr>
<tr>
<td>( next(m) )</td>
</tr>
<tr>
<td>( conv,aunr )</td>
</tr>
<tr>
<td>( next(n) )</td>
</tr>
</tbody>
</table>

The scheme chooses a value \( dt \) based on a norm of the displacement rates. The rates are specified by \( V_n = \Delta U_n / \Delta t_n \). With these rates an error is defined as \( E_{t+\Delta t} = \frac{dt}{2} \| V_{t+\Delta t} - V_t \| \). If \( R \geq dtol \) a backstep is performed and \( dt \) is set to \( q \cdot dt \), with \( q = \max \left( 0.8 \sqrt{dtol / R_{t+\Delta t}}, 0.1 \right) \). In case of no convergence in the current step also a backstep is performed and \( dt \) is set to 0.25-\( dt \). If the step converged and \( R < dtol \) then \( q \) is set to \( q = \min \left( 0.8 \sqrt{dtol / R_{t+\Delta t}}, 2.0 \right) \) and \( dt \) is set again to \( q \cdot dt \). The only default value for this scheme is \( dtol = 10^{-3} \). Additionally a maximum value for \( dt \) (\( dt_{max} \)) can be chosen to prevent unlimited growth of \( dt \).

And a minimum value for \( dt \) (\( dt_{min} \)) can be set, so the scheme stops the current loop if either the norm is not satisfied or the Newton-Raphson scheme do not converge and \( dt \) is already reduced to \( dt_{min} \).

References:
The theoretical background is described in Sheng et.al. [23].
COPY

copy

The copy macro may be used to transfer the current solution vector for displacements into the load vector. This option may be used in conjunction with results from a previous analysis and a problem with all degree-of-freedoms restrained to combine solutions for graphics outputs, etc.
The **crit** macro is up to now not documented.
The macro command `curv` is described under MESH.
DAMP

damp.xxxx

This macro computes the damping matrix. A lumped damping, a consistent or an unsymmetric consistent damping may be used for transient solutions computed using a dynamic algorithm (see macro command `trans`).

The permissible values for `xxxx` are:

- `xxxx = lump`
- `xxxx = cons` (default)
- `xxxx = ucon`

Each element computes a contribution to the damping matrix in the arrays 's' and 'p' when 'isw' is 12.

The specification of `lump` is used to compute a lumped (i.e., a diagonal) “damping” matrix.

The specification of `cons` is used to compute a consistent “damping” matrix.

The specification of `ucon` is used to compute an unsymmetric consistent “damping” matrix together with `utan` in transient analysis and only for the standard solver.
During macro execution it is sometimes desirable to progressively change parameters, e.g., the time step size or the solution tolerance accuracy or the actual number + multiplier for a fourier solution. This could become cumbersome and require an excessive number of macro commands if implemented directly. Accordingly, the \texttt{data} command may be used in instances when the time step or tolerance or the fourier solution is to be varied during a \texttt{loop} execution.

The permissible values for \texttt{xxxx} are:

\begin{verbatim}
xxxx = tol
xxxx = dt
xxxx = four
\end{verbatim}

The actual values of the tolerance or time step size or the fourier solution are given after the \texttt{end} macro statement using the data inputs specified in the \texttt{tol}, \texttt{dt} or \texttt{four} manuals. For example, to vary time steps during a loop the commands:

\begin{verbatim}
loop, time, 3
data, dt
....
next, time
....
end
dt,, 0.1
dt,, 0.2
dt,, 0.4
\end{verbatim}

could be given to indicate three time steps with $\Delta t = 0.1, 0.2, \text{ and } 0.4$ respectively.

In interactive mode the program prompts `input \texttt{xxxx} macro >’ and the input for the specified macro has to be done, see the manuals on \texttt{dt}, \texttt{tol} and \texttt{four}.\end{verbatim}
DEBU

**debu,xxxx**

The **debu** macro command can be used to debug the program code. The parameter **xxxx** is defined as follows:

- `'....'` set the debug option to true
- `on` set the debug option to true
- `off` set the debug option to false (default value)

**Implemented**

- Prints all informations set in PSETA
- Prints informations for history fields
- Userdefined values on element level

**Further options**

- `par` opens files `foutpar,i` with `i=1,8` on units 51-58.
  These files could be used via the subroutine MPRINTP, which writes data with respect to the actual THREAD-number (in parallel processing).
  Other data could be written directly with `write(51 + ip,*)`.
  where `ip = OMP_GET_THREAD_NUM()` defines the active THREAD-number (ip=0,1,2,3,...).

*Institut f"ur Baustatik, KIT*
The macro command `detk` is used to print the current value of the determinant of the stiffness matrix. The determinant is set to 1 at the end of the first time step of a nonlinear calculation.

`detk, init` set the initial value to 1.

The determinant is always calculated in Arc–Length–Method. To get a correct value the tangent matrix have to be factored. For a plot of the determinant versus a single displacement use before calculation the macro `tplo`.
DIBC

dibc,,n1

The macro command `dibc` is currently not documented.
The macro command **disp** may be used to print the current values of the “displacement” vector as follows:

1. Using the macro command:
   ```
   disp.,<n1,n2,n3>
   ```
   prints out the current displacement vector for nodes 'n1’ to 'n2’ at increments of 'n3’ (default = 1). If 'n2’ is not specified only the value of node 'n1’ is output. If both 'n1’ and 'n2’ are not specified only the first nodal displacements is reported.

2. If the macro command is specified as:
   ```
   disp.all
   ```
   all terms in the displacement vector are printed.

3. If the macro command is specified as:
   ```
   disp.line
   ```
   displacements are printed at nodes on a line defined by dplo or splo.

4. If the macro command is specified as:
   ```
   disp.eigv,<n1,n2,n3>
   ```
   then the eigenvectors from 'n1’ to 'n2’ are reported at increments of 'n3’ (default = 1).
   Note that it is not possible to print certain nodes in each eigenvector — the entire vector is output to the screen/output file!

5. If the macro command is specified as:
   ```
   disp.evex,<n1,n2,n3>
   ```
   then the eigenvector from extended system is printed from 'n1’(def.=1) to 'n2’(def.=numnp) are printed at increments of 'n3’ (def.= 1).

6. If the macro command is specified as:
   ```
   disp.eigi,<n1,n2,n3>
   ```
   then the eigenvector from inverse iteration is printed from 'n1’(def.=1) to 'n2’(def.=numnp) are printed at increments of 'n3’ (def.= 1).
7. If the macro command is specified as:
   \texttt{disp,pola,n1}
   switches the output to polar directions. \( 'n1' = 12,13,23 \) means the plane \( i/k \) in which the transformation acts. Within the next print macro the displacements \( i \) and \( k \) will then be printed in polar coordinates \( (u_i = u_{\text{radial}} \text{ and } u_k = u_{\text{tangential}}) \).

8. If the macro command is specified as:
   \texttt{disp,cart}
   switches the output to cartesian directions (default!). Within the next print macro all displacements will then be printed in cartesian coordinates.

In order to output a solution vector it is first necessary to specify macro commands to compute the desired values: either normal solutions for a static or dynamic analysis, or the eigenvectors.
DPLO

dplo, set
dplo

The macro command **dplo** is used to print nodal displacements or nodal stresses along a line.
With **dplo, set** coordinates of starting point and end point of line are set.
**dplo** reset all values to zero.
With **disp**, line nodal displacements are printed.
With **stre**, line nodal stresses are printed.
DT

\[ dt_{\text{new}} = dt_{\text{old}} \cdot v2 \]

The dt macro command specifies the value of the time step for time dependent problems (i.e., transient or quasistatic problems). The value of `v1` indicates the time step to be used and should be greater or equal to zero (only for dynamic problems). Generally, it is necessary to use a time macro command, in conjunction with the dt command, to advance the time and compute proportional loading values if necessary.

With dt,,v2 the current time increment is multiplied by `v2`. Thus it holds \( dt_{\text{new}} = dt_{\text{old}} \cdot v2 \).
**EIGI**

\[ \text{eigi,} \ n1, \ n2 \]

The `eigi` command computes the lowest eigenvalue and eigenvector of the current tangent matrix.

- `n1` = max. number of iteration steps (default = 50)
- `n2` = terminating tolerance (default = 1.e-8)

- In most times a few iterations are sufficient.
- This macro is convenient in a path following calculation when the lowest eigenvalue is required in each load step.
- This computation is equal to the subspace iteration done with:
  
  \[ \text{iden} \]
  
  \[ \text{subs,} 1 \]

  solving the eigenvalue problem \[ (K_T - \omega I) \phi = 0 \]

- The inverse iteration is usually much faster than subspace iteration.
- The eigenvector is not saved in a restart file.

**N.B.** The term 'lowest eigenvalue' means the eigenvalue closest to Zero. There may be also some bigger negative eigenvalues. This can be checked using a 'storm sequence test' (not implemented yet) or looking at the number of negative diagonals.
The eigk macro computes the lowest eigenvalue of the problem $(K_T - \lambda M)\varphi = 0$ via the coordinate overrelaxation method (COR). This method is designed for the detection of dynamic stability points (c-stability) during an implicit solution with the Newmark–scheme.

The following table defines the input options

<table>
<thead>
<tr>
<th>xxxx</th>
<th>n1</th>
<th>n2</th>
<th>n3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>maxit</td>
<td>ω</td>
<td>η</td>
</tr>
<tr>
<td>chec</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>off</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>init</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In the above table the parameters have the following meaning:

- **maxit**: maximum number of iterations, default $n1 = 20$.
- **ω**: Overrelaxation parameter; default $\omega = 1.4$.
- **η**: Convergence tolerance for algorithm; default: $\eta = 10^{-5}$.

$x$x$x = chec$: Initializes COR to compute the eigenvalue of the above system of equations for a converged solution within the incremental Newmark scheme. This leads to a printout of the eigenvalue within the time stepping algorithm.

$x$x$x = off$: Stops computation of eigenvalue via COR.

$x$x$x = init$: Initializes the COR method with a previous computed eigenvector from subs and computes the eigenvalue. The associated commands in subs are:

```
subs,,1
subs,init.
```

**REMARK:** Before the coordinate overrelaxation method is started use subspace to compute an eigenvector as a starting vector for COR. Otherwise COR will not run efficiently for detecting singular points.
The following macro commands are an example for the computation of singular point during dynamic calculations.

\[ \text{dt}, \Delta t \]  
Time increment \( \Delta t \).

\[ \text{prop}, 1 \]  
Proportional load.

\[ \text{lmas} \]  
Lumped mass matrix.

\[ \text{tang} \]  
Tangent stiffness matrix.

\[ \text{subs}, 1 \]  
Compute starting vector for COR.

\[ \text{subs}, \text{init} \]  
Move first eigenvector to COR.

\[ \text{eigk}, \text{chec} \]  
Initialize COR method with default values.

\[ \text{trans} \]  
Initialize Newmark method with default values.

\[ \text{loop}, \text{time}, m \]  
Perform \( M \)-time steps.

\[ \text{time} \]  
Advance time.

\[ \text{loop}, \text{iter}, n \]  
Iteration for nonlinear dynamic solution.

\[ \text{tang}, 1 \]  

\[ \text{next} \]  

\[ \text{next} \]  

**References** for theoretical background: P. Wriggers, C. Carstensen: In preparation
EIG1

\texttt{eig1,,<n1,n2>}

'\texttt{eig1}' calculates all eigenvalues and all eigenvectors for one element. Thus an input-file with one element, no loads and no boundary conditions is necessary.

Results for the eigenvalues are printed directly, results for eigenvectors are printed directly only in case of 'n1=2'.

In case of 'n2\neq0' the calculation of the stiffness matrix is omitted. Thus the complete procedure must be \texttt{tang + eig1}.

Furthermore results can be printed via \texttt{disp,eigv} and plotted via \texttt{eigv}
The **ELSE** command may be used with a matching pair of **IF-ENDIf** commands. The expression is optional and is used to control the actions taken during the solution. If the expression is absent the commands between the **ELSE** and **ENDIf** are executed. If the expression evaluates to be positive then the commands contained between the **IF** and the **ELSE** or **ENDIf** are executed, otherwise solution continues with a check of the next **ELSE**.

At present expression is restricted to 4 characters! An extension to 15 characters is projected!

For example, the sequence

```
ZEROA
... IF,10-a tang,,1 ZEROA ELSE form solv ENDIf INCRA ... 
```

would compute a tangent, residual, and solution increment if 10-a is positive; otherwise the solution increment is computed using a previous tangent. The parameter a may be computed using a function command. For example,

```
FUNCTION ZEROA
a = 0
END
would zero the counter a.
FUNCTION INCRA
a = a + 1
END
```

would define a function which increments a.
END

end,< n1 >

The last batch macro command must be end or quit. This terminates the macro execution and returns the program to subprogram “pcontr”, which may then perform additional tasks on the same data, enter a new problem, or “stop” execution. The use of end causes a restart file to be updated for subsequent resumptions of execution with the current status preserved.

If ’n1’ is > 0 the restart–file is in ascii–mode. Thus the file can be transformed to another machine. Immediately following the end macro command any data required by statements in the “macro program” should appear when a ‘macro’ execution is performed (i.e., “batch” executions).
The **ENDIF** command is used with a matching **IF** command to terminate the control construction. For example, the sequence

ZEROA
...
IF,10-a
tang,,1
ZEROA
ELSE
form
solv
ENDIF
INCRA
...

would compute a tangent, residual, and solution increment if 10-a is positive; otherwise the solution increment is computed using a previous tangent. The parameter a may be computed using a function command. For example,

**FUNCTION** ZEROA
a = 0
END
would zero the counter a.

**FUNCTION** INCRA
a = a + 1
END

would define a function which increments a.
EPSQ

\[ \text{epsq}, < n1 > \]

Calculation of prescribed displacements \( \mathbf{V} = \mathbf{A} \mathbf{E} \).

- \( n1 = 0 \) read strain data from file defined in material 8,
- \( n1 > 0 \) read strain data from input macro \( \text{epsq} \).
ERRO

\texttt{erro,< v1 >}
\texttt{erro,save,< v1 >}

The macro command \texttt{erro} calculates results of all implemented error indicators of the finite element analysis. Currently these are 'energy-based'(1) and 'L_2-based(2)'. Results are printed here and can be plotted with \texttt{erro}.

The theoretical background is described in the Theory Manual.

The results are given with respect to a user defined error value 'v1' in percent (default: 5 %).

With option \texttt{save} it is possible to write the indicator values on a file \texttt{fres.err} for further handling within a refinement strategy.
EXIT

```
exit, < n1 >
```

The last interactive macro command must be `exit` or `quit`. This terminates the macro execution and returns the program to subprogram “pcontr”, which may then perform additional tasks on the same data, enter a new problem, or “stop” execution. The use of `exit` causes a restart file to be updated for subsequent resumptions of execution with the current status preserved.

If ’n1’ is > 0 the restart–file is in ascii–mode. Thus the file can be transformed to another machine.

For interactive execution, using `inte`, any additional data will be requested as needed.
The `ext` macro computes a singular point using an extended system of equations. The following table defines the input options:

<table>
<thead>
<tr>
<th>xxxx</th>
<th>n1</th>
<th>n2</th>
<th>n3</th>
</tr>
</thead>
<tbody>
<tr>
<td>mext</td>
<td>ε</td>
<td>iev(mext=5)</td>
<td></td>
</tr>
<tr>
<td>eps</td>
<td>exeps</td>
<td>kex</td>
<td></td>
</tr>
<tr>
<td>on</td>
<td>mext</td>
<td>ε</td>
<td>iev(mext=5)</td>
</tr>
<tr>
<td>off</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In the above table the parameters have the following meaning:

- **mext**: initialization of extended system with the following approximation of the eigenvector:
  
  \[
  \begin{align*}
  \text{mext} = 1: & \quad \phi = \frac{\{1, 1, \ldots, 1\}}{\|\{1, 1, \ldots, 1\}\|} \\
  \text{mext} = 2: & \quad \phi = \frac{u}{\|u\|} \\
  \text{mext} = 3: & \quad \phi = K_T^{-1} I \\
  \text{mext} = 4: & \quad \phi = \{K_T^{-1} I\}^{10} \\
  \text{mext} = 5: & \quad \phi = \phi_{iev} = \text{ev. no. iev from subspace iteration}
  \end{align*}
  \]

- **ε**: Perturbation factor for the approximate computation of the derivatives of the tangent stiffness \(K_T\). Default: \(\epsilon = 0 \Rightarrow \epsilon = 10^{-7}\)

- **iev**: No. of eigenvector calculated from subspace iteration

- **exeps**: Penalty parameter for the stabilization of the extended system near a singular point.

- **kex**: Equation number to be stabilized.

With `<xxxx> = off` the extended system is switched off. `<xxxx> = on` reinitializes the extended system. `<xxxx> = eps` initializes the stabilization using the penalty approach. Here, one should use an equation number `kex` which is associated with a negative diagonal element. If there is none, a good choice is to take the last equation number of the finite element mesh. The penalty factor `exeps` can be set in most cases to 1. However, depending on the material data used, a different value may be appropriate. **REMARK:** Before the extended system is initiated one step using arclength control has to be performed, see `arcl`. 
Thus the following macro commands should be used for example to compute a singular point.

```
dt,,1      Time increment, should be 1 for arcl.
prop,,1    Proportional load.
arcl,2     Initialize arclength method.
time       Advance time.
loop,,n    Iteration for one step of arclength control.
tang,,1    
next       
arcl,off   stop arc–length–method
ext,,3     Initialize extended system with option 3.
time       Advance time.
loop,,n    Iteration to compute singular point.
tang,,1    
next       
ext,off    stop extended system
arcl,on    start arc–length–method
...
```

**References**

Details for the theoretical background can be found in Wriggers, Wagner, Miehe, (1988) [32] and Wriggers, Simo (1990) [33].
**FAST**

`fast`

- The `fast` command write all input data including all profile data and solver arrays (currently only for `solv`, 2, 3 and 4) on a binary file named `ifile.sys`. Using this file via the input-macro `fast` the start procedure in `FEAP` could be much faster, especially in case of `tie` etc.

- Comment: Up to now use macro `fast` as **first macro**!
FEAS

feas,,<n1,n2,n3>

- The **feas** macro command requests the solution of 'n1' eigenvalues of a problem about the current state.
- 'n1' eigenvalues and eigenvectors are calculated in the interval 'n2' ≤ 'm' eigenvalues ≤ 'n3'. To decide which values could be used compare the output of error norms. To be successful typically 'n1' should be 'n1' > 1.5-2·'m'.
- For the efficiency it is absolutely critical to know the limits 'n2' ≤ 'm' eigenvalues ≤ 'n3'!
- The macro could be used only in the INTEL-version and only for **solv**,4 (Pardiso-Solver). Thus the solution is parallel and could be used for large systems.
- If the solution has not converged it is possible to choose more eigenvalues for the first time using **feast**, **subspace**/ or **lanczos**.
- Within a calculation the number 'n1' of eigenvalues is limited to a maximum value. This value is defined by the 'n1' of the first **feast** or **subspace**/**lanczos** iteration.
- The **feas** macro must be preceded by the specification of the tangent stiffness array using a **tang** command, and a second matrix. This can be either the mass matrix (a lumped mass by **inmas** or a consistent mass by **cmas**) or an identity.

Thus one of the following eigenvalue problems can be solved:

\[
\begin{align*}
[K_T - \omega^2M] \varphi & = 0 \\
[K_T - \omega^2MD] \varphi & = 0 \\
[K_T - \omega I] \varphi & = 0
\end{align*}
\]

- The first and second matrix must be symmetric and stored in CSR-technique, in addition the second matrix must be positive definite.
- If 'n1' is larger than the number of non-zero mass diagonals it is truncated to the actual number that exist. Whenever 'n1' is close to the number of nonzero mass diagonals one should compute the entire set since convergence will be attained in one iteration (this applies primarily to small problems).
- Eigenvectors are scaled that the maximum entry is one.
The **form** macro computes the residual for the current time and iteration of a solution. FEAP is a fully nonlinear program and computes a residual for each solution by subtracting from any applied loads the force computed for the stresses in each element, called the “stress divergence” or “internal force” term, and if the problem is dynamic the inertia forces.

At the end of each computation FEAP reports the value of the current residual in terms of the Euclidean norm, which is the square root of the sum of squares of each component of force.

If the **acel**-option is present an acceleration is computed by solving the equation:

\[
Ma_0 = G_0
\]

where \( 
M \)
 is a consistent mass (e.g., \( \text{cmas} \)) or a lumped mass (\( \text{lmas} \)) which must be available before the specification of the **form** command. \( G \) is the residual calculated by the **form** macro.

This option must be used to compute consistent accelerations for starting a transient analysis (\( \text{trans} \)) using the Newmark (or similar)-integration algorithms when the initial force or the initial displacements/velocities are specified.

If the **ener**-option is present an external energy is computed by solving the equation:

\[
\pi_a = u^T F = v^T \int_{\Omega} N^T \bar{\xi} \, d\Omega
\]

where \( u \) is the displacement vector and \( F \) the load vector. This could be used e.g. for the evaluation of influence areas for plate structures.

If the **expl**-option is present this is a part of an explicit time integration procedure, see \( \text{trans}, \text{expl} \) or \( \text{trans}, \text{expa} \). In this case the equation:

\[
Ma_{n+1} = G_{n+1}
\]

is solved, where \( M \) is a lumped mass (\( \text{lmas} \)) matrix.
The macro \texttt{four} is a set up for a fourier series solution. ’nf’ defines the number of the calculated fourier harmonic. The sign of ’nf’ influences the calculation of the displacements:

\[
\begin{array}{|c|c|c|c|}
\hline
\text{sign (nf)} & u_1 & u_2 & u_3 \\
\hline
nf > 0 & \cos nf \theta & \sin nf \theta & \cos nf \theta \\
nf = 0 & 1 & 1 & 1 \\
nf < 0 & \sin nf \theta & \cos nf \theta & \sin nf \theta \\
\hline
\end{array}
\]

The solution can be multiplied by a value defined by ’factor’. The default value is 1.

Within a fourier calculation the program uses the files \texttt{four.dis} and \texttt{four.str}.

The fourier solution needs the following macros \texttt{fsol, fsum}. 

The macro \texttt{four} is a set up for a fourier series solution. ’nf’ defines the number of the calculated fourier harmonic. The sign of ’nf’ influences the calculation of the displacements:

\[
\begin{array}{|c|c|c|c|}
\hline
\text{sign (nf)} & u_1 & u_2 & u_3 \\
\hline
nf > 0 & \cos nf \theta & \sin nf \theta & \cos nf \theta \\
nf = 0 & 1 & 1 & 1 \\
nf < 0 & \sin nf \theta & \cos nf \theta & \sin nf \theta \\
\hline
\end{array}
\]

The solution can be multiplied by a value defined by ’factor’. The default value is 1.

Within a fourier calculation the program uses the files \texttt{four.dis} and \texttt{four.str}.

The fourier solution needs the following macros \texttt{fsol, fsum}. 

The macro \texttt{four} is a set up for a fourier series solution. ’nf’ defines the number of the calculated fourier harmonic. The sign of ’nf’ influences the calculation of the displacements:

\[
\begin{array}{|c|c|c|c|}
\hline
\text{sign (nf)} & u_1 & u_2 & u_3 \\
\hline
nf > 0 & \cos nf \theta & \sin nf \theta & \cos nf \theta \\
nf = 0 & 1 & 1 & 1 \\
nf < 0 & \sin nf \theta & \cos nf \theta & \sin nf \theta \\
\hline
\end{array}
\]

The solution can be multiplied by a value defined by ’factor’. The default value is 1.

Within a fourier calculation the program uses the files \texttt{four.dis} and \texttt{four.str}.

The fourier solution needs the following macros \texttt{fsol, fsum}. 

The macro \texttt{four} is a set up for a fourier series solution. ’nf’ defines the number of the calculated fourier harmonic. The sign of ’nf’ influences the calculation of the displacements:

\[
\begin{array}{|c|c|c|c|}
\hline
\text{sign (nf)} & u_1 & u_2 & u_3 \\
\hline
nf > 0 & \cos nf \theta & \sin nf \theta & \cos nf \theta \\
nf = 0 & 1 & 1 & 1 \\
nf < 0 & \sin nf \theta & \cos nf \theta & \sin nf \theta \\
\hline
\end{array}
\]

The solution can be multiplied by a value defined by ’factor’. The default value is 1.

Within a fourier calculation the program uses the files \texttt{four.dis} and \texttt{four.str}.

The fourier solution needs the following macros \texttt{fsol, fsum}.
FSOL

\texttt{fsol}, \texttt{factor}

The macro \texttt{fsol} saves the results of a fourier series solution for the current harmonic. The solution can be multiplied by a value defined by 'factor'. The default value is 1. Displacements may be calculated by a macro \texttt{tang}.1 (?) whereas stresses are calculated automatically und this macro. Displacement field and stress field are stored for the actual harmonic on the files four\_dis and four\_str. The fourier solution needs the following macros \texttt{four}, \texttt{fsum}. 
FSUM

\texttt{fsum}, \theta

The macro \texttt{fsum} is used to sum the displacements and stresses at a given angle \( \theta \) (input in degree).

The fourier solution needs the following macros \texttt{four}, \texttt{fsol}.

In total a fourier calculation is given by

\begin{verbatim}
Loop over all harmonics
\texttt{four},nf,factor
\texttt{tang},1 (?) \texttt{fsol},nf,factor
\end{verbatim}

Results can be calculated at a given angle \( \theta \)

\begin{verbatim}
\texttt{fsum},\theta
\end{verbatim}

\textbf{Note:} The macros \texttt{four}, \texttt{fsol}, \texttt{fsum} are not tested. Thus additional work has to be done—especially for harmonic loading and associated element formulation. There exists an old 3–9 node axisymmetric FEAP–element.
GEOM

geom

The macro command geom is used to compute a geometrical matrix $K_{NL}$.

It is calculated from $K_T = K_L + K_{NL}$, where $K_L$ is derived from $K_L = K_T(v = 0)$.

The geometrical matrix is necessary for the buckling analysis, which can be done at any load state via

$$[K_L + \Lambda(K_{NL})] \phi = 0.$$

Sometimes a separation $K_{NL} = K_U + K_G$ may be possible, which lead to the 'Classical Buckling analysis':

$$[K_L + \Lambda K_G] \phi = 0$$

and the 'Linear Buckling analysis':

$$[K_L + \Lambda(K_U + K_G)] \phi = 0.$$

This separation is no longer available in FEAP.

- **Buckling analysis at initial state** with $P = P_0$

  ```
  # tang,,1 to calculate displacements
  # geom
  # subs,,n
  # P_c = \Lambda_1 \cdot P_0
  ```

  (Be sure that the displacements are small.)

- **Buckling analysis within the nonlinear regime at any other load state.**

  ```
  # based on an equilibrium state \bar{v}, \bar{\lambda}P
  # geom
  # subs,,n
  # P_c = \Lambda_1 \cdot (\bar{\lambda}P)
  ```

- **Remarks:**

  1. Do not use in combination with single displacement control!
  2. Find the lowest positive eigenvalue ($\bar{\lambda}$)
  3. Note that the stiffness matrix is destroyed after geom. Thus go on with e.g. tang,,1.
FEAP — MACRO COMMAND USER MANUAL

HELP

help

- The use of the help macro will produce a list of the currently implemented macro commands.
- This feature is useful only in an interactive mode of macro execution.
- If additional information is required for a specific command it is necessary for the user to consult the MACRO command users manual or to type
  
  help, macroname

  where macroname is the name of the command for which information is required. The available information will be given on the screen.
The use of the `hist` macro permits the user to keep a history of the previously executed macro commands and use this history to reexecute specific commands. The `hist` macro has several different modes of use which permit easy control of the execution of macro commands while in an interactive mode (use is not recommended in a batch `macr` execution). The following options are available:

<table>
<thead>
<tr>
<th><code>clab</code></th>
<th><code>n1</code></th>
<th><code>n2</code></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>read</code></td>
<td></td>
<td></td>
<td>Input the list of macro commands which were 'saved' in a previous execution. Warning, this command will destroy all items currently in the 'history' list, hence it should be the first command when used.</td>
</tr>
<tr>
<td><code>save</code></td>
<td></td>
<td></td>
<td>Save the previous 'history' of macro commands which have been 'added' to the 'history' list on the file named 'Feap.his'.</td>
</tr>
<tr>
<td><code>add</code></td>
<td></td>
<td></td>
<td>Add all subsequent macro commands executed for the current analysis to the 'history' list. (default)</td>
</tr>
<tr>
<td><code>noad</code></td>
<td></td>
<td></td>
<td>Do not add subsequent macro commands executed to the 'history' list.</td>
</tr>
<tr>
<td><code>list</code></td>
<td><code>n1</code></td>
<td><code>n2</code></td>
<td>List the current 'history' of macro statements. 'n1' to 'n2' , (default is all in list).</td>
</tr>
<tr>
<td><code>edit</code></td>
<td><code>n1</code></td>
<td><code>n2</code></td>
<td>Delete items 'n1' to 'n2' from current 'history' list.</td>
</tr>
<tr>
<td><code>xxxx</code></td>
<td><code>n1</code></td>
<td><code>n2</code></td>
<td>Reexecute macro commands 'n1' to 'n2' in the current 'history' list. (note: <code>xxxx</code> may be anything not defined above for <code>clab</code> including a blank field.</td>
</tr>
</tbody>
</table>

Use of the `hist` option can greatly reduce the effort in interactive executions of MACRO programs. Since it is not possible to name the file which stores the 'history' of macro commands, it is necessary for the user to move any files needed at a later date to a file other than 'Feap.his' before starting another analysis for which a 'history' will be retained. Prior to execution it is necessary to restore the list to file 'Feap.his' before a `hist.read` command may be issued.

Note that the 'history' of macro commands will not be saved in 'Feap.his' unless a macro command `hist.save` is used. It is, however, possible to use the `hist` option without any `read` or `save` commands.

Remark:

With the input '!' it is possible to repeat the last macro.
The **iden** macro command is used to specify an identity matrix.

- In general it may be used in conjunction with an eigen computation to compute the eigenpairs of a stiffness matrix for example to test the correct rank of the stiffness matrix. When used in this mode all boundary restraints must be omitted and a shift used to compute any “zero” eigenvalues.
- The macro **iden** is often used for nonlinear buckling analysis.
- \([K_T - \omega 1] \varphi = 0\).
- An alternative buckling analysis can be found under the macro command **geom**.
- When ‘n1’ and ‘n2’ are specified they indicate the node range (i.e., ‘n1’ to ‘n2’) for which the identity matrix is to be specified.
The **IF** command must be used with a matching **ENDIF** command. Optionally, one or more **ELSE** commands may be included between the **IF-ENDIF** pair. The expression is used to control the actions taken during the solution. If the expression evaluates to be positive then the commands contained between the **IF** and the **ELSE** or **ENDIF** are executed, otherwise solution continues with a check of the next **ELSE**.

At present expression is restricted to 4 characters! An extension to 15 characters is projected!

For example, the sequence

```
ZEROA
...
IF,10-a
tang,,1
ZEROA
ELSE
form
solv
ENDIF
INCRA
...
```

would compute a tangent, residual, and solution increment if 10-a is positive; otherwise the solution increment is computed using a previous tangent. The parameter a may be computed using a function command. For example,

```
FUNCTION ZEROA
a = 0
END
would zero the counter a.
FUNCTION INCRA
a = a + 1
END
```

would define a function which increments a.
The macro command `iimp` is used to generate user-defined values/fields on element level. Each element computes a contribution when 'isw' is 20.

For example a stochastical imperfection field can be calculated to disturb perfect homogeneous situations.
Non–zero initial displacements or rates (e.g., velocities) for a dynamic solution may be specified using the \texttt{init} macro command. The values for any non–zero vector are specified after the \texttt{end} macro command for \texttt{batch} executions and may be generated in a manner similar to nodal generations in the mesh input. For interactive execution prompts are given for the corresponding data. Accordingly, the vectors are input as:

\begin{verbatim}
n1,ng1,v1–1, . . . ,v1–ndf
n2,ng2,v2–1, . . . ,v2–ndf
etc.
\end{verbatim}

where, ‘n1’ and ‘n2’ define two nodes;

\begin{itemize}
  \item ‘ng1’ defines an increment to node ‘n1’ to be used in generation;
  \item ‘v1–1’,‘v2–1’ define values for the first degree of freedom at nodes ‘n1’, ‘n2’, respectively; etc. for the remaining degree of freedoms.
  \item Generated values are linearly interpolated using the ‘v1’ and ‘v2’ values; etc. for the remaining degree of freedoms.
  \item Note that ‘ng2’ is used for the next pair of generation records.
  \item If a value of ‘ng1’ or ‘ng2’ is zero or blank, no generation is performed between ‘n1’ and ‘n2’.
  \item Do not generate over nodes with boundaries.
\end{itemize}

Initial accelerations are basically calculated using the macro \texttt{form,acel}. For special cases it could be necessary to prescribe these values directly. This could be done via \texttt{init,acce}. Input is similar to \texttt{init,rate}.
JINT

\[ \texttt{jint}, <n1,n2,n3> \]
\[ \texttt{jint.all} \]
\[ \texttt{jint.eps,n1,n2} \]

- Nodal material forces may be computed for all nodes in the problem and reported for nodes 'n1' to 'n2' at increments of 'n3' (default = 1). If 'n2' is not specified then only the value for node 'n1' is output. When both 'n1' and 'n2' are not specified only values for node '1' is output.
- All material forces may be output with the \texttt{jint.all} command.
- With the \texttt{jint.eps} command material forces for a node n are printed only if \(|jint(n1, n)| > n2\).
The **lamb** macro is used to compute approximately buckling load factors $\Lambda$ from eigenvalues $\omega$.
With the parameter 'n1' the number $i$ of eigenvalue has to be specified (default=1).
With **lamb,all** the buckling loads are calculated for all available eigenvalues.

The calculation bases on the formula

$$\Lambda_i = \frac{\varphi_i^T K_L \varphi_i}{\varphi_i^T K_L \varphi_i - \omega_i \varphi_i^T \varphi_i}$$

**Reference**
Details for the theoretical background can be found in Wagner (1995) [29].
LAN

\texttt{lan},<n1,n2,n3>
\texttt{lan,prin},<n1,n2,n3>

- The \texttt{lan} macro command requests the solution of 'n1' eigenpairs of a problem about the current state.
- Typically 2·'n1' eigenvalues and eigenvectors are calculated. Only the first 'n1' results could be used (print and plot), see the output of error norms.
- To find a solution a working space of 'n2' (default = 50) eigenvectors is used.
- All eigenvalues are computed until two subsequent iterations produce values which are accurate to the current defined tolerance. The iteration behaviour can be controlled by the parameters 'tols' (tolerance for Lanczos iteration) by 'n3' as follows

\begin{center}
\begin{tabular}{|c|c|}
  \hline
  'n3' = 0 & 'tols' = 1.d-12 (default) \\
  'n3' > 0 & 'tols' = tol \\
  \hline
\end{tabular}
\end{center}

The default value for tol is 1.d-16, see the tol macro.
- If the solution has not converged it is possible to compute further iterations (in the interactive version).
- The \texttt{lan} macro must be preceded by the specification of the tangent stiffness array using a \texttt{tang} command, and a second matrix. This can be either the mass matrix (a lumped mass by \texttt{lmass} or a consistent mass by \texttt{cmass}) or a 'buckling' matrix (an identity matrix by \texttt{iden}). Thus one of the following eigenvalue problems can be solved:

\[ [K_T - \omega^2 M] \varphi = 0 \]
\[ [K_T - \omega^1] \varphi = 0 \]

- Note that the smallest 'n1' eigenvalues and eigenvectors are computed with reference to the current "shift" specified on the tang command.
- If 'n1' is larger than the number of non-zero mass diagonals it is truncated to the actual number that exist. Whenever 'n1' is close to the number of nonzero mass diagonals one should compute the entire set since convergence will be attained in one iteration (this applies primarily to small problems).
- When a large number of eigenvalues should be calculated the subspace iteration may be very time consuming. Then the lanczos iteration is advantageous.
- In case of large systems often iterative solvers are used. Then the subspace iteration is very time consuming whereas the lanczos iteration can be used without any problems.
- Use of the prin option in second place produces additional output for testing. This option is recommended for small problems only.
- Eigenvectors are scaled that the maximum entry is one.
LINE

line

The **line** macro command can be used to restrict parts of the code to linear calculations or to switch between linear and nonlinear calculations only by setting a logical variable 'linear'. The default value of 'linear' is false. Thus nonlinear calculations are permitted.

Up to now the execution of the convergence test by **tol** is implemented.

A change of the variable 'linear' can be done with the macro **nonl**.

Thus:

**line** allows linear calculations whereas

**nonl** allows nonlinear calculations (default).
The macro command `lmas` is used to compute a lumped (i.e., a diagonal) “mass” matrix. Each element computes a contribution to the lumped mass in the array `p` when `isw` is 5 and `imtyp` is 1.

A lumped mass or a consistent mass (see macro command `cmas`) may be used for transient solutions computed using the Newmark method (see macro command `trans`). Both may also be used for eigencomputations (see macro command `subs`).

---

**LMAS**

- **lmas**

The macro command `lmas` is used to compute a lumped (i.e., a diagonal) “mass” matrix. Each element computes a contribution to the lumped mass in the array `p` when `isw` is 5 and `imtyp` is 1.

A lumped mass or a consistent mass (see macro command `cmas`) may be used for transient solutions computed using the Newmark method (see macro command `trans`). Both may also be used for eigencomputations (see macro command `subs`).
The **loop** macro command is used with the **next** command to repeat the set of macro commands between them 'n1' times.

The four characters 'xxxx' may be used to describe the action taken by the loop, e.g., **loop**,time,5 can indicate that the looping is for time steps and is to be performed for 5 steps. Alternatively, the 'xxxx' may be left blank.

During interactive executions, **loop — next** commands are not executed until the **next** macro command is issued. In this way a set of macro statements may be grouped together and 'batch' executed.

The **loop — next** commands may be nested to a depth of 8.
The macro commands \texttt{mac\textsubscript{n}} ($n = 1,...,5$) are currently not implemented. These commands refer to subroutine calls in SR \texttt{pmacr}. They can be used by a programmer who wants to add new macro features to \texttt{FEAP}.

With the command \texttt{mac\textsubscript{n},name} the macro name \texttt{mac\textsubscript{n}} is reset to \texttt{name} which can then be used during the analysis. \texttt{name} is any four character name. It is not allowed to use already existing macro names like e.g. \texttt{tang}.
The complete documentation is available using the macro \texttt{man}.
The macro command \texttt{mate} can be used to modify the material relations of each element. Several options are possible.

- \texttt{mate.def} defines rules for a modification of the material relations. This option is up to now not active.
- \texttt{mate.save} makes a copy of the actual material relations. This is done at the beginning by default.
- \texttt{mate.org} makes a reset of material relations based on \texttt{mate.save}.
- \texttt{mate.new,<v1,n2,n3>} modifies the material relations. To do this, an error calculation has to be done in advance via \texttt{erro}.

'v1' defines the value to decide for a change (def.=1.0), 'n2' defines the no. of norm to be used (def.=1).

Typically material '1' is modified to material '2. Here, the \texttt{actual} material number is changed to material number 'n3' (def.=2).
The use of the `mesh` macro command permits the redefinition of some of the mesh data.

Nodal forces may be redefined during macro execution to consider additional loading distributions. In addition, nodal coordinates, values of temperatures, angles of sloping boundaries, constants, material numbers on elements, and material properties may be redefined.

It is not permitted to change the boundary restraint codes or the element connection data as these change the profile of the resulting equations and this is not recomputed to permit the necessary redistribution of memory use.

The description of data input can be found in the MESH command users manual. In interactive mode online help will be given in the MESH mode with the commands `help` or `help, macroname`. 
The use of the **newf** macro command will set a fixed pattern of nodal forces and displacements to the values of the current pattern in “forced” boundary loadings plus the previous “fixed” pattern. That is:

\[ f_0(i,n) \leftarrow f(i,n) \times \text{prop}(t) + f_0(i,n) \]

where \( f_0(i,n) \) is the “fixed” pattern loads, \( f(i,n) \) is the pattern specified in “forced” boundary loads, and \( \text{prop}(t) \) is the current value of the proportional loading at the current time ‘t’.

**Remarks:**

- When execution is initiated the values in \( f_0(i,n) \) are all zero.
- Initial values could be set on input level via the macro **loa0**.
- **NOTE** at restart \( f_0(i,n) \) again will have only the initial values set via **loa0**. Caution must be exercised at any restart where **newf** had been used in generating the results.
The `next` macro command must be used in conjunction with a `loop` command. The `loop`—`next` pair are used to repeat the execution of a set of macro commands. The `loop` appears first, followed by one or more macro commands then a `next` command.

If one makes a mistake and start the loop with the macro `next`, or type one `next` more than necessary, FEAP waits on a macro to close the group of commands. In this situation type `loop`. Thus the mistake can be corrected without any calculation.

The `loop`—`next` commands may be nested to a depth of 8.

If desired, the 'xxxx' may be used to describe the type of `next` which is being closed, i.e., `next`,time would indicate the end of a time loop.

During interactive executions, `loop`—`next` commands are not executed until the `next` macro command is issued.

In this way a set of macro statements may be grouped together and 'batch' executed.
The **nonl** macro command can be used to switch between linear and nonlinear calculations by setting a logical variable 'linear'. The default value of 'linear' is false which allows nonlinear calculations.

A change of the variable 'linear' can be done with the macro **line**.

Thus:
- **line** allows linear calculations whereas
- **nonl** allows nonlinear calculations (default).
NOPR

nopr

- The use of the nopr macro command will discontinue the output of a statement (or statements) describing action taken during each macro command execution. However, the residual which shows the convergence behaviour in nonlinear iterations is still output. This holds also for the load level reported by the arc length scheme. Furthermore, plot results and element outputs will still be reported.

- The use of prin will cause the output of macro execution descriptions to again be reported.

- The default value is prin at start of each macro program execution.
The use of the `para` command permits the input of data parameters during execution. Typically these parameters are set during the data input phase to vary the input values. For example, parameters may be set and used during proportional loading table inputs. Only 1 or 2 character parameters are permitted and should be lower case letters and numerals (first character must be a letter) only. All further details of the macro `para` are described in MESH-Input by the mesh-macro `para`.

In Macro-mode two options for the use of the `para` command are allowed.

- Directly in Macro-modus

  Possible Examples:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PARA,a=200</code></td>
<td>sets the parameter a to the value of a=200.</td>
</tr>
<tr>
<td><code>PARA,a+10</code></td>
<td>sets the parameter a to the value of a=a+10.</td>
</tr>
<tr>
<td><code>PARA,a-10</code></td>
<td>sets the parameter a to the value of a=a-10.</td>
</tr>
<tr>
<td><code>PARA,a=0</code></td>
<td>sets the value of a = 1,2,3,4,5 repeatedly during the loop.</td>
</tr>
<tr>
<td><code>....</code></td>
<td></td>
</tr>
<tr>
<td><code>LOOP,,25</code></td>
<td></td>
</tr>
<tr>
<td><code>....</code></td>
<td></td>
</tr>
<tr>
<td><code>PARA,a+1</code></td>
<td></td>
</tr>
<tr>
<td><code>....</code></td>
<td></td>
</tr>
<tr>
<td><code>NEXT</code></td>
<td></td>
</tr>
<tr>
<td><code>END</code></td>
<td></td>
</tr>
</tbody>
</table>

  **Note:** At present the expression after macro `para` is restricted to 4 characters! An extension to 15 characters is projected!

  Use of LIST will display parameters and values for all letters set previously to non-zero values.

- in Batch -Modus

  Use of

  ```
  BATCh
  ....
  PARA
  ....
  END
  a = 12
  b = a/9
  ```

  leads to an input of parameters set from data after the END command.
The macro command `parv` can be used to save data for postprocessing with the program PARAVIEW, see [http://www.paraview.org/](http://www.paraview.org/).

Alternatively the program TECPLOT could be used, using the macro `tec`.

More than one material is possible. Associated elements must have the same type and could differ only in the number of nodes.

- **parv.init, n1,<n2>**
  
  opens a file `Rname.pvd` including a group of files `Rname.mxx.tyyyy.vtu` for the postprocessing data. Here `mxx` defines different materials with `1 ≤ xx ≤ 99` and `tyyy` results at time-steps `1 ≤ yyyyy ≤ 9999`. A separation into materials is introduced for `n2 = 0` (default). No separation is chosen automatically for `nummat > 99`.

  Furthermore initial data like nodal coordinates and elements are written.

  With `n1` the type of used element is defined.

<table>
<thead>
<tr>
<th>No.</th>
<th>Element type</th>
<th>ndf</th>
<th>ndm</th>
<th>nel</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>beam-2d</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>axishell</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>beam-3d</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>incl. warping</td>
</tr>
<tr>
<td>11</td>
<td>beam-3d</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>plane stress</td>
<td>2</td>
<td>2,3</td>
<td>3,4,8,9</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>plane strain</td>
<td>3</td>
<td>2</td>
<td>4,8,9</td>
<td>3 dofs for cosserat theory</td>
</tr>
<tr>
<td>13</td>
<td>plane strain</td>
<td>1</td>
<td>2</td>
<td>4,8,9</td>
<td>2D Phase field</td>
</tr>
<tr>
<td>14</td>
<td>plane strain</td>
<td>5</td>
<td>2</td>
<td>4,8,9</td>
<td>2D Phase field</td>
</tr>
<tr>
<td>3</td>
<td>plate</td>
<td>3</td>
<td>2,3</td>
<td>3,4,8,9</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>shell-5</td>
<td>5</td>
<td>3</td>
<td>4,9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>shell-6</td>
<td>6</td>
<td>3</td>
<td>4,9</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>shell-7</td>
<td>7</td>
<td>3</td>
<td>4,9</td>
<td>incl. warping dof of beam</td>
</tr>
<tr>
<td>5</td>
<td>solid</td>
<td>3</td>
<td>3</td>
<td>8,20,27</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>solid</td>
<td>4</td>
<td>3</td>
<td>8,20,27</td>
<td>Add. dof for th.-mech. coupling</td>
</tr>
<tr>
<td>7</td>
<td>solid</td>
<td>6</td>
<td>3</td>
<td>8,20,27</td>
<td>6 dofs for cosserat theory</td>
</tr>
<tr>
<td>15</td>
<td>solid</td>
<td>7</td>
<td>3</td>
<td>8,20,27</td>
<td>3D Phase field</td>
</tr>
<tr>
<td>16</td>
<td>solid</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

- **parv.next,<n1>**

  writes all current data on files `Rname.mxx.tyyyy.vtu` in each time step. These are: nodal coordinates, elements, nodal displacements, (velocities, accelerations in case of `trans`), deformed mesh (displacements `u1, u2, [u3]`) and nodal stresses which have to be calculated by `stre.node` before! `n1 = 0`(default)writes results for the `next` step, whereas `n1 = t` writes results for step `t + 1`.

- **parv.eigv,n1,n2** opens a file `Rname.EVxx.vtu` for the postprocessing of eigenvector `xx = n2` and element type `n1`. Components of the eigenvector can be plotted similar to nodal displacements, the shape of the eigenvector is presented as deformed mesh (displacements `u1, u2, [u3]`).
- In case of polar coordinates, `pola` has to be used in plot-mode before.

- The node numbering (starting from 0, but similar to FEAP) of the associated elements in Paraview is presented in the following table.

<table>
<thead>
<tr>
<th>1D</th>
<th>Linear</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VTK_LINE (=3)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2D</th>
<th>Linear</th>
<th>Quadratisch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VTK_LINE (=5)</td>
<td>VTK_QUAD (=9)</td>
</tr>
<tr>
<td></td>
<td>VTK_QUADRATIC_QUAD (=23)</td>
<td>VTK_BIQUADRATIC_QUAD (=28)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3D</th>
<th>Linear</th>
<th>Quadratisch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VTK_TETRA (=10)</td>
<td>VTK_HEXAHEDRON (=12)</td>
</tr>
<tr>
<td></td>
<td>VTK_QUADRATIC_HEXAHEDRON (=25)</td>
<td></td>
</tr>
</tbody>
</table>
A non-linear calculation could be written completely to PARAVIEW with

\begin{verbatim}
  stre, node
  parv, init, n1

  prop,,1
  dt,,1
  loop,,n-time-steps
    time
    loop,,m-iteration steps
      tang,,1
      next
  stre, node
  parv, next,,10
  next
\end{verbatim}
The **paus** macro command can be used in interactive macro mode to stop the iteration behaviour in case of a diverging solution. Within each iteration step the actual internal energy is compared to the value at beginning of iteration. If the actual value is 100 times larger then the first one FEAP asks you if the iteration should be continued:

- 'y' or 'Y' continues iteration
- 'n' or 'N' terminates loop

An associated sequence of macro solutions looks like:

- **time**
- **loop**,n
- **tang**,1
- **paus**
- **next**
With the command \texttt{pbcg} the parameter of the Pre-conditioned bi-Conjugated Gradient method (PBCG) for the iterative solution of a linear system of equations could be modified. This solver is chosen in the input file via \texttt{solv,5}.

With \texttt{<xxxx>} = iter or 'blank' the parameters \texttt{n1,n2,v3} are set.

- The parameter 'n1' (Def = 150) denotes the maximum numbers of iterations. Typical values are in the range of 50-1000.
- The parameter 'v2' (Def = $1 \cdot 10^{-08}$) sets the tolerance value.
- The parameter 'n3' (Def = 1) sets the type of tolerance criterion (1-4).

Note that for the successful use of the solver a preconditioning is necessary based on the macro \texttt{prco}.
With the `pgmr` command the parameter of the Pre-conditioned Generalized Minimum Residual method (PGMRES) for the iterative solution of a linear system of equations could be modified. This solver is chosen in the input file via `solv,6`

With `<xxxx>` = iter or 'blank' the parameter n1,v2 are set.

- The parameter 'n1' (Def = 150) denotes the maximum numbers of iterations. Typical values are in the range of 10-500.
- The parameter 'v2' (Def = $1 \cdot 10^{-08}$) sets the tolerance value.

Note that for the successful use of the solver a preconditioning is necessary based on the macro `prco`. 

```
pgmr,<xxxx>,n1,v2
```
PLOT

\texttt{plot.xxxx, <n1,n2,n3>}

or

\texttt{plot}

\texttt{xxxx, <n1,n2,n3>}

\texttt{....}

\texttt{end}

- In FEAP, screen and some types of hard copy plots may be made for several quantities of interest. A \texttt{plot} statement must be specified to initiate graphics outputs.

- Two possibilities to use the \texttt{plot} macros exist.

  # 1.) Type \texttt{plot} and FEAP switches to graphics mode ('Plot mode') and the prompt 'Plot >' will be displayed. At this time, all quantities \texttt{xxxx} and the 'n1', 'n2', and 'n3' values may be specified. A return to the macro mode can be done by \texttt{xxxx = end} or \texttt{= q}.

  # 2.) Alternatively, a \texttt{plot.xxxx, n1,n2,n3'} command may be issued while in macro execution mode (this is the only option for batch executions).

- On terminals where only one plane is visible (text or plot) the plot image may disappear after performing a plot action (this is what happens when using a GraphOn). In this case it may be necessary to press a key on the terminal to switch between graphics and text mode – the image is retained between changes in mode! If FEAP runs on a PC the switch is done automatically.

The following values may be used for the quantity \texttt{xxxx}:

<table>
<thead>
<tr>
<th>aacc</th>
<th>acce</th>
<th>adis</th>
<th>aeig</th>
<th>aeve</th>
<th>angl</th>
<th>avel</th>
<th>axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>back</td>
<td>base</td>
<td>bord</td>
<td>boun</td>
<td>cart</td>
<td>ceig</td>
<td>cent</td>
<td>clip</td>
</tr>
<tr>
<td>colo</td>
<td>copy</td>
<td>defm</td>
<td>defo</td>
<td>dimp</td>
<td>disp</td>
<td>dmag</td>
<td>dplo</td>
</tr>
<tr>
<td>draw</td>
<td>eigi</td>
<td>eigv</td>
<td>elem</td>
<td>end</td>
<td>eplo</td>
<td>erro</td>
<td>evan</td>
</tr>
<tr>
<td>evex</td>
<td>fact</td>
<td>flux</td>
<td>forc</td>
<td>fram</td>
<td>hide</td>
<td>hids</td>
<td>hmsb</td>
</tr>
<tr>
<td>init</td>
<td>isec</td>
<td>ints</td>
<td>isom</td>
<td>jint</td>
<td>line</td>
<td>link</td>
<td>load</td>
</tr>
<tr>
<td>logo</td>
<td>magn</td>
<td>man</td>
<td>matn</td>
<td>maxi</td>
<td>mesh</td>
<td>mono</td>
<td></td>
</tr>
<tr>
<td>move</td>
<td>movi</td>
<td>ndii</td>
<td>node</td>
<td>outl</td>
<td>pele</td>
<td>pers</td>
<td>pdis</td>
</tr>
<tr>
<td>plof</td>
<td>pnod</td>
<td>pola</td>
<td>prin</td>
<td>pris</td>
<td>prof</td>
<td>quit</td>
<td>reac</td>
</tr>
<tr>
<td>rmsh</td>
<td>resi</td>
<td>rot0</td>
<td>rot1</td>
<td>rot2</td>
<td>rot3</td>
<td>rotm</td>
<td>rplo</td>
</tr>
<tr>
<td>rsum</td>
<td>scal</td>
<td>sect</td>
<td>show</td>
<td>size</td>
<td>slee</td>
<td>splo</td>
<td>stre</td>
</tr>
<tr>
<td>str1</td>
<td>symm</td>
<td>text</td>
<td>tie</td>
<td>tplo</td>
<td>traj</td>
<td>ueig</td>
<td>velo</td>
</tr>
<tr>
<td>wipe</td>
<td>xsca</td>
<td>zoom</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• The action to be taken by each command is described in the following sections of the PLOT Command Users Manual. Furthermore in the interactive Plot mode online help is available by typing `help` which gives information on the available plot macros and `help,xxxx` which gives information on the plot macro `xxxx`.
POLA

\texttt{pola,,n1}

Displacements, velocities, accelerations and eigenvectors can be plotted in cartesian coordinates (\texttt{n1} = 0, default) or in polar coordinates (\texttt{n1} = \texttt{ik} = 12,13,23). \texttt{i} and \texttt{k} describe the plane in which the transformation should be done. Afterwards the component \texttt{i} describes the radial e.g. displacement whereas the component \texttt{k} describes the e.g. tangential displacement.

The plot is performed in standard manner via e.g. the \texttt{disp} – macro.
The macro command `post` may be used to save data for postprocessing.

- `post,init,<n1>` opens a file `Rname.pos` for the postprocessing data. Furthermore general informations, the nodal coordinates and element connections are written on `Rname.pos`. When `tie` is used different output is possible.
  - In case n1 = 0 the output is:
    1) list of numnp nodes. Tied nodes have coordinates x1=-999.0
    2) list of numel elements. Tied nodes are here not used.
  - Thus the resulting mesh has only additional(tied) nodes.
  - In case n1 = 1 nodes are renumbered. Total number of nodes is numnpn = numnp - number of tied nodes. Furthermore the list of nodes for each element is modified.
    1) list of renumbered numnpn nodes. Tied nodes do not occur.
    2) list of numel elements with new node numbers. Tied nodes do not occur.

- `post,disp` writes the current nodal displacements on file `Rname.pos`.

- `post,react` writes the current nodal reactions on file `Rname.pos`. Calculate reactions directly before `post,react`.

- `post,eigv,n1` writes the calculated eigenvector n1 on file `Rname.pos`.

- `post,stre` writes the current nodal stresses on file `Rname.pos`.

- `post,warp,<n1>` writes the current nodal warping values on file `Rname.pos`. These values are calculated with `Element 12` using `stre,node`.
  - n1=0,1 store $\tilde{w}$
  - n1=2 store $\bar{w}$

- `post,clos` closes the file `Rname.pos`. 
PRIN

prin
prin,<xxxx>

- The use of the prin macro will cause a statement (or statements) describing action taken during each macro command execution to be reported on the output.
- The use of nopr will discontinue the output of macro execution descriptions (except actual plots).
- The default value is prin.
- The specification of:
  
  # xxxx = tang
  # xxxx = utan
  # xxxx = cmas
  # xxxx = lmas
  # xxxx = geom
  # xxxx = damp
  # xxxx = resi

will output the diagonal entries for the specified array for solv.0. Other solvers may give output for the first n stored elements. This may be useful in debugging elements, etc.

- Be sure that you have calculated the specified arrays directly before! For example with the macros
  
  # iden
  # lmas
  # prin.iden
  # prin.lmas

only the elements of lmas are printed.

- The specification of:
  
  # xxxx = eigv

will print the actual eigenvalues.

- The specification of:
  
  # xxxx = off

will suppress the following output
  
  * Echo the macro command and the associated options
  * Echo the output: computation time / prop. load value
  * Echo the output: residual / actual iteration / max no. of iterations

# This option can be redefined (default) by
xxxx = on
PRCO

\texttt{prco, <xxxx>, n1, v2, n3}

With the \texttt{prco} command the type of preconditioning for the iterative solvers (PBCG) and (PGMRES) could be chosen or modified.

With \texttt{ith <xxxx> = iter or 'blank'} the parameter \texttt{n1}, \texttt{v2}, \texttt{n3} are set.

- The parameter \texttt{'n1'} (Def = 3) defines the method of preconditioning. At present four methods are available:
  # 'n1'=1 uses a unit matrix. \( \mathbf{M} = \mathbf{1} \). This option should not be used.
  # 'n1'=2 uses the diagonal entries of the stiffness matrix. \( \mathbf{M} = \left[ \frac{1}{K_{ii}} \right] \). This option is very fast and works well only for 3D-problems with same length values.
  # 'n1'=3 uses an incomplete LU factorization of \( \mathbf{K} \) with dropout and without pivoting which could be very time consuming for large scale problems.
  # 'n1'=4 uses an incomplete LU factorization of \( \mathbf{K} \) with level k fill-in which could be very time consuming for large scale problems.

- With \texttt{v2} and \texttt{n3} parameters for preconditioner 3 are set.
  # With \texttt{'v2'} a threshold value tol for \( \mathbf{L} \) and \( \mathbf{U} \) is set. Any element whose size is less than some tolerance (relative to the norm of current row) is dropped. As recommendation use values in the range of \( 1 \cdot 10^3 \) to \( 5 \cdot 10^3 \) (Def = \( 1 \cdot 10^3 \)).
  # With \texttt{'n3'} the number of elements lfil in a row will be defined. The higher lfil the more reliable the code is but one looses efficiency. Recommendation is lfil = 5 to 10 (Def. = 5).

- With \texttt{'n3'} parameters for preconditioner 4 are set.
  # With \texttt{'n3'} the number of fill-ins will be defined. The higher lfil the more reliable the code is but one looses efficiency. Recommendation is lfil = 2 to 3 (Def. = 5).

Remarks:

- In the limit with lfil = neq and tol = 0 preconditioner 3 will yield the same factors as a Gaussian elimination without pivoting. Thus no iteration is necessary for the iterative solver for the solution which is nonsense.

- Use always preconditioner 3,4 especially for 2D-problems.
The procedure command is used for the repeated input of certain macro sequences. It can be seen as a user defined macro which contains a number of FEAP macros, which can be defined before an analysis. Every time this macro is used the whole sequence of macros is done.

A procedure is created during a macro analysis by entering the command:

```
proc, name, <n1,n2,n3>
```

- `name` is any 1-8 character alphanumeric identifier which specifies the procedure name (the first 4 characters must not be the same as an existing macro command name!)
- `n1,n2,n3` are any 1 to 4 character parameter names for the procedure, which can be used everywhere. The parameters are optional and may be blank.
- A procedure is saved in a file with the extender '.pcd'.
- On HP-WorkStations the procedure has to be written in the FEAP dialog window. Here, the procedure is terminated by using an `end` macro command.
- On PC’s a procedure editor supports the input of the procedure. Here, the first line of the procedure has to be `v1,v2,v3` (see above), `end` is not necessary.
- Furthermore the procedure can be written with any editor.
- FEAP assumes that procedures are in the same directory as the input file of the current problem. With the command `proc, path` this can be changed to any other directory.

**Example:**

For example the procedure for a load step in a nonlinear static analysis - named step.pcd - may be defined by:

```
Macros
n1,n2,n3
time
loop,,n1	
tang,,1
next
disp,,n2
stre,,n3
reac,all
(end)
```

Thus `step,n1,n2,n3` leads to the following actions: With `time` the load is updated; then ‘n1’ iterations will be performed. Finally the displacements are shown for node ‘n2’, the stresses for element ‘n3’ and the reactions for all nodes.
In the solution of transient or quasi–static problems in which the time command is used to describe each new time state the loading may be varied proportionally. At each time the loading applied will be computed from:

\[ F(i,t) = f0(i) + f(i) \text{prop}(t) \]

where \( f0(i) \) is a fixed pattern which is initially zero but may be reset using newf (see macro manual on newf); \( f(i) \) are the “forced” nodal conditions defined during mesh input or revised during a 'MESH' macro command; and \( \text{prop}(t) \) is the value of the proportional loading at time ‘t’.

The specific proportional loading is defined by specifying one record for each of the ‘n1’ prop–cards (default for ‘n1’ is 1, maximum is 10) with the following data:

- type, \( k \), \( t_{\text{min}} \), \( t_{\text{max}} \), \( a(i), i = 1,4 \)

Currently three different types of proportional loading may be specified for all time values between \( t_{\text{min}} \) and \( t_{\text{max}} \):

1. Type 1 defines a function by:
   \[ \text{prop}(t) = a(1) + a(2) \cdot (t - t_{\text{min}}) + a(3) \cdot \sin[a(4) \cdot (t - t_{\text{min}})]^k \]

2. Type 2 defines a “sawtooth” loading with input data: \( k, a(1), a(2) \), see the following figure.

3. Type 3 defines a polynomial by:
   \[ \text{prop}(t) = a(1) + a(2) \cdot (t - t_{\text{min}}) + a(3) \cdot (t - t_{\text{min}})^2 + a(4) \cdot (t - t_{\text{min}})^3 \]

The default values are

- \( n1 = 1 \)
- \( \text{type} = 1 \)
- \( k = 1 \)
- \( t_{\text{min}} = 0 \)
- \( t_{\text{max}} = 1.e8 \)
- \( a(1) = 0 \)
- \( a(2) = 1 \)
- \( a(3) = 0 \)
- \( a(4) = 0 \)

Thus the default load function

\[ f(i,t) = f0(i) + f(i) \cdot t \]

is given by the input prop, <cr>

Within the defined time interval \( t_{\text{min}} \leq t \leq t_{\text{max}} \) the load factor is calculated from all defined ‘n1’ prop–cards:

\[ \text{prop}(t) = \sum_{i=1}^{nprop} \text{prop}_i(t). \]

An example how to use is presented in the application manual part prop.
QUIT

quit

The last macro command may be **quit**, or just **q**. This terminates the macro execution and returns the program to subprogram “pcontr”, which may then perform additional tasks on the same data, enter a new problem, or “stop” execution.

The **quit** macro causes termination of execution without writing the restart files (they remain the same as at the beginning of execution).
**REAC**

- Nodal reactions may be computed for all nodes in the problem and reported for nodes \('n1'\) to \('n2'\) at increments of \('n3'\) (default = 1). If \('n2'\) is not specified then only the value for node \('n1'\) is output. When both \('n1'\) and \('n2'\) are not specified only values for node \('1'\) is output.

- All reactions may be output with the `reac,all` command.

- With the `reac,eps` command reactions for a node \(n\) are printed only if \(|reac(n1,n)| > n2\).

  With the `reac,sigq` command stresses are calculated on a RVE. Results are only valid for equilibrium state! Macro is implemented only for 3D-case.

\[
S = \frac{1}{V} F_b
\]

- In addition to the “reaction” at each degree of freedom an equilibrium check is performed by summing the values for each degree of freedom over all nodes in the analysis. The sum of the absolute value of the reaction at each degree of freedom is also reported to indicate the accuracy to which equilibrium is attained.

- NOTE that problems with rotational degrees of freedom or in curvilinear coordinates may not satisfy an equilibrium check of this type. For example, the sum for the radial direction in an axisymmetric analysis will not be zero due to the influence of the “hoop stresses”.

- In addition to sums over all the nodes a sum is computed for only the nodes output. This permits the check of equilibrium on specified series of nodes, for example nodes with boundary constraints, or the computation of the applied load on a set of nodes in which motions or restraints are specified.

- `reac` prints the following values:
  - \(t=0\): loads from mate + single loads
  - after tang: reactions

- If the macro `rsum` is used during input, the resulting reaction force is always printed for the specified nodes.

- In case of a dynamic analysis dynamic forces are included.
The `read` macro command may be used to input the values of displacements and nodal stresses previously computed and saved using the `writ` macro command – it is primarily used for plots related to deformations or nodal stresses. It is not intended for a restart option (see `rest`) but may be used to restore displacement states of linear and non-linear elastic elements (or other elements with no data base requirements) for which reactions, stresses, etc. may then be computed.

The values of `xxxx` are used to specify the file name (4-characters only), manipulate the file, and write out displacements. The values permitted are:

<table>
<thead>
<tr>
<th><code>xxxx</code></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>wind</code></td>
<td>rewind the current file (note in a file is normally opened and positioned at the end-of-file mark).</td>
</tr>
<tr>
<td><code>clos</code></td>
<td>close the current output file.</td>
</tr>
<tr>
<td><code>disp</code></td>
<td>read a displacement state from the current file.</td>
</tr>
<tr>
<td><code>stre</code></td>
<td>read a nodal stress state from the current file.</td>
</tr>
<tr>
<td>anything else</td>
<td>will be used to name the current file. Only four characters are permitted and only one file may be opened at any time. Files may be opened and closed several times during any run to permit the use of more than one file name.</td>
</tr>
</tbody>
</table>

A `read` input is created using the `writ` macro command which has identical options for `xxxx`. 
The macro command **reme** may be used to refine a mesh of 4-node elements. The remeshing options shall be specified as:

- **reme,unif** - refine complete mesh (not regarding any error indicator), number of elements in the generated mesh will be four times the original number.

- **reme,adap** - refine mesh (using an error indicator)
  
  1. n1 - indicates the percentage of error norm (default 5.)
  2. n2 - specifies the type of error norm (1 - energy-norm (default), 2 - L2-norm, 3 - $\sigma_v$-norm )

- **reme,new** - read the new input file (switch i.. to n..)

Remarks:

1. To calculate problems efficiently, provide the input file with the following options:
2. Use the macro **opti**, if the standard solver is used.
3. Use the macro **nopr**, to avoid large output files, due to the fact that nodal values are written for each node separately!
4. Only meshes with 4-node elements can be generated.
5. To use the option **adap**, your element must provide the desired error indicator (isw=9).
6. The new input file will be written into a new file (ifile → nfile).
7. Nodal markings are necessary for the restart, they will be written on the ‘nfile.mrm’.
8. The ‘old’ and ‘new’ mesh can be viewed in plot with **mesh** and **rmsh**.
9. Boundary and load conditions have to be set with macros like **edge**, **eloa**, .... where data are generated.
A macro-sequence for using the adaptivity option is given by the following macros (e.g., as procedure ada1.pcd):

1. start program
   - `tang,,1`: calculate a solution
   - `reme,unif`: uniform mesh refinement
2. for each adaptive step
   - `reme,new`: read the new input file
3. then (e.g., as procedure ada2.pcd):
   - `tang,,1`: calculate a solution
   - `stre,node`: calculate nodal stresses
   - `reme,adap,n1,n2`: adaptive mesh refinement
A restart may be made using the terminal results of a previous analysis (which are retained on the restart read file specified at the start of each analysis). After entering the macro program the restart may be specified. If the previously computed problem was “dynamic”, it is necessary to specify the trans macro command prior to issuing a rest command in order to restore the terminal velocity and acceleration state. If the previous problem was static and the new analysis is to be continued as a dynamic calculation, the rest is issued before the trans macro command (since the previous analysis did not write a velocity or acceleration state to the restart file. Also specify arcl first when the problem which was analysed previously used the arc-length method.

If the save command has been used during the analysis a restart at the solution state saved on the standard restart file or on files with the extension ‘fileext’ is possible by specifying rest,<fileext>. The extension is restricted to 4 characters. This command can be used in batch and interactive mode. The File Filename.Fileext is loaded.

The use of the restart option requires considerable care to ensure that the previous results used are proper. At the termination of any analysis which computes a solution state a new file is saved on the restart write file specified at the start of the analysis. If the last analysis performed is for a different problem than the current one an error will result.

If no new solution state is computed during macro execution (e.g., only plotting is performed) no restart file is written to the specified files set – the previous restart file is retained on the original files set.

If ‘n1’ is > 0 the restart file is in ascii-mode. Thus the file can be checked explicitly or read by FEAP from a version running on another computer. This may be dangerous due to a loss of accuracy by missing digits.
The macro command **tec** can be used to save data for postprocessing with the program TECPLOT. see [http://www.tecplot.de](http://www.tecplot.de)

Alternatively the program PARAVIEW could be used, using the macro **parv**.

- **tec.init**, n1
  
  opens a file **Rname.tec** for the postprocessing data.

  With n1 the type of used element is defined.

<table>
<thead>
<tr>
<th>No.</th>
<th>Element type</th>
<th>ndf</th>
<th>ndm</th>
<th>nel</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>List of available elements</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>beam-2d</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>axishell</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>beam-3d</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>beam-3d incl. warping</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>plane stress</td>
<td>2</td>
<td>3</td>
<td>4,9</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>plane strain</td>
<td>3</td>
<td>2</td>
<td>4,9</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>plane strain</td>
<td>1</td>
<td>2</td>
<td>4,9,16</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>plane strain</td>
<td>5</td>
<td>2</td>
<td>4,9,16</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>plate</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>shell-5</td>
<td>5</td>
<td>3</td>
<td>4,9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>shell-6</td>
<td>6</td>
<td>3</td>
<td>4,9</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>shell-7 incl. warping dof of beam</td>
<td>7</td>
<td>3</td>
<td>4,9</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>brick</td>
<td>3</td>
<td>3</td>
<td>8,27</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>brick</td>
<td>4</td>
<td>3</td>
<td>8,27</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>brick</td>
<td>6</td>
<td>3</td>
<td>8,27</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>brick</td>
<td>7</td>
<td>3</td>
<td>8,27</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>tetraeder</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

- **tec.write**

  writes all current data on file **Rname.tec** e.g. in each time step. These are: nodal coordinates, elements (only in first step), nodal displacements and nodal stresses which have to be calculated by **stre,node** before!

  **DON'T FORGET** to write the initial configuration for nonlinear problems!

- **tec.eigv.n1,n2**

  writes current data on file **Rname.tec** These are: nodal coordinates, elements (only in first step), **HERE** nodal values of eigenvector ‘n1’, scaled by factor ‘n2’ (def.=1) and nodal stresses which have to be calculated by **stre,node** before!
• `tec,close`
  close the file `Rname.tec`. 
A non-linear calculation could be written completely to TECPLOT with

- `tec.init.n1`
- `stre.node`
- `tec.write` (initial configuration)
- `prop.,1`
- `dt.,1`
- `loop.,n-time-steps`
  - `time`
  - `loop.,m-iteration steps`
  - `tang.,1`
  - `next`
- `stre.node`
- `tec.write`
- `next`
- `tec.close`
RINP

- \texttt{rinp} starts FEAP reading the actual input file again.
- \texttt{rinp,new} starts FEAP reading a new input file.
- \texttt{rinp,ini} starts FEAP again without reading any input file. Only all used arrays like displacements etc. are set to zero.
SAVE

- The `save` macro command may be used to save the current solution state and history data for use as a restart file. At the current state of implementation the `save` command saves also not converged states. Thus it cannot be used without caution in iteration loops. The problem may then be initiated from the saved state by the `rest` command and continued.

- During an analysis more than one state may be saved. In the solution of complicated non-linear problems where difficulties are expected in achieving convergence (e.g., a solution step may produce an overflow which terminates execution, in conjunction with the arc-length method or in elastoplastic calculations when the maximum load increment or step-length is not known) a restart state may be saved on the disk for each converged state.

This can be done by using a file extension within the saving procedure. This `fileext` is optional and may be any 1–4 characters and is appended to the name of the current restart save file which was named when the problem was started if specified. In interactive mode should a name be given which already exists the user receives a prompt for an alternate name or given the opportunity to write over the old file.

- If ’n1’ is > 0 the save file is in ascii–mode. Thus the file can be checked explicitly or read by FEAP on another computer. This may be dangerous due to a loss of accuracy by missing digits.

- If for example restart files for all load steps are necessary then it is useful to mark the restart files with numbers. For this purpose the macro `save,init` set the first number (called ‘icount’) to ’n1’. Next each command `save,next` updates ‘icount’ and writes the next restart file. Thus a sequence

  Filename.0001, Filename.0002, Filename.0003,...

is stored.
SHOW

\texttt{show.xxxx,n1,n2,n3}

This command shows different values of the system and gives information on the currently chosen parameters on the macro level.

The permissible values and actions for \texttt{xxxx} are:

\begin{itemize}
  \item \texttt{xxxx = coor}
    Coordinates are shown for node \texttt{n1} to node \texttt{n2} with increments \texttt{n3}.
  \item \texttt{xxxx = boun}
    Boundary conditions are shown for node \texttt{n1} to node \texttt{n2} with increments \texttt{n3}.
  \item \texttt{xxxx = forc}
    Forces are shown for node \texttt{n1} to node \texttt{n2} with increments \texttt{n3}.
  \item \texttt{xxxx = elem}
    Associated nodes are shown for element \texttt{n1} to element \texttt{n2} with increments \texttt{n3}.
  \item \texttt{xxxx = node}
    Coordinates, forces and b.c. are shown for node \texttt{n1}.
  \item \texttt{xxxx = memo}
    Length and position of the currently used arrays in common M
    Length of all allocated arrays
    The currently chosen parameters on the macro level are shown.
\end{itemize}
Calculate on a RVE:

\[
S = \frac{1}{V}(F_b - L^T K^{-1} F_a)
\]
\[
C = \frac{1}{V}(M - L^T K^{-1} L)
\]

Associated 'Load' vector is defined with macro `epsq`. 
SMOO

\textbf{smoo},xxxx,n1,n2,n3

The macro command \textit{smoo} optimizes geometrically an given mesh.
The permissible values and actions for \textit{xxxx} are:

\textbf{xxxx=oesp}

simple mesh smoothing without consideration of edge- or boundary-nodes

\textbf{xxxx=wesp}

simple mesh smoothing with consideration of edge- or boundary-nodes

All boundaries for plane surfaces must be defined by curves. For spatial meshes the surface has to be
prescribed by curves. (in macro \texttt{curv} – \textit{n3} has to set to 15, 16, 17 or 18)

The parameters are:

- \textbf{n1} number of the node from which the mesh smoothing should be executed (default=1)
- \textbf{n2} number of iterations of mesh smoothing (useful 1, 2 or 3) (default=1)
- \textbf{n3} weight factor for equality of sidelengths (first two numbers;w1)
  and 90-degree-angles (third and fourth number;w2)
  (for instance w1w2=0210 denotes w1=2 and w2=10)(default=0101)
The macro command splo is used to print nodal displacements or nodal stresses along a user-defined line.

With splo,set coordinates of starting point and end point of line are set.
splo reset all values to zero.
With disp,line nodal displacements are printed.
With stre,line nodal stresses are printed.
Furthermore coordinates of starting and end point of line are set for the plot-macros dplo, eplo, rplo, splo,
SOLV

```
solv
solv,<line,v1>
```

- The macro command **solv** is used to specify when the equations generated by a **tang**, **utan** and/or **form** are to be solved. In FEAP, a direct solution of the equations is performed using a profile storage with a variable band (active column) method of solution.

- In the solution of some nonlinear problems it is possible to obtain convergence for a wider range of loading and time step size using a “line search”. The line search may be requested by placing **line** in the second field of the ’solv’ macro command. The parameter ’v1’ is the required energy reduction to preclude a line search being performed (if the current energy is larger than ’v1’ times the minimum energy in the step so far, a line search is performed). If not specified ’v1’ defaults to 0.8 (recommended values are between 0.6 and 0.9). Line search should never be used in a linear problem since extra evaluations of the residual are required during the line search.

- Note that a ’complete’ calculation can be done by **tang,,1** which is equal to **tang, form, solv**.
STRE

\textbf{stre}, <n1,n2,v3>
\textbf{stre}, all
\textbf{stre}, line
\textbf{stre}, <ints,n1,n2,n3>
\textbf{stre}, <lay ,n1,n2,v3>
\textbf{stre}, <node,n1,n2,v3>
\textbf{stre}, <pris,n1,n2,m3>
\textbf{stre}, <dmag,n1,n2,v3>

The \textbf{stre} macro command is used to output stress results in elements 'n1' to 'n2' at increments of 'v3' (defaults 'n1' = 1, 'n2' = 'n1', 'v3' = 1), or at nodes using 'projected' values. Thus, two options exist for reporting stress values. These are:

1. Stresses may be reported at selected points within each element. The specific values reported are described in each element type. In general elements report values at gauss points. The values at all points are reported when the command \textbf{stre}, all is used.

2. For 2/3-dimensional elements results may be reported at nodes using the \textbf{stre}, node option. A projection method using stresses at points in each element is used to compute nodal values. In general, nodal values are not always as accurate as stresses within elements. This is especially true for reported 'yield' stresses where values in excess of the limit value result in the projection method employed. For a mesh producing accurate results inside elements this degradation should not be significant.

3. Principal stress 'm3' may be reported at nodes 'n1' to 'n2' (increments of 1) using the \textbf{stre}, pris option. Details on principal stresses, especially on the use of 'm3' may be found within macro \textbf{pris}.

For specific values reported in each element see the description of \textbf{Elements in Student Version}.

With the option \textbf{stre}, line stresses are printed at nodes on a line defined by \textbf{splo} or \textbf{dplo}.

With the option \textbf{stre}, ints the distribution of the stresses at the center of element \textit{n1} over the thickness in a shell type element is printed. For the output \textit{n2} intervals (default \textit{n2}=10) are used for each layer.

The value of 'v3' can be used also for layered elements. If 'v3' is < 0 than abs('v3') is the layer number and the position for which the stresses should be reported. E.g. 'v3=-3.2' prints stresses in layer 3 at position 2. For the definition of layer and position see the associated element manual or formulation. In this case the incrementation between 'n1'and 'n2' by 'v3' does not work (set to 1). Output is possible at Gauss-points (\textbf{stre},lay) or at nodes (\textbf{stre},node).

Stresses at damaged integration points may be reported within each element. The specific values reported are described in each element type. Here, the command \textbf{stre},dmag prints stresses in this points.
The `subs` macro command requests the solution of \( n_1 \) eigenpairs of a problem about the current state.

Typically 2\( \cdot n_1 \) eigenvalues and eigenvectors are calculated. Only the first \( n_1 \) results could be used (print and plot), see the output of error norms.

An additional number of \( n_2 \) vectors are used to expand the subspace and improve convergence (\( n_1 \) plus \( n_2 \) is set to the minimum of the input values or \( n_1 \) plus 8 or 2 times \( n_1 \) or the maximum number of eigenvalues in the problem).

All eigenvalues are computed until two subsequent iterations produce values which are accurate to the current defined tolerance. The iteration behaviour can be controlled by the parameters 'tols' (tolerance for subspace iteration) and 'nits' (number of iterations for subspace computation) by \( n_3 \) as follows

\[
\begin{array}{ccc}
\text{\( n_3 \)} & \text{\( \text{\texttt{tols}} \)} & \text{\( \text{\texttt{nits}} \)} \\
\leq 0 & \text{1.d-12} & 25 \\
> 0 & \text{\( \text{\texttt{tol}} \)} & \text{\( \text{\texttt{tol}} \)} \\
\end{array}
\]

Thus, the default values are given in the first row of this table. The default value for \texttt{tol} is 1.d-16, see the \texttt{tol} macro.

If the solution has not converged it is possible to compute further iterations (in the interactive version).

The `subs` macro must be preceded by the specification of the tangent stiffness array using a `tang` command, and a second matrix. This can be either the mass matrix (a lumped mass by \texttt{linas} or a consistent mass by \texttt{cmas}) or a 'buckling' matrix (an identity matrix by \texttt{idem} or a geometrical matrix by \texttt{geom}). Thus one of the following eigenvalue problems can be solved:

\[
\begin{align*}
\left[K_T - \omega^2 M\right] \varphi & = 0 \\
\left[K_T - \omega I\right] \varphi & = 0 \\
\left[K_L + \Lambda K_{NL}\right] \varphi & = 0 \\
\left[K_L + \Lambda (K_U + K_G)\right] \varphi & = 0
\end{align*}
\]

Note that the smallest \( n_1 \) eigenvalues and eigenvectors are computed with reference to the current “shift” specified on the `tang` command.

If \( n_1 \) is larger than the number of non-zero mass diagonals it is truncated to the actual number that exist. Whenever \( n_1 \) is close to the number of nonzero mass diagonals one should compute the entire set since convergence will be attained in one iteration (this applies primarily to small problems).
• Use of the `prin` option in second place produces an output of all subspace matrices in addition to the estimates on the reciprocals of the shifted eigenvalues. For large problems considerable output results from a use of this option, and thus it is recommended for small problems only.

• Use of the `init` option initialize the COR–method, see macro command `eigk`.

• When a large number of eigenvalues should be calculated the subspace iteration may be very time consuming. Then the `lanczos` iteration is advantageous.

• In case of large systems often iterative solvers are used. Then the `subspace` iteration is very time consuming whereas the `lanczos` iteration can be used without any problems.

• Eigenvectors are scaled that the maximum entry is one.
SUMM

summ, file, fac
summ, zero

The `summ` macro command is used to summarize results from different calculations. The macro works correct, if the results are calculated on the same mesh (only nodes and elements) and if the displacement and stress fields have the same variables. Results in previous calculations have to be saved with the macro `writ`. The values are stored in the actual displacement and stress fields! Thus `FEAP` can be used under `summ` only as a postprocessor.

Save data:  
```
  writ,fil1
  writ,disp
  writ,clos
  ....
```

Summarize data:  
```
  summ,fil1,fac1
  summ,fil2,fac2
  ....
```

Here values of different calculations can be added by different factors fac1,fac2,...
Thus it holds for example:
```
v = fac1 * v(fil1) + fac2 * v(fil2) + ...
```

Then all `print`– and `plot`–options of `FEAP` are available.

With `summ,zero` the displacement and stress fields are cleared.
TANG

\texttt{tang,,<n1,v2>}
\texttt{tang.line,,<n1,v2,v3>}

The \texttt{tang} macro computes the tangent stiffness matrix about the current value of the solution vector.

- For linear applications the current stiffness matrix is just the normal 'stiffness' matrix.
- If the value of 'n1' is greater than zero a force vector for the current residual is also computed (this is identical to the \texttt{form} macro computation) — thus leading to greater efficiency when both the tangent stiffness and a residual force vector are needed.
- If the value of 'v2' is non–zero a “shift” is applied to the stiffness matrix in which the mass matrix (which must already be formed using a \texttt{cmas} or \texttt{lmas} macro) is multiplied by 'v2' and subtracted from the stiffness matrix. This option may be used with the \texttt{subs} pace algorithm to compute the closest eigenvalues to the shift, 'v2'. Alternatively, the shift may be used to represent a forced vibration solution in which all loads are assumed to be harmonic at a value of the square–root of 'v2' (rad/time–unit).
- After the tangent matrix is computed, a triangular decomposition is available for subsequent solutions using \texttt{form} and \texttt{solv}, \texttt{bfgs}, etc. This decomposition is not computed when the value 'n1' is negative.
- Note that a 'complete' calculation can be done by \texttt{tang,,1} which is equal to \texttt{tang}, \texttt{form}, \texttt{solv}.
- In the solution of non–linear problems, using a full or modified Newton method, convergence from any starting point is not guaranteed. Two options exist within available macro commands to improve chances for convergence.

\# One is to use a line search to prevent solutions from diverging rapidly. Specification of the command \texttt{tang.line} plus options invokes the line search option (it may also be used in conjunction with 'solv, line' in modified Newton schemes). The parameter 'v3' may be chosen between 0.5 and 0.8 and will generally produce good performance.

\# The second option to improve convergence of non–linear problems is to reduce the size of the load step increments. The macro command \texttt{back,,dt} may be used to “back–up” to the beginning of the last time step (all data in the solution vectors is reset and the history data base for inelastic elements is restored to the initial state when the current time is started). Repeated use of the 'back' command may NOT be used. Only one back–up in time is permitted. The loads may then be adjusted and a new solution with smaller step sizes started.

- In combination with \texttt{arcl} the backup has to be done by \texttt{back,,1}!
TEC

```
tec, init, n1
  tec, write
  tec, eigv, n1, n2
  tec, close
```

The macro command **tec** can be used to save data for postprocessing with the program TECPLOT. See [http://www.tecplot.de](http://www.tecplot.de)

Alternatively the program **PARAVIEW** could be used, using the macro **parv**.

- **tec, init, n1**
  
  opens a file **Rname.tec** for the postprocessing data.
  
  With n1 the type of used element is defined.

<table>
<thead>
<tr>
<th>No.</th>
<th>Element type</th>
<th>ndf</th>
<th>ndm</th>
<th>nel</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>List of available elements</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>beam-2d</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>axishell</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>beam-3d</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>beam-3d (incl. warping)</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>plane stress</td>
<td>2</td>
<td>2,3</td>
<td>3,4,9</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>plane strain</td>
<td>3</td>
<td>2</td>
<td>4,9</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>plane strain (incl. warping)</td>
<td>5</td>
<td>2</td>
<td>4,9,16</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>plane strain (incl. warping)</td>
<td>5</td>
<td>2</td>
<td>4,9,16</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>plate</td>
<td>3</td>
<td>2,3</td>
<td>3,4,9</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>shell-5</td>
<td>5</td>
<td>3</td>
<td>4,9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>shell-6</td>
<td>6</td>
<td>3</td>
<td>4,9</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>shell-7</td>
<td>7</td>
<td>3</td>
<td>4,9</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>brick</td>
<td>3</td>
<td>3</td>
<td>8,27</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>brick</td>
<td>4</td>
<td>3</td>
<td>8,27</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>brick</td>
<td>6</td>
<td>3</td>
<td>8,27</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>brick</td>
<td>7</td>
<td>3</td>
<td>8,27</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>tetraeder</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

- **tec, write**

  writes all current data on file **Rname.tec** e.g. in each time step. These are: nodal coordinates, elements (only in first step), nodal displacements and nodal stresses which have to be calculated by **stre,node** before!

  Don’t forget to write the initial configuration for nonlinear problems!

- **tec, eigv, n1, n2**

  writes current data on file **Rname.tec**. These are: nodal coordinates, elements (only in first step), here nodal values of eigenvector 'n1', scaled by factor 'n2' (def.=1) and nodal stresses which have to be calculated by **stre,node** before!

- **tec, close**

  close the file **Rname.tec**.
A non-linear calculation could be written completely to TECPLOT with

```plaintext
  tec,init,n1
  stre,node
  tec,write (initial configuration)
  prop,,1
dt,,1
loop,,n-time-steps
  time
  loop,,m-iteration steps
    tang,,1
  next
  stre,node
tec,write
next
tec,close
```
**TIME**

`time, <tmax>`
`time, start`

- The use of the `time` macro command will increment the current time by the current time increment $\Delta t$ prescribed by $dt$. In addition, a new value of the proportional loading will be computed if necessary. The value of the current time and proportional loading are reported in the output (or to the screen).

- The `time` macro also will perform the first update for the Newmark–beta time integration of the equations of motion, as well as, update the stress database for any elements with non-linear constitutive equations which require variables other than the displacement state to compute a solution. Accordingly, it is imperative to include a time macro for these classes of problems.

- As an option, it is possible to specify the maximum time that integration is to be performed. Accordingly, when a variable time step is employed the 'tmax' value may be used as a convenient stop marker. This also is essential if an automatic time stepping algorithm is implemented.

- As a second option, it is possible to set a start increment with `time, start` as an extrapolation from the last increment. This may lead to less iterations within a nonlinear solution process.

$$u_{n+1}^0 = u_n + \frac{\Delta t}{\Delta t_0} \cdot (u_n - u_{n-1})$$

Here $\Delta t$ is the current and $\Delta t_0$ the last time increment. Time increments can be prescribed with $dt$.

`time, start` is tested for different algorithms

```
# implicit dynamic algorithms: ok
# explicit dynamic algorithms: not allowed
# arcl: should not be used at present
# modified Newton-Raphson: ok
# different time steps $\Delta t_i$: ok
```

**Reference**

Details can be found in Hartmann (2009) [6].
The `tol` macro command is used to specify the solution tolerance values to be used at various stages in the analysis by the value `v1`.

Uses include:

- Convergence of non-linear problems in terms of the norm of energy in the current iterate (the inner dot product of the displacement increment and the solution residual vectors).
- Convergence of the subspace eigenpair solution which is measured in terms of the change in subsequent eigenvalues computed.

The default value of `tol` is

- 1.0d-16 for norm of energy
- 1.0d-12 for eigenvalue calculation.
TPLO

tplo,<xxxx>,n1,n2,n3

The \texttt{tplo} macro is used to plot load deflection curves within FEAP.
The following values will be provided for each load step \( n \) at node \( i \) dof \( k \):
\[ \lambda^n, t^n, v^n_{ik}, \dot{v}^n_{ik}, \ddot{v}^n_{ik}, R^n_{ik}, \sigma^n_{ik}, \text{val1}^n_{ik}, \text{val2}^n_{ik}, \]
Typically values are plotted with respect to displacement \( v_{ik} \) or time \( t \).

Initialisation

<table>
<thead>
<tr>
<th>xxxx</th>
<th>def</th>
<th>define up to 10 nodes for plotting. Do not use this command after restart!!</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxx</td>
<td>init</td>
<td>initialize macro and define node and dof</td>
</tr>
<tr>
<td>xxxx</td>
<td>mark</td>
<td>put marks on load deflection curve</td>
</tr>
<tr>
<td>xxxx</td>
<td>set</td>
<td>define active node and values</td>
</tr>
</tbody>
</table>

- \texttt{tplo.init,n1,n2,n3} initializes that from node ‘\( n1 \)’ the displacement ‘\( n2 \)’ will be plotted. The load factor and the nodal reaction force is multiplied by \( n3 \) (default = 1). This is helpful in case of symmetry conditions (e.g. double symmetry \( \rightarrow \) \( n3 = 4 \)). If ‘\( n2 \)’ is negative a plot is shown versus the negative displacement.

- Alternatively with \texttt{tplo.def} up to 10 nodes can be defined for plotting. Up to 3 nodes can be defined by \( n1,n2 \) and \( n3 \). Otherwise nodes are defined interactively (if \( n1=0 \)). Then \texttt{tplo.init,n1,n2,n3} is not used. \texttt{tplo.set,n1,n2,n3} sets the active node.

The active plotting values \( n1, n2, n3 \) can be changed by using \texttt{tplo.set,n1,n2,n3}. Further \texttt{tplo} macros act on this active node, e.g. \texttt{tplo.save}.

- \texttt{tplo.sset,n1,n2} sets that stress \( n1 \) at node \( n2 \) will be plotted with respect to values set by \texttt{tplo.set,n1,n2,n3}.

- The command \texttt{tplo.mark,n1,n2,n3} put marks on the chosen curve in increments ‘\( n3 \)’. The marks are plotted in color ‘\( n1 \)’ (default \( n1=1 \)). Consult the \texttt{colo} command for available colors. ‘\( n2 \)’ defines the type of mark (default \( n2=0 \)). The following markers are available:

<table>
<thead>
<tr>
<th>‘( n2 )’</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>marker</td>
<td>small dot</td>
<td>large dot</td>
<td>plus sign</td>
<td>triangle</td>
<td>square</td>
<td>cross</td>
<td>lozenge</td>
</tr>
</tbody>
</table>

All marks can be dropped by \texttt{tplo.mark}.

Basically the actual point of the curve is plotted in a separate color(red).
Plotting of data during Calculations

<table>
<thead>
<tr>
<th>xxxx</th>
<th>acce</th>
<th>plot acceleration vs time</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxx</td>
<td>detd</td>
<td>plot determinant vs displacement 2)</td>
</tr>
<tr>
<td>xxxx</td>
<td>dett</td>
<td>plot determinant vs time 2)</td>
</tr>
<tr>
<td>xxxx</td>
<td>disp</td>
<td>plot displacement vs time</td>
</tr>
<tr>
<td>xxxx</td>
<td>load</td>
<td>plot load vs displacement</td>
</tr>
<tr>
<td>xxxx</td>
<td>loat</td>
<td>plot load vs time</td>
</tr>
<tr>
<td>xxxx</td>
<td>phas</td>
<td>plot velocity vs displacement (phase diagram)</td>
</tr>
<tr>
<td>xxxx</td>
<td>reac</td>
<td>plot reaction vs displacement 1,5)</td>
</tr>
<tr>
<td>xxxx</td>
<td>reat</td>
<td>plot reaction vs time 1,5)</td>
</tr>
<tr>
<td>xxxx</td>
<td>strd</td>
<td>plot stress i at node k vs displacement 3)</td>
</tr>
<tr>
<td>xxxx</td>
<td>strt</td>
<td>plot stress i at node k vs time 3)</td>
</tr>
<tr>
<td>xxxx</td>
<td>velo</td>
<td>plot velocity vs time</td>
</tr>
<tr>
<td>xxxx</td>
<td>tdis</td>
<td>plot time vs displacement</td>
</tr>
<tr>
<td>xxxx</td>
<td>us1d</td>
<td>plot valuse1 vs displacement 4)</td>
</tr>
<tr>
<td>xxxx</td>
<td>us1t</td>
<td>plot valuse1 vs time 4)</td>
</tr>
<tr>
<td>xxxx</td>
<td>us2d</td>
<td>plot value2 vs displacement 4)</td>
</tr>
<tr>
<td>xxxx</td>
<td>us2t</td>
<td>plot value2 vs time 4)</td>
</tr>
</tbody>
</table>

Provision of data during Calculations

1. The nodal reactions have to be calculated within each time step via reac.
2. The determinant is calculated only within arc–length method, extended system, or using SM-solver or if calculated explicitly by the macro detk.
3. stress k at node i has to be defined with the macro tplo,sset. Furthermore stress has to be calculated via stre,node for each time step.
4. Values have to be provided by user. More informations are given in the chapter Adding elements.
5. With tplo,react the reaction force is plotted for the specified dof. If via the macro rsum a sum of forces is defined, then this sum is plotted. Same holds for tplo,reat.
Storage of data and restart options

| xxxx  | save  | save load deflection data in file rfilename.ldf |
| xxxx  | read  | read load deflection data from files rfilename.ldf and rfilename.ldd |

- The command `tplo/save` saves load deflection data until the last time step for the active node in file `rfilename.ldf`. All values (including the active time step) were saved with the command `tplo/save,,1`.
- With `tplo/save,n1` the values of the active node are saved in file `rfilename_n1.ldf` \((0 \leq n1 \leq 9)\).
- The command `tplo/read` reads load deflection data for the active node from file `rfilename.ldf`.
- With `tplo/read,n1` the values of the active node are read from file `rfilename_n1.ldf` \((0 \leq n1 \leq 9)\).
- Values for nodes defined by `tplo/def` are saved in the file `rfilename.ldd` automatically. Do not use this command after restart!! Then, these values are automatically set by `tplo/read`. If no macro `tplo/save` has been used, in a restart values are read only from `rfilename.ldd`. The active node is set to the first defined node.

Further options for displaying curves

| xxxx  | fact  | multiply curve by factors |
| xxxx  | xval  | plot between xmin and xmax on screen |
| xxxx  | yval  | plot between ymin and ymax on screen |
| xxxx  | conv  | define action in case of no convergence |
| xxxx  | incr  | set plot increment to n1 (default = 1) |

- The command `tplo/fact,n1,n2` multiply loads or reactions by the factor n2 and displacements by \(+\)\((n1 > 0)\) or \(-\)\((n1 < 0)\).
- The command `tplo/xval,n1,n2` set the plot region on the first axis to xmin = n1 and xmax = n2; n1=n2=0: full region due to curve.
- The command `tplo/yval,n1,n2` set the plot region on the second axis to ymin = n1 and ymax = n2; n1=n2=0: full region due to curve.
- With `tplo/conv,n1` action is defined in case of no convergence in last time step: n1=0: plot point, n1=1(default): do not plot point.
- `tplo/increment,n1` plots only every n1-point.

Behavior in case of a backstep, see back

- After a backstep the nodal reaction force has to be calculated again via `reac/all`. 
The use of the macro command \texttt{trans} indicates that a transient solution has to be computed. Six options are implemented. The method used depends on the specified \texttt{name} in the command.

<table>
<thead>
<tr>
<th>name</th>
<th>method</th>
<th>type</th>
<th>v1</th>
<th>v2</th>
<th>v3</th>
</tr>
</thead>
<tbody>
<tr>
<td>–</td>
<td>Newmark beta</td>
<td>implicit</td>
<td>$\beta$ (0.25)</td>
<td>$\gamma$ (0.50)</td>
<td></td>
</tr>
<tr>
<td>newm</td>
<td>Newmark beta</td>
<td>implicit</td>
<td>$\beta$ (0.25)</td>
<td>$\gamma$ (0.50)</td>
<td></td>
</tr>
<tr>
<td>back</td>
<td>Euler backward</td>
<td>implicit</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>hht</td>
<td>HHT-alpha</td>
<td>implicit</td>
<td>$\alpha$ (0.05)</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>alpha</td>
<td>Generalized-alpha</td>
<td>implicit</td>
<td>$\rho$</td>
<td>$\alpha_m$</td>
<td>$\alpha_f$</td>
</tr>
<tr>
<td>ener</td>
<td>Energy momentum conserving</td>
<td>implicit</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>bath</td>
<td>Implicit composite scheme</td>
<td>implicit</td>
<td>$\gamma$</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>expl</td>
<td>Central differences Taylor</td>
<td>explicit</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>expa</td>
<td>Central differences Verlet</td>
<td>explicit</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>off</td>
<td>stop, switch to static calc.</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

**General Remarks:**

- Algorithms NEWM, BACK, HHT, ALPHA, BATH allow and ENER has always unsymmetric stiffness matrices.
- It is possible to specify nonzero initial values.
  - displacements: using the macro command \texttt{init,disp}
  - velocities: using the macro command \texttt{init,rate}
  - accelerations: using the macro command \texttt{init,acce}
- If the initial state is not in equilibrium an initial acceleration must be obtained by using a \texttt{form,acel} macro command before initiating any transient state.
- It is also possible to compute self–equilibrating static states with non–zero displacements and then switch to a dynamic solution. Alternatively, a restart mode (\texttt{rest}) may be used to start from a previously computed non–zero state.
- A good choice for a time step within implicit methods is $\Delta t = T/20$, where $T$ is a period of the vibration of the structure. It can be calculated via: \texttt{cmas, tang, subs}, which leads to values for $\omega^2$ and $\omega$. Then it holds $T = \frac{1}{f} = \frac{2\pi}{\omega}$.
- Using the macro command \texttt{auto} leads to a variable length of the time step within a dynamic analysis. Currently this macro is implemented for the time stepping algorithms \texttt{trans}, NEWM, HHT and ALPHA.
Remarks on different methods:

- **NEWM** The Newmark–beta step–by–step integration of the equations of motion is a well known reliable method. The values of the Newmark parameters are 

  \[
  \beta \text{ - the parameter which primarily controls stability (default is 0.25).} 
  \]

  \[
  \gamma \text{ - the parameter which primarily controls numerical damping (default is 0.50). Note: } \gamma \geq 0.50. 
  \]

  With \( \beta = 0.25 \) and \( \gamma = 0.50 \) the method becomes the 'average' acceleration method. The method is absolute stable. Thus large time steps are possible.

  After a number of time steps oscillations may occur especially for accelerations in the nonlinear range. This may change the total energy dramatically and may lead to divergence. A better numerical damping can be introduced using the HHT-alpha or Generalized-alpha algorithms. The energy momentum conserving algorithm ENER conserves the total energy of the system.

  Macros for algorithm:

<table>
<thead>
<tr>
<th>initialisation</th>
<th>time steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmas or lmas</td>
<td>loop,,n</td>
</tr>
<tr>
<td>&lt;damp&gt;</td>
<td>time</td>
</tr>
<tr>
<td>dt,,xx</td>
<td>loop,,m</td>
</tr>
<tr>
<td>prop</td>
<td>tang,,l</td>
</tr>
<tr>
<td>trans</td>
<td>next(m)</td>
</tr>
<tr>
<td></td>
<td>next(n)</td>
</tr>
</tbody>
</table>

- **BACK** The Euler-backward difference solution is used for first-order ordinary differential equations such as heat transfer, etc. The method is absolute stable. Thus large time increments are possible.

  Macros for algorithm:

<table>
<thead>
<tr>
<th>initialisation</th>
<th>time steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmas or lmas</td>
<td>loop,,n</td>
</tr>
<tr>
<td>dt,,xx</td>
<td>time</td>
</tr>
<tr>
<td>prop</td>
<td>loop,,m</td>
</tr>
<tr>
<td>trans,back</td>
<td>tang,,l</td>
</tr>
<tr>
<td></td>
<td>next(m)</td>
</tr>
<tr>
<td></td>
<td>next(n)</td>
</tr>
</tbody>
</table>

- **HHT** The HILBER-HUGHES-TAYLOR (HHT) step-by-step integration of the equations of motion is a modification of the Newmark-beta method. Here the numerical dissipation is handled much better. As input the parameter \( \alpha \) has to be defined, with \( 0 \leq \alpha \leq 1/3 \). Newmark method is used for \( \alpha = 0 \).

  From \( \alpha \) follows:

  \[
  \beta = 0.25(1 + \alpha)^2 
  \]

  \[
  \gamma = 0.50 + \alpha 
  \]

  For \( \alpha = 0 \) no numerical damping occur whereas the maximum damping is introduced for \( \alpha = 1/3 \). A value \( \alpha = 0.05 \) is recommended (default).

  Macros for algorithm: see Newmark beta

- **ALPHA** The generalized-alpha step-by-step integration of the equations of motion is a modification of the Newmark-beta method. Here the numerical dissipation is handled much better. As input the spectral radius \( \rho \) has to be defined,
From $\rho$ follows with $0.5 \leq \rho \leq 1$:

$$\alpha_m = \frac{2\rho - 1}{\rho + 1}$$

$$\alpha_f = \frac{\rho}{\rho + 1}$$

For $\rho = 0$ a direct input of $\alpha_m, \alpha_f$ is possible with $0.5 \leq \alpha_m, \alpha_f \leq 0.5$.

It follows

$$\beta = 0.25(1 - \alpha_m + \alpha_f)^2$$

$$\gamma = 0.50 - \alpha_m + \alpha_f$$

Newmark method is used for $\alpha_m = \alpha_f = 0$.

Macros for algorithm: see Newmark beta

- **ENER** The energy momentum conserving step–by–step integration of the equations of motion preserves energy in the nonlinear range in contrast to the Newmark method. Here an unsymmetric stiffness matrix has to be used. The method is absolute stable. Thus large time steps are possible. To use this algorithm the residual $G$ and stiffness matrix $K_T$ have to be calculated at time $t_{n+1/2}$. Check if these features are available for the used element! The main necessary formulas are presented in section Energy-conserving algorithm: element-formulation.

<table>
<thead>
<tr>
<th>initialisation</th>
<th>time steps</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cmas or lmas</strong></td>
<td>loop,,n</td>
</tr>
<tr>
<td><strong>dt,,xx</strong></td>
<td>time</td>
</tr>
<tr>
<td><strong>prop</strong></td>
<td>loop,,m</td>
</tr>
<tr>
<td><strong>trans,ener</strong></td>
<td>utan,,l</td>
</tr>
<tr>
<td></td>
<td>next(m)</td>
</tr>
<tr>
<td></td>
<td>next(n)</td>
</tr>
</tbody>
</table>

- **BATH** The implicit composite step–by–step integration scheme is able to produce stable solutions of the equations of motion in the nonlinear range by using numerical damping. In contrast to the energy momentum conserving scheme there is no need for modifications on element level. The method is absolute stable. Thus large time steps can be used. But two Newton-Raphson-iterations have to be performed per time step.

Remarks:

# A perfect choice for linear calculations:

$$\gamma = 2 - \sqrt{2}$$

Resulting in minimal period elongation with maximum high frequency dissipation. With this choice of $\gamma K_T$ has to be calculated only one time.

# For nonlinear calculations:

$$\gamma = 0.731$$

resulting in perfect approximations of the accelerations.
Macros for algorithm:

<table>
<thead>
<tr>
<th>initialisation</th>
<th>time steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmas or lmas</td>
<td>loop,,n</td>
</tr>
<tr>
<td>dt,,xx</td>
<td>time</td>
</tr>
<tr>
<td>prop</td>
<td>loop,,m</td>
</tr>
<tr>
<td>trans,bath</td>
<td>tang,,l</td>
</tr>
<tr>
<td></td>
<td>next(m)</td>
</tr>
<tr>
<td></td>
<td>next(m)</td>
</tr>
<tr>
<td></td>
<td>next(n)</td>
</tr>
</tbody>
</table>

- **EXPL** (Version Taylor) The explicit central difference step-by-step integration of the equations of motion is only stable for small time increments. Thus a large number of time steps are needed. The method is very fast (no iteration, only right hand side terms). Only lumped mass and lumped damping matrices are allowed.

**Remarks:**

# No initial calculation is necessary.

# A simple but conservative choice of the critical time step $\Delta t_c$ is the CFL-condition (Courant, Friedrichs, Lewy):

$$\Delta t_c \leq \frac{L_{\text{min}}}{c} = \frac{2}{\omega_{\text{max}}}$$

with: $L_{\text{min}} =$ smallest element length, $c =$ speed of sound $c = \sqrt{E/\rho}$, $\rho = \gamma/g$, $E =$ elastic modulus.

# An alternate estimate for $\omega_{\text{max}}$ is provided by the Gerschgorin bound:

$$\omega_{\text{max}} \leq \max \left( \frac{1}{M_{ii}} \sum_{j=1}^{\text{ndf}} K_{ij} \right) \quad i, j = 1, \text{ndf}, \quad \Delta t_c = \frac{2}{\omega_{\text{max}}}$$

# At present no automatic choice for $\Delta t_c$ is implemented.

# Typically a large number of solution points occur. Thus, to plot not all steps in a load deflection curve use the macro tplo,incr,n1; thus only every n1-step is plotted.

Macros for algorithm:

<table>
<thead>
<tr>
<th>initialisation</th>
<th>time steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>lmas</td>
<td>loop,,n</td>
</tr>
<tr>
<td>&lt;damp,lump&gt;</td>
<td>time</td>
</tr>
<tr>
<td>dt,,xx</td>
<td>loop,,m</td>
</tr>
<tr>
<td>prop</td>
<td>tang,,l</td>
</tr>
<tr>
<td>trans,expl</td>
<td>next(m)</td>
</tr>
<tr>
<td></td>
<td>next(m)</td>
</tr>
<tr>
<td></td>
<td>next(n)</td>
</tr>
</tbody>
</table>

- **EXPA** (Version Verlet) This algorithm is a modification of EXPL and the standard implementation in commercial FE-codes. The algorithm needs initial velocities $\mathbf{v}_{n+1/2}$, which have to be calculated using the macro command init,expa!
Macros for algorithm:

<table>
<thead>
<tr>
<th>initialisation</th>
<th>time steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>lmas</td>
<td>loop,,n</td>
</tr>
<tr>
<td>&lt;damp,lump&gt;</td>
<td>time</td>
</tr>
<tr>
<td>dt,,xx</td>
<td>form,expa</td>
</tr>
<tr>
<td>prop</td>
<td>next(n)</td>
</tr>
<tr>
<td>trans,expa</td>
<td></td>
</tr>
<tr>
<td>&lt;init,rate&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;init,acce&gt;</td>
<td></td>
</tr>
<tr>
<td>init,expa</td>
<td></td>
</tr>
</tbody>
</table>

# initial velocities have to be set using the macro command init,rate!
# initial accelerations have to be set using the macro command acce,rate or form,acel!
# initial velocities $v_{n+1/2}$ have to be calculated using the macro command init,expa!
# implemented is only a version with constant time steps.

- **OFF** - stops dynamic analysis, FEAP will continue with a static analysis, e.g. TRANS,HHT starts dynamic algorithm again. Only one choice of the dynamic algorithm is allowed. Intermediate restart files are different for static or dynamic calculations!
The macro command *ueig* is used to compute *all* conjugated complex eigenvalues and eigenforms of a non-symmetric matrix generated by the *tang* command. It is not necessary to specify the identity matrix with *iden*.

The eigencomputations uses a special assembly procedure which needs at $isw = 21$ the computation of the non-symmetric matrix within the element routine.

Be careful to use this command for large problems. Since all eigenvalues and -vectors are determined the computing time might be very long.
UMAS

umas

The macro command `umas` is used to compute a non-symmetric 'mass' matrix. Each element computes a contribution to the consistent mass in the array 's' when 'isw' is 5. A non symmetric 'mass' may occur in transient solutions and can only be used together with `utan` and only for the standard solver.
**UPDH**

The `updh` macro command leads to an update of internal variables $H_2 \rightarrow H_1$ on the micro-problem within a multiscale problem.

With $'n_1'=1$ the update is initialized on **Macro**-level.

With $'n_1'=2$ the update is performed on **Micro**-level.

**Remarks:**

- The procedure for a time step on macro level must the form
  
  ```
  n1,0,0
  time set new time
  loop,,n begin iteration for solution
  tang,,1 solve problem
  next end iteration at convergence
  updh,,1 activate update on local level: done by batch,updh on micro-level
  ```

- The update of internal variables on the micro-problem is done via the procedure batch,updh, see **Material model 8**.

- The used element on macro level must have a call of the **3D-Material library** for isw=15, see chapter **Adding elements**.
The `utan` macro computes the complete tangent stiffness matrix about the current value of the solution vector. The complete tangent stiffness matrix consists of both the upper and lower non-zero profile of the matrix. Accordingly, problems which have an “unsymmetric” tangent stiffness matrix may be considered. For linear applications the current stiffness matrix is just the normal ‘stiffness’ matrix.

If the value of ‘n1’ is non-zero, a force vector for the current residual is also computed (this is identical to the `form` macro computation) — thus leading to greater efficiency when both the tangent stiffness and a residual force vector are needed.

If the value of ‘v2’ is non-zero a “shift” is applied to the stiffness matrix in which the lumped mass matrix (which must already be formed using a `lmas` macro) is multiplied by ‘v2’ and subtracted from the diagonals of the stiffness matrix. This option may be used with the `subspace` algorithm to compute the closest eigenvalues to the shift, ‘v2’. Alternatively, the shift may be used to represent a forced vibration solution in which all loads are assumed to be harmonic at a value of the square–root of ‘v2’ (rad/time–unit).

In the solution of non-linear problems using a full or modified Newton method convergence from any starting point is not guaranteed. Two options exist within available macro commands to improve chances for convergence. One is to use a line search to prevent solutions from diverging rapidly. Specification of the command `utan,line` plus options invokes the line search option (it may also be used in conjunction with `solv,line` in modified Newton schemes). Line search has been generally used only with symmetric tangents, the algorithm may not be robust for arbitrary unsymmetric tangents. The parameter ‘v3’ may be chosen between 0.5 and 0.8 and will generally produce good performance.

The second option to improve convergence of non-linear problems is to reduce the size of the load step increments. The macro command `back` may be used to “back-up” to the beginning of the last time step (all data in the solution vectors is reset and the history data base for inelastic elements is restored to the initial state when the current time is started). This is the only method which is quite general when unsymmetric tangents are used. Repeated use of the `back` command may NOT be used. Only one back-up in time is permitted. The loads may then be adjusted and a new solution with smaller step sizes started.

After the tangent matrix is computed, a triangular decomposition is performed for subsequent solutions using `form` and `solv`, etc. Note that the `bfgs` algorithm is available only for symmetric tangents and will not work in conjunction with `utan`. 

---

**UTAN**

`utan,,<n1,v2>`
`utan,line,,<n1,v2,v3>`
VELO

\texttt{velo,,<n1,n2,n3>}
\texttt{velo,all}

The macro command \texttt{velo} may be used to print the current values of the “velocity” vector as follows:

1. Using the macro command:
   \texttt{velo,,n1,n2,n3}
   prints out the current velocity vector for nodes ‘n1’ to ‘n2’ at increments of ‘n3’ (default = 1). If ‘n2’ is not specified only the value of node ‘n1’ is output. If both ‘n1’ and ‘n2’ are not specified only the first nodal velocity is reported.

2. If the macro command is specified as:
   \texttt{velo,all}
   prints all nodal quantities.

In order to output a solution vector it is first necessary to specify macro commands to compute the desired values, i.e., a dynamic analysis.
WRIT

writ,xxxx

The `writ` macro command may be used to save the current values of displacements and nodal stresses for subsequent use. This option is particularly useful for saving states which are to be plotted later. It is not intended as a restart option (see `rest` for restarting a previously saved problem state).

The values of `xxxx` are used to specify the file name (4-characters only), manipulate the file, and write out states. The values permitted are:

| xxxx = wind | rewind the current file (note in a file is normally opened and positioned at the end-of-file mark). |
| xxxx = clos | close the current output file. |
| xxxx = disp | write the current displacement state onto the current file. |
| xxxx = stre | write the current nodal stress state onto the current file. |
| xxxx = anything else | will be used to name the current file. Only four characters are permitted and only one file may be opened at any time. Files may be opened and closed several times during any run to permit the use of more than one file name. |

A `writ` output is reinput using the `read` macro command which has identical options for `xxxx`.

`writ` outputs can be summarized with the `summ` macro command.
YANG

**yang**

The macro command **yang** may be used to print the angles of the deflected planes after a yield-line calculation step performed by **ytab**.
The macro command *yevo* performs an evolution step during a yield-line optimization process.
The macro command `ygra` detects the gradient of the search directions during a yield-line optimization process based on the gradient method. The step length is calculated automatically, but it can be defined by using the `dt` command. To reactivate its automatical definition set ’dt’ to zero. After a gradient step every new mesh has to be loaded by the `ymsh` command.

For further details concerning the yield-line optimization process using linear programming, see the `ytab` macro command.
The macro command `ymsh` is used to reload the yield-line mesh which was modified during the yield-line optimization process based on the gradient method.

For further details concerning the yield-line optimization process using linear programming, see the `ytab` macro command.
YTAB

ytab

The ytab macro computes the simplex optimization tableau during a yield-line calculation based on linear programming. The minimal load factor is solved and printed for the current optimization step.

- Performing the gradient method the ygra command has to be used afterwards. Furthermore the modified mesh has to be reloaded by the ymsh command before using ytab again.
- Performing the direct search method the ytry command is used. It is not necessary to reload the mesh.

For both methods the step length may be changed by using the dt command.

Note: the selected optimization method should not be changed during the calculation process.
The macro command \texttt{ytry} performs a trial step during a yield-line optimization calculation based on the direct search method. At the beginning of a calculation process the step length is defined by $\frac{1}{10}$ of the smaller plate dimension along the coordinates. The step length may be modified by using the \texttt{dt} command.

For further details concerning the yield-line optimization process using linear programming, see the \texttt{ytab} macro command.
Chapter 4

Plot Commands

4.1 Available Macros

In FEAP, plots may be made for several quantities of interest in 2/3-dimensional problems. A plot quantity must be specified to initiate graphics outputs. After entering graphics mode a prompt will be displayed. At this time, quantity and the 'n1', 'n2', and 'n3' values may be specified. Alternatively, a plot,quantity,n1,n2,n3 command may be issued while in macro execution mode (this is the only option for batch executions).

The values n1,n2,n3 can be predefined parameters. This can be done e.g. on input-level via para or on macro-level via para. Furthermore calculations for n1,n2,n3 are possible, each up to 15 characters. For details see the description how to define parameter.

The following may be used for quantity:

<table>
<thead>
<tr>
<th>aacc</th>
<th>acce</th>
<th>adis</th>
<th>aeig</th>
<th>aeve</th>
<th>angl</th>
<th>avel</th>
<th>axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>back</td>
<td>base</td>
<td>bord</td>
<td>boun</td>
<td>cart</td>
<td>ceig</td>
<td>cent</td>
<td>clip</td>
</tr>
<tr>
<td>colo</td>
<td>copy</td>
<td>defm</td>
<td>defo</td>
<td>dimp</td>
<td>disp</td>
<td>dmag</td>
<td>dplo</td>
</tr>
<tr>
<td>draw</td>
<td>eigi</td>
<td>eigv</td>
<td>elem</td>
<td>end</td>
<td>eplo</td>
<td>erro</td>
<td>evan</td>
</tr>
<tr>
<td>evex</td>
<td>fact</td>
<td>flux</td>
<td>forc</td>
<td>fram</td>
<td>hide</td>
<td>hids</td>
<td>hmsh</td>
</tr>
<tr>
<td>init</td>
<td>isec</td>
<td>ints</td>
<td>isom</td>
<td>jint</td>
<td>line</td>
<td>link</td>
<td>load</td>
</tr>
<tr>
<td>logo</td>
<td>magn</td>
<td>man</td>
<td>mate</td>
<td>matn</td>
<td>maxi</td>
<td>mesh</td>
<td>mono</td>
</tr>
<tr>
<td>move</td>
<td>movi</td>
<td>ndii</td>
<td>node</td>
<td>out1</td>
<td>pele</td>
<td>pers</td>
<td>pdis</td>
</tr>
<tr>
<td>plof</td>
<td>pnod</td>
<td>pola</td>
<td>prin</td>
<td>pris</td>
<td>prof</td>
<td>quit</td>
<td>reac</td>
</tr>
<tr>
<td>rmsh</td>
<td>resi</td>
<td>rot0</td>
<td>rot1</td>
<td>rot2</td>
<td>rot3</td>
<td>rotm</td>
<td>rplo</td>
</tr>
<tr>
<td>rsum</td>
<td>scal</td>
<td>sect</td>
<td>show</td>
<td>size</td>
<td>slee</td>
<td>splo</td>
<td>stre</td>
</tr>
<tr>
<td>str1</td>
<td>symm</td>
<td>text</td>
<td>tie</td>
<td>tplo</td>
<td>traj</td>
<td>ueig</td>
<td>velo</td>
</tr>
<tr>
<td>wipe</td>
<td>xsca</td>
<td>zoom</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A short overview on commands is given in section 4.2, see macro Index in FEAP, whereas possible actions can be found in section 4.3, see macro Action in FEAP.

The action to be taken by each command is described in the following sections of the PLOT Manual.
### 4.2 Overview on commands

<table>
<thead>
<tr>
<th>Macro</th>
<th>Aufgabe</th>
<th>Macro</th>
<th>Aufgabe</th>
</tr>
</thead>
<tbody>
<tr>
<td>aacc</td>
<td>Beschleunigungen (Pfeile)</td>
<td>line</td>
<td>Liniendarstellung</td>
</tr>
<tr>
<td>acce</td>
<td>Beschleunigungen</td>
<td>link</td>
<td>Knoten: gelinkt</td>
</tr>
<tr>
<td>adis</td>
<td>Verschiebungen (Pfeile)</td>
<td>load</td>
<td>Kräfte</td>
</tr>
<tr>
<td>aeig</td>
<td>Eigenvektoren (Pfeile)</td>
<td>logo</td>
<td>Logo - Farbe</td>
</tr>
<tr>
<td>aeve</td>
<td>Eigenvektor Ext.Syst. (Pfeile)</td>
<td>magn</td>
<td>Vergrößerung des Plotbildes</td>
</tr>
<tr>
<td>angl</td>
<td>Knoten: mit Winkeln</td>
<td>man</td>
<td>Manual</td>
</tr>
<tr>
<td>avel</td>
<td>Geschwindigkeiten (Pfeile)</td>
<td>mate</td>
<td>Materialdarstellung</td>
</tr>
<tr>
<td>axis</td>
<td>Koordinatensystem</td>
<td>matn</td>
<td>Materialnummer</td>
</tr>
<tr>
<td>back</td>
<td>Hintergrund</td>
<td>maxi</td>
<td>Maximalwerte Versch./Spannungen etc.</td>
</tr>
<tr>
<td>base</td>
<td>Dreibein</td>
<td>mesh</td>
<td>System</td>
</tr>
<tr>
<td>borp</td>
<td>Rahmen</td>
<td>mono</td>
<td>Farbe: Farbe/schwarz - weiss</td>
</tr>
<tr>
<td>boun</td>
<td>Randbedingungen</td>
<td>move</td>
<td>Position Bild auf Schirm</td>
</tr>
<tr>
<td>cart</td>
<td>Cartesische Darstellung</td>
<td>movi</td>
<td></td>
</tr>
<tr>
<td>ceig</td>
<td>Zentriere Bild</td>
<td>ndii</td>
<td>Knoten: negative Diagonalelemente</td>
</tr>
<tr>
<td>cent</td>
<td>Ausschnitt</td>
<td>node</td>
<td>Knoten</td>
</tr>
<tr>
<td>clip</td>
<td>Farbe ändern</td>
<td>outl</td>
<td>System: Umrisse</td>
</tr>
<tr>
<td>copy</td>
<td>Bildausgabe (WINDOWS)</td>
<td>pele</td>
<td>Elementnummer per Mausklick</td>
</tr>
<tr>
<td>defm</td>
<td>System verformt</td>
<td>pers</td>
<td>Perspektivische Darstellung</td>
</tr>
<tr>
<td>defo</td>
<td>Plote auf verformtes System</td>
<td>pdis</td>
<td>Vorgegebene Verschiebungen</td>
</tr>
<tr>
<td>dimp</td>
<td>Imperfektionen+Verschiebungen</td>
<td>plof</td>
<td>Bildausgabe (CGM)</td>
</tr>
<tr>
<td>disp</td>
<td>Verschiebungen</td>
<td>pnod</td>
<td>Knotennummer per Mausklick</td>
</tr>
<tr>
<td>dmag</td>
<td>Schädigung</td>
<td>pola</td>
<td>Polarkoordinaten</td>
</tr>
<tr>
<td>dplo</td>
<td>Schnitt: Verschiebungen</td>
<td>prin</td>
<td>Bildausgabe (PS)</td>
</tr>
<tr>
<td>draw</td>
<td>Zeichne Linien</td>
<td>prof</td>
<td>Profil der Steifigkeitsmatrix</td>
</tr>
<tr>
<td>eigi</td>
<td>Eigenvector: inverse Iteration</td>
<td>rot0</td>
<td>Rotation: Zurücksetzen</td>
</tr>
<tr>
<td>eigv</td>
<td>Eigenvector: Subspace Iteration</td>
<td>rot1</td>
<td>Rotation: um Achse 1</td>
</tr>
<tr>
<td>elem</td>
<td>Elementnummern</td>
<td>rot2</td>
<td>Rotation: um Achse 2</td>
</tr>
<tr>
<td>end</td>
<td>Ende Plot</td>
<td>rot3</td>
<td>Rotation: um Achse 3</td>
</tr>
<tr>
<td>eplo</td>
<td>Schnitt: Eigenvektoren</td>
<td>rotm</td>
<td>Rotation: Drehung mit Maus</td>
</tr>
<tr>
<td>erro</td>
<td>Fehlerverteilung</td>
<td>rpl0</td>
<td>Schnitt: Reaktionskräfte</td>
</tr>
<tr>
<td>evan</td>
<td>Eigenvector: Animation</td>
<td>rsum</td>
<td>Reaktionskraefte bei mehreren Knoten</td>
</tr>
<tr>
<td>evex</td>
<td>Eigenvector: Extended System</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fact</td>
<td>Grössenfaktor</td>
<td>scal</td>
<td>Vergrösserungsfaktor</td>
</tr>
<tr>
<td>flux</td>
<td>Fluss</td>
<td>sect</td>
<td>Querschnittswerte</td>
</tr>
<tr>
<td>forc</td>
<td>Schnittgrössen</td>
<td>show</td>
<td>Zustandsinformationen</td>
</tr>
<tr>
<td>fram</td>
<td>Teilrahmen</td>
<td>size</td>
<td>Schriftgroesse</td>
</tr>
<tr>
<td>hide</td>
<td>Hidden Line 3D</td>
<td>slee</td>
<td>Pause</td>
</tr>
<tr>
<td>hids</td>
<td>Hidden Line Schale</td>
<td>splo</td>
<td>Schnitt: Spannungen</td>
</tr>
<tr>
<td>hmsl</td>
<td>System gefüllt</td>
<td>stre</td>
<td>Spannungen</td>
</tr>
<tr>
<td>init</td>
<td>Anfangswerte setzen</td>
<td>str1</td>
<td>Spannungen - elementweise konstant</td>
</tr>
<tr>
<td>ints</td>
<td>Spannungen über Dicke</td>
<td>symm</td>
<td>Symmetrie</td>
</tr>
<tr>
<td>isec</td>
<td>Verschneidungen</td>
<td>text</td>
<td>Text</td>
</tr>
<tr>
<td>isom</td>
<td>Isometrische Darstellung</td>
<td>tie</td>
<td>Knoten: zusammenfallend</td>
</tr>
<tr>
<td>jint</td>
<td>Materielle Kräfte</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macro</td>
<td>Aufgabe</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tplo</td>
<td>Kurvendiagramme</td>
<td></td>
<td></td>
</tr>
<tr>
<td>traj</td>
<td>Trajektorien der Spannungen</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ueig</td>
<td>Eigenvektor: unsymmetrisch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>velo</td>
<td>Geschwindigkeiten</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wipe</td>
<td>Löschte Bild</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xsca</td>
<td>Skalierung</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zoom</td>
<td>Ausschnitt</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 4.3 Overview on actions

<table>
<thead>
<tr>
<th>Aufgabe</th>
<th>Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anfangswerte setzen</td>
<td>init</td>
</tr>
<tr>
<td>Ausschnitt</td>
<td>zoom</td>
</tr>
<tr>
<td>Ausschnitt</td>
<td>clip</td>
</tr>
<tr>
<td>Beschleunigungen</td>
<td>acce</td>
</tr>
<tr>
<td>Beschleunigungen (Pfeile)</td>
<td>aacc</td>
</tr>
<tr>
<td>Bildausgabe (CGM)</td>
<td>plof</td>
</tr>
<tr>
<td>Bildausgabe (PS)</td>
<td>prin</td>
</tr>
<tr>
<td>Bildausgabe (WINDOWS)</td>
<td>copy</td>
</tr>
<tr>
<td>Cartesische Darstellung</td>
<td>cart</td>
</tr>
<tr>
<td>Dreibein</td>
<td>base</td>
</tr>
<tr>
<td>Eigenvektor: Subspace Iteration</td>
<td>eigv</td>
</tr>
<tr>
<td>Eigenvektor: (Pfeile)</td>
<td>aeig</td>
</tr>
<tr>
<td>Eigenvektor: Animation</td>
<td>evan</td>
</tr>
<tr>
<td>Eigenvektor: Extended System</td>
<td>evex</td>
</tr>
<tr>
<td>Eigenvektor: Ext.Syst. (Pfeile)</td>
<td>aeve</td>
</tr>
<tr>
<td>Eigenvektor: inverse Iteration</td>
<td>eigi</td>
</tr>
<tr>
<td>Eigenvektor: unsymmetrisch</td>
<td>ueig</td>
</tr>
<tr>
<td>Elementnummern</td>
<td>elem</td>
</tr>
<tr>
<td>Elemente: Nr. per Maus</td>
<td>pele</td>
</tr>
<tr>
<td>Ende Plot</td>
<td>end</td>
</tr>
<tr>
<td>Ende Plot</td>
<td>quit</td>
</tr>
<tr>
<td>Farbe ändern</td>
<td>colo</td>
</tr>
<tr>
<td>Farbe: Farbe/schwarz - weiss</td>
<td>mono</td>
</tr>
<tr>
<td>Fehlerverteilung</td>
<td>erro</td>
</tr>
<tr>
<td>Fluss</td>
<td>flux</td>
</tr>
<tr>
<td>Geschwindigkeiten</td>
<td>velo</td>
</tr>
<tr>
<td>Geschwindigkeiten (Pfeile)</td>
<td>avel</td>
</tr>
<tr>
<td>Grössenfaktor</td>
<td>fact</td>
</tr>
<tr>
<td>Hidden Line 3D</td>
<td>hide</td>
</tr>
<tr>
<td>Hidden Line Schale</td>
<td>hids</td>
</tr>
<tr>
<td>Hintergrund</td>
<td>back</td>
</tr>
<tr>
<td>Isometrische Darstellung</td>
<td>isom</td>
</tr>
<tr>
<td>Imperfektionen+Verschiebungen</td>
<td>dimp</td>
</tr>
<tr>
<td>Koordinatensystem</td>
<td>axis</td>
</tr>
<tr>
<td>Knoten</td>
<td>node</td>
</tr>
<tr>
<td>Knoten: gelinkt</td>
<td>link</td>
</tr>
<tr>
<td>Knoten: negative Diagonalelemente</td>
<td>ndii</td>
</tr>
<tr>
<td>Knoten: mit Winkeln</td>
<td>angl</td>
</tr>
<tr>
<td>Knoten: zusammenfallend</td>
<td>tie</td>
</tr>
<tr>
<td>Knoten: Nr. per Maus</td>
<td>pnod</td>
</tr>
<tr>
<td>Kräfte</td>
<td>load</td>
</tr>
<tr>
<td>Kurvendiagramme</td>
<td>tplo</td>
</tr>
<tr>
<td>Liniendarstellung</td>
<td>line</td>
</tr>
<tr>
<td>Lösche Bild</td>
<td>wipe</td>
</tr>
<tr>
<td>Logo - Farbe</td>
<td>logo</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Aufgabe</th>
<th>Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual</td>
<td>man</td>
</tr>
<tr>
<td>Materialdarstellung</td>
<td>mate</td>
</tr>
<tr>
<td>Materialnummer</td>
<td>matn</td>
</tr>
<tr>
<td>Materielle Kräfte</td>
<td>jint</td>
</tr>
<tr>
<td>Maximalwerte Versch./Spannungen</td>
<td>maxi</td>
</tr>
<tr>
<td>Pause</td>
<td>slee</td>
</tr>
<tr>
<td>Perspektivische Darstellung</td>
<td>pers</td>
</tr>
<tr>
<td>Plotte auf verformtes System</td>
<td>defo</td>
</tr>
<tr>
<td>Polarkoordinaten</td>
<td>pola</td>
</tr>
<tr>
<td>Position Bild auf Schirm</td>
<td>move</td>
</tr>
<tr>
<td>Profil der Steifigkeitsmatrix</td>
<td>prof</td>
</tr>
<tr>
<td>Querschnittswerte</td>
<td>sect</td>
</tr>
<tr>
<td>Rahmen</td>
<td>bord</td>
</tr>
<tr>
<td>Randbedingungen</td>
<td>boun</td>
</tr>
<tr>
<td>Reaktionskräfte</td>
<td>reac</td>
</tr>
<tr>
<td>Residuum</td>
<td>resi</td>
</tr>
<tr>
<td>Rotation: Drehung mit Maus</td>
<td>rotm</td>
</tr>
<tr>
<td>Rotation: um Achse 1</td>
<td>rot1</td>
</tr>
<tr>
<td>Rotation: um Achse 2</td>
<td>rot2</td>
</tr>
<tr>
<td>Rotation: um Achse 3</td>
<td>rot3</td>
</tr>
<tr>
<td>Rotation: Zurücksetzen</td>
<td>rot0</td>
</tr>
<tr>
<td>Reaktionskrafte: Knoten bei Summe</td>
<td>rsum</td>
</tr>
<tr>
<td>Schädigung</td>
<td>dmag</td>
</tr>
<tr>
<td>Schnitt: Eigenvektoren</td>
<td>eplo</td>
</tr>
<tr>
<td>Schnitt: Reaktionskräfte</td>
<td>rplo</td>
</tr>
<tr>
<td>Schnitt: Spannungen</td>
<td>splo</td>
</tr>
<tr>
<td>Schnitt: Verschiebungen</td>
<td>dplo</td>
</tr>
<tr>
<td>Schnittgrössen</td>
<td>forc</td>
</tr>
<tr>
<td>Schriftgrösse</td>
<td>size</td>
</tr>
<tr>
<td>Skalierung</td>
<td>xsca</td>
</tr>
<tr>
<td>Spannungen</td>
<td>stre</td>
</tr>
<tr>
<td>Spannungen - elementweise konstant</td>
<td>str1</td>
</tr>
<tr>
<td>Spannungen über Dicke</td>
<td>ints</td>
</tr>
<tr>
<td>Symmetrie</td>
<td>symm</td>
</tr>
<tr>
<td>System</td>
<td>mesh</td>
</tr>
<tr>
<td>System: gefüllt</td>
<td>hmsh</td>
</tr>
<tr>
<td>System: Umrisse</td>
<td>outl</td>
</tr>
<tr>
<td>System: verformt</td>
<td>defm</td>
</tr>
<tr>
<td>Trajektorien der Spannungen</td>
<td>tray</td>
</tr>
<tr>
<td>Teilrahmen</td>
<td>fram</td>
</tr>
<tr>
<td>Text</td>
<td>text</td>
</tr>
<tr>
<td>Aufgabe</td>
<td>Macro</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Vergrösserung des Plotbildes</td>
<td>magn</td>
</tr>
<tr>
<td>Vergrösserungsfaktor</td>
<td>scal</td>
</tr>
<tr>
<td>Verschiebungen</td>
<td>disp</td>
</tr>
<tr>
<td>Verschiebungen (Pfeile)</td>
<td>adis</td>
</tr>
<tr>
<td>Verschneidungen</td>
<td>isec</td>
</tr>
<tr>
<td>Vorgegebene Verschiebungen</td>
<td>pdis</td>
</tr>
<tr>
<td>Zeichne Linien</td>
<td>draw</td>
</tr>
<tr>
<td>Zentriere Bild</td>
<td>cent</td>
</tr>
<tr>
<td>Zustandsinformationen</td>
<td>show</td>
</tr>
<tr>
<td></td>
<td>ceig</td>
</tr>
<tr>
<td></td>
<td>movi</td>
</tr>
</tbody>
</table>
4.4 Macros in detail

AACC(E)

\texttt{aacc,<n1,n2,n3>}

Plot all nodal accelerations currently calculated. The maximum length will be automatically scaled. All other accelerations will be scaled in proportion to the maximum, accordingly some very small values may scale to be too small to be visible.

In addition the results can be scaled by the value of \textquote{n3}'.

If \textquote{n1}' is negative accelerations will be positioned relative to the deformed mesh. If \textquote{n2}' is nonzero the vector tip will appear next to the node whereas if \textquote{n2}' is zero the tail of the acceleration vectors are on the nodes.
ACCE

\texttt{acce,\langle v1,v2,v3,v4\rangle}

Plot contours for degree of freedom (i.e., acceleration) ‘n1’. For negative values of ‘n1’ the acceleration are plotted on the deformed mesh.

The macro leads to a filled plot for ‘n2’= 0. On the other hand contour lines are plotted for ‘n2’ > 0.

\textbf{Filled plots}

In interactive mode \texttt{FEAP} presents a profile of results with maximum and minimum values. Then \texttt{FEAP} asks for the number of used colors and the maximum and minimum value. Up to 14 colors are possible. The default value for the number of colors is 14. As default \texttt{FEAP} calculates the contour lines automatically between the extremal values, otherwise between the given input values.

\textbf{Contour plots}

In interactive mode \texttt{FEAP} presents a profile of results with maximum and minimum values. Then \texttt{FEAP} asks for the number of used contours and the maximum and minimum value. Up to 14 contours are possible. The default value for the number of contours is 14. As default \texttt{FEAP} calculates the contour lines automatically between the extremal values, otherwise between the given input values.

For contour line plots, a non-zero ‘n3’ value will suppress numbers near each contour line. With ‘n4’ the mesh is plotted in color ‘n4’ additionally.

Compare the handling of this macro with \texttt{stre}.

In batch mode, the requisite data described above must appear after the \texttt{end} macro command in the order that the program needs inputs.
ADIS(P)

\texttt{adis,<n1,n2,n3>}

Plot all nodal displacements currently calculated. The maximum length will be automatically scaled. All other displacements will be scaled in proportion to the maximum, accordingly some very small values may scale to be too small to be visible.

In addition the results can be scaled by the value of \texttt{'n3'}. If \texttt{'n1'} is negative displacements will be positioned relative to the deformed mesh. If \texttt{'n2'} is nonzero the vector tip will appear next to the node whereas if \texttt{'n2'} is zero the tail of the displacement vectors are on the nodes.
AEIG(V)

```
aeig,<n1,n2,n3>
```

Plot all nodal displacements of eigenvector ’n1’. The maximum length will be automatically scaled. All other displacements will be scaled in proportion to the maximum, accordingly some very small values may scale to be too small to be visible.

In addition the results can be scaled by the value of ’n3’.

If ’n1’ is negative displacements will be positioned relative to the deformed mesh. If ’n2’ is nonzero the vector tip will appear next to the node whereas if ’n2’ is zero the tail of the displacement vectors are on the nodes.
AEVE(X)

\texttt{aeve, n1, n2, n3}

Plot all nodal displacements of eigenvector from extended system. The maximum length will be automatically scaled. All other displacements will be scaled in proportion to the maximum, accordingly some very small values may scale to be too small to be visible.

In addition the results can be scaled by the value of ‘n3’.

If ‘n1’ is negative displacements will be positioned relative to the deformed mesh. If ‘n2’ is nonzero the vector tip will appear next to the node whereas if ‘n2’ is zero the tail of the displacement vectors are on the nodes.
**ANGL**

`angl,<n1,n2,n3>`

Plot locations of all nodes with angles. Only nodes within the current plot region are displayed.

In case `n1` is negative the nodes are plotted relative to the deformed mesh.

In case `n2` is greater than zero the node numbers are plotted too.

In case `n2` is less than zero only the node with number `n2` is plotted.

In case `n3` is greater than zero the local basis is plotted. The length can be scaled by `n3` (base=1).
**AVEL(O)**

`avel,<n1,n2,n3>`

Plot all nodal velocities currently calculated. The maximum length will be automatically scaled. All other velocities will be scaled in proportion to the maximum, accordingly some very small values may scale to be too small to be visible.

In addition the results can be scaled by the value of `n3`.

If `n1` is negative velocities will be positioned relative to the deformed mesh. If `n2` is nonzero the vector tip will appear next to the node whereas if `n2` is zero the tail of the displacement vectors are on the nodes.
AXIS

axis,<v1,v2,v3>

A set of axes defining the coordinate directions will be plotted with the origin of the axes placed at coordinates \(x_1=v1, x_2=v2, x_3=v3\). The \(x_1, x_2\) and \(x_3\) coordinates are in the dimensions of the problem distances.
back,<n1,n2,n3,n4>

Set the background color if ’n1’ = 1.

- WIN
  - n1 = 0 back = white (default for n1)
  - n1 = 1 back = black

- GKS
  - n1 = 0 back = black (default for n1)
  - n1 = 1 back = white (default, if n1 = 1)
  - n1 = 1 back = red(n2), green(n3), blue(n4)
BASE

\texttt{base,\langle n1,n2,n3\rangle}

Plot nodal basis at nodes. The maximum length will be automatically scaled. The results can be scaled by the value of \texttt{'n1'} (default 1). If the basis is calculated on element level vectors can be plotted from element \texttt{'n2'} to \texttt{'n3'} (default 1 to numel). Otherwise if the basis is calculated for global nodes the basis vectors can be plotted from node \texttt{'n2'} to \texttt{'n3'} (default 1 to numnp).
BORD

\texttt{bord}, n1

The command redraws the border in color ‘n1’. Consult the \texttt{colo} command for tables of available colors.

In a similar way the \texttt{FEAP}–logo is redrawn by the \texttt{logo}–command.

If ‘n1’ is 0, \texttt{bord} = \texttt{wipe}.

If ‘n1’ is negative, border and logo are not plotted!
The boundary restraints will be indicated by small triangles drawn at each node in the direction of the imposed restraint. If 'n1' is zero or positive the restraints are drawn through the undeformed position of the node, whereas, if 'n1' is negative the restraints are through the deformed position of the nodes.

The absolut value of 'n1' is used as scaling factor.

With 'n2' it is possible to plot only the boundary conditions for 'dof 2'. The default value is 0, which leads to a plot of all b.c.

The boundary conditions 1-3 are drawn in red, 4-6 and further conditions in orange.

The boundary conditions 4-6 overwrite the values 1-3.

Remarks:

1. If \texttt{angl} is specified in the mesh input data, the boundary codes appear in the sloping direction.

2. Linked dofs/nodes are also plotted when master nodes have boundary conditions! Additional use of the macro \texttt{link} plot the link-behaviour.
CART

cart

All plots will be drawn in a cartesian frame. This is the default view of plots. A plot may also be in a perspective view (see pers plot manual) or an isometric view (see isom manual entry).
N.B. Problems of centering the isometric view may still exist.
This macro is up to now not documented.
With the macro `cent`er it is possible to center the plot output on the screen. $v_1$ and $v_2$ are center coordinates $[0 < v_1,v_2 < 1]$. For $v_1 = v_2 = 0$ the position can be defined with the mouse cursor. $v_1 = v_2 = -1$ resets the center coordinates to 0.5/0.5.
Using the `clip` command a closeup view of any part of the mesh may be plotted.

Use the mouse to define two corners of the region to appear in subsequent plots. After the first point is located and marked (by pressing the left mouse button) the region will appear in a ‘rubber band’ box (still press the left mouse button) until the second position is located and marked (release the left mouse button). Any subsequent plots will be positioned to show only the ‘clipped’ part of the mesh. To return to the full view of the mesh enter `zoom` as a plot instruction.

The `clip` command may be repeated on any mesh or part as many times as an element is still appearing on the screen.
COLO

colo,n1,n2

- Change plot color to ‘n1’ value. This command works not in combination with all macros. If ‘n1’ is negative the previous plot color is set to the next value in the table. Colo,0 resets all colors to the initial values.

- Contour plots are plotted in the range color1 → color2.
  colo,-1 change the color direction in: color2 → color1.
  colo,0 resets the range to the initial direction.

- Color table for **FEAP**

<table>
<thead>
<tr>
<th>n1</th>
<th>Color</th>
<th>change for VGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>white</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>red</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>blue</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>cyan</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>green</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>orange</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>yellow</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>magenta</td>
<td>light cyan</td>
</tr>
<tr>
<td>9</td>
<td>light green</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>light cyan</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>dark red</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>dark cyan</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>med red</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>dark orange</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>white</td>
<td>dark blue</td>
</tr>
<tr>
<td>16</td>
<td>black</td>
<td></td>
</tr>
<tr>
<td>17-30</td>
<td>filled plots</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>white</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>black</td>
<td></td>
</tr>
</tbody>
</table>

The colors for filled plots are from blue to red (color) or from black to white (b & w).
COPY

```
COPY.
```

With the `copy` command the actual graphic window can be copied to the clipboard for further use within standard Windows applications. `n1=1` (default) copies the whole window whereas `n1=2` copies a part of the window defined with the mouse. Move mouse to first point and press left button. Continue to press this button while moving mouse to the 2. point. Loose button which terminates the procedure.

**Remark:** Use `back`,,1 before to have a white background.

The `copy` command acts only in the Windows-Version.
DEFM

\texttt{defm,n1,n2,n3>}

This command will plot a deformed mesh with the displacements multiplied by the ‘n1’ value (default: \(n1 = 1\)). If any part of an element in the deformed mesh leaves the plot region, it will not appear in the plot. If nodes or element numbers are added some “unconnected” values may appear on the screen if the deformed plot option is used.

The \texttt{scal} command may be used to increase the region of the plot, and thus include all elements in a plot. An alternative to the \texttt{defm} command is \texttt{mesh,-1}.

The deformed mesh can be plotted only for material ‘n2’ (default: complete mesh)

The deformed mesh is plotted in color ‘n3’. Consult the \texttt{colo} command for tables of available colors. The default value is ‘n3’ = 5 (green).
DEFO

defo,n1

defo defines if the plot results acts on the undeformed or the deformed mesh. For ‘n1’ > 0 all results are plotted on the deformed mesh. Thus in all other plot-macros ‘n1’ has not to be negative. The macro is active until ‘defo,0’ (0= default). Then all results are plotted on the undeformed mesh.
DIMP

\texttt{disp,n1,n2,n3,n4}

Plot imperfections (introduced in the input-file via \texttt{impf} and displacements. Parameter and handling are similar to the macro \texttt{disp}. 

Institut für Baustatik, KIT Cont. Mesh Macro Plot Mate3D Elem Nege Gmesh Cylt Isogeo 284
**DISP**

```
disp,n1,n2,n3,n4
```

Plot contours for degree of freedom (i.e. displacement) ’n1’. For negative values of ’n1’ the displacements are plotted on the deformed mesh.

The macro leads to a filled plot for ’n2’= 0. On the other hand contour lines are plotted for ’n2’> 0.

**Filled/Contour plots**

FEAP calculates 14 contour lines automatically between the extremal values. Values can be changed by *init.*

In interactive mode (by *init.* 1) FEAP presents a profile of results with maximum and minimum values. Input is the number of used colors and the maximum and minimum value. Up to 14 (default) colors are possible. As default FEAP calculates the contour lines automatically between the extremal values, otherwise between the given input values.

For contour line plots, a non-zero ’n3’ value will suppress numbers near each contour line. With ’n4’ the mesh is plotted in color ’n4’ additionally.

For a further description see the macro *stre.*
DMAG

dmag

Plot gauss-points where a damage has been found by a user defined criterion. This criterion is has to be defined in each element.
DPLO

dplo,n1,<n2,n3>

- The 'dplo' command may be used to construct plots of displacements along any specified line in the mesh for 2D-problems. 3D-problems can be calculated too if 'pers' and 'hide' is used. The line is than on the 3d-element faces.
- The line is scaled to 0-1 for the intersected part of mesh, not between the line ends.
- To define the ends of the line two options are possible:
  
  # Input of cartesian coordinates via the Macro splo.set in Macro-modus.
  # If the first option is not active, the screen coordinates of the ends of the line are defined via the mouse cursor.
  
  For INTEL: Mark the the first point(1) with a left button click of the mouse. Do the same for the second point(2).
  For FTN: Mark the the first point(1) by pressing the left button. Hold the button down until the mouse cursor is located at the position of the second point(2). Here, release the left button.
  
  If this option does not work set coordinates to zero via splo in Macro-modus.

- A 3-d formulation is possible, but note that now hidden line technic is used. Thus all visible intersections are calculated.
- The intersection bases only on the 4 corner nodes, thus 8/9 nodes are possible but the additional nodes are not used.
- Admissible screen macros for view are move, rot, isom and pers. clip may 'work'.
- The 'n1' parameter is used to specify the displacement component to be plotted. A negative value of 'n1' acts on the deformed mesh.
- The 'n2' parameter is used to specify the scaling process.

  n2 = 0  plot new value in diagram
  n2 = 1  plot other value in same diagram with rescaling
  n2 = 2  plot other value in same diagram without rescaling

- If 'n2' is < 0 the coordinates are not plotted but printed to screen and output-file.
- The 'n3' parameter is not used.
DRA W

draw,<n1,n2,n3>

Draw a line. Color and line type can be set by colo and line.

- If ’n1’ is = 0, the line is defined between node n2 and node n3.
- If ’n1’ is < 0, the line is drawn on the deformed mesh.
- If ’n1’ is = 1, the line is drawn interactiveley with the mouse. A point is defined with the left mouse button. The procedure is finished by pressing the right mouse button.
- If ’n1’ is = 2, the line is defined by coordinates of begin and end point.
EIGI

\texttt{eigi,<n1,n2,n3,n4>}

Plot the deformed mesh which is proportional to the eigenvector from inverse iteration. The use of \texttt{\textasciitilde n2} to \texttt{\textasciitilde n3} depends on what type of plot should be made. Three plots are possible.

### Eigenvector as deformed mesh

<table>
<thead>
<tr>
<th>ni</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1</td>
<td>number of eigenvector (only 1 possible), n1 &lt; 0 plot filled mesh</td>
</tr>
<tr>
<td>n2</td>
<td>n2=2 plot from numel to 1, otherwise from 1 to numel</td>
</tr>
<tr>
<td>n3</td>
<td>0</td>
</tr>
</tbody>
</table>

filled plot for one DOF of eigenvector

<table>
<thead>
<tr>
<th>ni</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1</td>
<td>number of eigenvector (only 1 possible), n1 &lt; 0 plot on defm mesh (=eigi)</td>
</tr>
<tr>
<td>n2</td>
<td>\leq 0</td>
</tr>
<tr>
<td>n3</td>
<td>dof to be plotted</td>
</tr>
</tbody>
</table>

contour plot for one DOF of eigenvector

<table>
<thead>
<tr>
<th>ni</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1</td>
<td>number of eigenvector (only 1 possible), n1 &lt; 0 plot on defm mesh (=eigi)</td>
</tr>
<tr>
<td>n2</td>
<td>&gt; 0, n2 = number of contour lines (&lt; 8)</td>
</tr>
<tr>
<td>n3</td>
<td>dof to be plotted, n3 &gt; 0 : plot with numbers</td>
</tr>
<tr>
<td>n3</td>
<td>dof to be plotted, n3 &lt; 0 : plot without numbers</td>
</tr>
</tbody>
</table>

With \texttt{\textasciitilde n4} the mesh is plotted in color \texttt{\textasciitilde n4} additionally. For a further description see the macros \texttt{stre} and \texttt{disp}.
EIGV

\[ \text{eigv}, <n1, n2, n3, n4> \]

Plot the deformed mesh which is proportional to eigenvector \('n1'\) (an eigensolution must be performed before attempting a plot!).

The use of 'n1' to 'n3' depends on what type of plot should be made. Three plots are possible.

### Eigenvector as deformed mesh

<table>
<thead>
<tr>
<th>( n_i )</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_1 )</td>
<td>number of eigenvector, ( n_1 &lt; 0 ) plot filled mesh</td>
</tr>
<tr>
<td>( n_2 )</td>
<td>( n_2 = 2 ) plot from numel to 1, otherwise from 1 to numel</td>
</tr>
<tr>
<td>( n_3 )</td>
<td>0</td>
</tr>
</tbody>
</table>

### filled plot for one DOF of eigenvector

<table>
<thead>
<tr>
<th>( n_i )</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_1 )</td>
<td>number of eigenvector, ( n_1 &lt; 0 ) plot on defm mesh (=eigv)</td>
</tr>
<tr>
<td>( n_2 )</td>
<td>( \leq 0 )</td>
</tr>
<tr>
<td>( n_3 )</td>
<td>dof to be plotted</td>
</tr>
</tbody>
</table>

### contour plot for one DOF of eigenvector

<table>
<thead>
<tr>
<th>( n_i )</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_1 )</td>
<td>number of eigenvector, ( n_1 &lt; 0 ) plot on defm mesh (=eigv)</td>
</tr>
<tr>
<td>( n_2 )</td>
<td>&gt; 0, ( n_2 ) = number of contour lines (&lt; 8)</td>
</tr>
<tr>
<td>( n_3 )</td>
<td>dof to be plotted, ( n_3 &gt; 0 ): plot with numbers</td>
</tr>
<tr>
<td>( n_3 )</td>
<td>dof to be plotted, ( n_3 &lt; 0 ): plot without numbers</td>
</tr>
</tbody>
</table>

With 'n4' the mesh is plotted in color 'n4' additionally. Eigenvectors are scaled that the maximum entry is one.

For a further description see the macros `stre` and `disp`. 
ELEM

elem,<n1,n2,n3>

Plot numbers in or near the elements appearing in the plot region visible. After a zoom some numbers may appear for elements surrounding the plot region even though no lines for element edges are plotted.

For ’n1’ less than zero the element numbers are plotted on the deformed mesh.

For ’n2’ greater than zero only the element with number n2 is plotted. For ’n2’ less than zero element with number n2 is plotted on the deformed mesh.

The element numbers are plotted in color ’n3’. Consult the colo command for tables of available colors. The default value is 3 (blue).
This macro terminates the plot modus. The same action is achieved with `quit` or `q`.
The `eplo` command may be used to construct plots of displacements of the eigenvector n3 along any specified line in the mesh for 2D-problems. 3D-problems can be calculated too if `pers` and `hide` is used. The line is then on the 3d-element faces.

- The line is scaled to 0-1 for the intersected part of mesh, not between the line ends.
- To define the ends of the line two options are possible:
  
  # Input of cartesian coordinates via the Macro `splo`, `set` in `Macro-modus`.  
  # If the first option is not active, the screen coordinates of the ends of the line are defined via the mouse cursor.  
  For INTEL: Mark the the first point(1) with a left button click of the mouse. Do the same for the second point(2).  
  For FTN: Mark the the first point(1) by pressing the left button. Hold the button down until the mouse cursor is located at the position of the second point(2). Here, release the left button.  
  If this option does not work set coordinates to zero via `splo` in `Macro-modus`.

- A 3-d formulation is possible, but note that now hidden line technic is used. Thus all visible intersections are calculated.

- The intersection bases only on the 4 corner nodes, thus 8/9 nodes are possible but the additional nodes are not used.

- Admissible screen macros for view are `move`, `rot`, `isom` and `pers`. `clip` may ‘work’.

- The ’n1’ parameter is used to specify the displacement component to be plotted. A negative value of ’n1’ acts on the deformed mesh.

- The ’n2’ parameter is used to specify the scaling process.

\[
\begin{align*}
  n2 &= 0 \quad \text{plot new value in diagram} \\
  n2 &= 1 \quad \text{plot other value in same diagram with rescaling} \\
  n2 &= 2 \quad \text{plot other value in same diagram without rescaling}
\end{align*}
\]

- If ’n2’ is < 0 the coordinates are not plotted but printed to screen and output-file.

- The ’n3’ parameter defines the number of the plotted eigenvector.
ERRO

`erro, n1,n2 >`

The distribution of error indicators is plotted with the macro `erro`.
To do this, these values have to be calculated with `erro`.
See documentation if error analysis is available for used element.
Results can be printed with `erro`.
The theoretical background is described in the Theory Manual.
The results are given with respect to a user defined error value ’n1’ in percent (default: 5 %).
’n2’ allows to choose the type of error indicator. With ’n2=1’ the Energy–based (default) and with ’n2=2’ the $L_2$–based indicator is chosen.
**Evan**

\[ \text{evan,} <n1,n2,n3> \]

Plot the eigenvector ‘\( n1 \)’ in animation mode. At first the eigenvector is shown by 8 steps which switches between \( \pm 1 \) in steps of 0.5. Then 6 steps are shown which switches between \( \pm 1 \) in steps of 1.

- For ‘\( n1 \)’ > 0 the mesh is plotted with contours whereas for ‘\( n1 \)’ < 0 the mesh is filled plotted.
- Hidden line can be achieved for ‘\( n1 \)’ < 0 by
  
  \[ \text{‘n2’} \geq 0 \text{ plot from 1 to numel} \]
  
  \[ \text{‘n2’} < 0 \text{ plot from numel to 1} \]

- The speed may depend on the computer on which **FEAP** runs. Thus the program stops (‘sleeps’) after a mesh is plotted for ‘\( n3 \)’ seconds (default = 0 sec.).
**EVEX**

`evex, <n1,n2,n3,n4>`

Plot the deformed mesh which is proportional to eigenvector of the extended system (plot can be shown only if `ext = on`).

The use of 'n2' to 'n3' depends on what type of plot should be made. Three plots are possible.

**Eigenvector as deformed mesh**

<table>
<thead>
<tr>
<th>n1</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of eigenvector (only 1 possible), n1 &lt; 0 plot filled mesh</td>
<td></td>
</tr>
<tr>
<td>n2</td>
<td>n2=2 plot from numel to 1, otherwise from 1 to numel</td>
</tr>
<tr>
<td>n3</td>
<td>0</td>
</tr>
</tbody>
</table>

filled plot for one DOF of eigenvector

<table>
<thead>
<tr>
<th>n1</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of eigenvector (only 1 possible), n1 &lt; 0 plot on defm mesh (=evex)</td>
<td></td>
</tr>
<tr>
<td>n2</td>
<td>≤ 0</td>
</tr>
<tr>
<td>n3</td>
<td>dof to be plotted</td>
</tr>
</tbody>
</table>

contour plot for one DOF of eigenvector

<table>
<thead>
<tr>
<th>n1</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of eigenvector (only 1 possible), n1 &lt; 0 plot on defm mesh (=evex)</td>
<td></td>
</tr>
<tr>
<td>n2</td>
<td>&gt; 0, n2 = number of contour lines (&lt; 8)</td>
</tr>
<tr>
<td>n3</td>
<td>dof to be plotted, n3 &gt; 0 : plot with numbers</td>
</tr>
<tr>
<td>n3</td>
<td>dof to be plotted, n3 &lt; 0 : plot without numbers</td>
</tr>
</tbody>
</table>

With 'n4' the mesh is plotted in color 'n4' additionally. For a further description see the macros **stre** and **disp**.
FACT

fact,v1

The entire plot is scaled by the value of 'v1' (default = 1.). Usually is better to scale the plot using \texttt{scal} to permit the entire deformed region to appear in the screen area.
FLUX

\texttt{flux,n1,<n2,n3,n4>}

Flow vectors for 2D- and 3D-elements based on the Laplace-Equation, e.g. heat-flux problems or groundwater problems, can be plotted for \texttt{'n1=1'}.

The principal stresses can be plotted by 'load'-vectors for \texttt{'n1=2'}.

The vectors are plotted at each nodal point.

If \texttt{'n1'} is negative the plot acts on the deformed mesh.

\texttt{'n2=2'} (default) plots for 2D-elements, \texttt{'n2=3'} plots for 3D-elements (only for Principal Stresses, must be implemented for heat flux!)

With \texttt{'n3'} the length of the vectors can be multiplied by a factor.

With \texttt{'n4'} the length of the tip can be multiplied by a factor (only 2D-stress).

Heat fluxes must be stored in elements (see manual)

2D: \(f_x, f_y\)

Stresses must be stored in elements (see manual)

2D: \(S_{xx}, S_{xy}, S_{yy}\)

3D: \(S_{xx}, S_{yy}, S_{zz}, S_{xy}, S_{xz}, S_{yz}\)

A similar action to \texttt{flux,n2} is given by the \texttt{stre} command. (\texttt{stre,27,<n2,n3,n4>}).
For beam elements a plot of the element forces may be available by the command `forc`. `n1` defines the number of force to be plotted. If `n1` is negative the force will be plotted relative to the deformed mesh (if available).

**Filled/Contour plots**

FEAP calculates 14 contour lines automatically between the extremal values. Values can be changed by `init, 2`.

In interactive mode (by `init, 1`) FEAP presents a profile of results with maximum and minimum values. Input is the number of used colors and the maximum and minimum value. Up to 14 (default) colors are possible. As default FEAP calculates the contour lines automatically between the extremal values, otherwise between the given input values.

The force can be scaled by a positive value `n3`.

For layered beam elements the number of layer for which the force `n1` should be plotted is specified by `-n3` (default = 1, which is also valid for non-layered elements).

Furthermore one can influence the plane of plotting the forces with `n1`. (This may not work for all beam elements!)

- `forc, 12` leads to plots in the local 1-2 plane.
- `forc, 13` leads to plots in the local 1-3 plane.

For -12 and -13 the plots are multiplied by -1. The initial set is the plane 13.
FRAM

fram,n1

This macro command defines a window on the screen for a plot.

<table>
<thead>
<tr>
<th>n1</th>
<th>Region used</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>entire screen used</td>
</tr>
<tr>
<td>1</td>
<td>upper left quadrant</td>
</tr>
<tr>
<td>2</td>
<td>upper right quadrant</td>
</tr>
<tr>
<td>3</td>
<td>lower left quadrant</td>
</tr>
<tr>
<td>4</td>
<td>upper right quadrant</td>
</tr>
</tbody>
</table>

The default value is n1 = 0.

Using the various frames a large amount of information may be placed on a single screen. Each 'frame' may be cleared independently for some devices by using the wipe command.
HIDE

\texttt{hide,n1}

\texttt{hide} is used to plot 3D elements (8-nodes) in a 3-d space using a hidden line technique. 20,21,27-node elements may be plotted correct, but only with respect to node 1-8. 2D shell elements (4 nodes) may occur. Further combinations with beams are not tested. In a first step the nearest faces of the elements to the viewpoint have to be calculated. In a second step the elements are sorted with respect to the distance from the viewpoint.

'\texttt{n1}' \neq 0 enables and 'n1' = 0 disables the hidden line plot.

'n1' > 0 acts on the undeformed and 'n1'< 0 acts on the deformed mesh.

For a perspective view the viewpoint is defined within \texttt{pers}, whereas for \texttt{rot} the viewpoint coordinates are chosen within this macro.

\texttt{hide} acts on \texttt{mesh,hmsh, eigv, disp, velo, acce, stre} and similar macros.

Partial parts of the mesh are plotted using \texttt{matn}.

Remark:

1) \texttt{hids} is used for 2-d shell-elements.

2) Symmetry from \texttt{symm} is not taken into account.
HIDS

\texttt{hids,n1}

**Hids** is used to plot shell elements in a 3-d space using a hidden line technique. Here the elements are sorted with respect to the distance from the viewpoint.

'\texttt{n1}' \neq 0 enables and '\texttt{n1}' = 0 disables the hidden line plot.

'\texttt{n1} > 0' acts on the undeformed and '\texttt{n1} < 0' acts on the deformed mesh.

For a perspective view the viewpoint is given within \texttt{pers}, whereas for \texttt{rot} the viewpoint coordinates are chosen within this macro.

\texttt{hids} acts on \texttt{hmsh, eigv, disp, velo, acce, stre} and similar macros.

Partial parts of the mesh are plotted using \texttt{matn}.

Remark:

1) \texttt{hide} is used for 3-d brick-elements.

2) Symmetry from \texttt{symm} is not taken into account.
HMSH

hmsh,<n1,n2,n3>

Plot the undeformed and deformed mesh in hidden line technique. The plot is correct if the elements are plotted from back to front of the screen. This can be arranged by a special element numbering technique or by using the rot-macros.

If abs ’n1’ is equal 2 elements are plotted from numel to 1, otherwise elements are plotted from 1 to numel.

Example: Define a mesh in the first quadrant in x,y,z-direction. Use the macros rot1,-70 and symm,3 and a complete hidden line mesh of the quadrants 1-4 can be seen.

The undeformed mesh is plotted for ’n1’ ≥ 0 (default), whereas the deformed mesh is plotted for ’n1’ < 0.

The mesh is plotted in color ’n2’and ’n3’. ’n3’ is the interior color (default values are 4(cyan) for the undeformed mesh and 5(green) for the deformed mesh. ’n2’ is the color of the element border. The default value is 32 (black). Consult the ‘colo’ command for tables of available colors.
**INIT**

`init, n1`

- **n1 = 0**: All plot values will be reset to its initial state.
- **n1 = 1**: Interactive calculation of contour or filled plots
- **n1 = 2**: Automatique calculation of contour or filled plots (def.)
  
  Input of plot values
  - no of lines/colors (def.=14)
  - min value (def.=min)
  - max value (def.=max)
INTS

`ints, n1, n2, <n3>`

Plot the distribution of stress $n_2$ at the center of element $n_1$ over the thickness in a shell type element. For the output $n_3$ intervals (default $n_3=10$) are used for each layer.

<table>
<thead>
<tr>
<th>Stress</th>
<th>$n_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_x$</td>
<td>1</td>
</tr>
<tr>
<td>$\sigma_y$</td>
<td>2</td>
</tr>
<tr>
<td>$\sigma_z$</td>
<td>3</td>
</tr>
<tr>
<td>$\tau_{xy}$</td>
<td>4</td>
</tr>
<tr>
<td>$\tau_{xz}$</td>
<td>5</td>
</tr>
<tr>
<td>$\tau_{yz}$</td>
<td>6</td>
</tr>
<tr>
<td>$\phi_x$</td>
<td>7</td>
</tr>
<tr>
<td>$\phi_y$</td>
<td>8</td>
</tr>
</tbody>
</table>

The stress numbers $n_2$ are defined in the following table:

These stresses could be printed with the `stre, ints` command.
ISEC

\texttt{isec,<n1,n2>}

Plot locations of all nodes on intersections. The following options are possible:

In case \texttt{`n1’} is negative the nodes are plotted relative to the deformed mesh.

In case \texttt{`n2’} is greater 0 node numbers are plotted too.
ISOM

isom

Plot all views with an 'isometric' reference frame. The other options are a 'cartesian' reference frame (see cart plot command) and a 'perspective' view (see pers plot command).

N.B. This option does not always center correctly.
JINT

\texttt{jint,<n1,n2,n3>}

Plot all nodal material forces currently calculated on the deformed mesh.

- Material forces are plotted according to the following input:
  \begin{align*}
  n1 &= 0 \quad \text{material forces (1-ndm) (default)} \\
  n1 &= i \quad \text{material force } i \\
  n1 &= \text{ndf+1} \quad \text{material forces (1-ndm) (like } n1=0) \\
  n1 &= \text{ndf+2} \quad \text{material forces (4-ndf) = moments}
  \end{align*}

- If ’n1’ is negative material forces will be positioned relative to the deformed mesh.

- If ’n2’ is nonzero the vector tip will appear next to the node whereas if ’n2’ is zero the tail of the material force vectors are on the nodes.

- In addition the results can be scaled by the value of ’n3’.

- If ’n3’ is negative all nodes with material forces are shown.
LINE

\texttt{line,n1}

Set line type to 'n1' (must be between 1–8) If 'n1' is set negative the line type will increment the previous value by 1.

<table>
<thead>
<tr>
<th>n1</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>solid</td>
</tr>
<tr>
<td>2</td>
<td>short dash</td>
</tr>
<tr>
<td>3</td>
<td>dotted</td>
</tr>
<tr>
<td>4</td>
<td>dash–dot</td>
</tr>
<tr>
<td>5</td>
<td>long dash</td>
</tr>
<tr>
<td>6</td>
<td>double dot</td>
</tr>
<tr>
<td>7</td>
<td>dot–dot–dash</td>
</tr>
<tr>
<td>8</td>
<td>solid</td>
</tr>
</tbody>
</table>

The initial (default) line type is 1 – solid.
**LINK**

`link,<n1,n2,n3>`

Plot locations of all linked nodes and linking conditions. Only nodes within the current plot region are displayed.

In case `n1` < 0 only the nodes are plotted. Furthermore with `n2` > 0 the node numbers are plotted too. Macro nodes are plotted in colour blue whereas slave nodes are plotted in colour green.

With `n1` > 0, linked nodes and linking conditions for linking dof `n1` are plotted. Repetition of calling this macro for all values of `n1` show all linking conditions, where for each `n1` a different colour is used. Results may not be correct when nodes are affected more than once! In any case of doubt check the results using only one linking card when using the macro `link`.

With `n1` = 0 all linked nodes and linking conditions are plotted in one colour.

With the value of `n3` the linking conditions can be scaled. The default value is 1.

**Remarks:**

1. Be extremely careful using the macro `link`, because input errors lead to completely wrong results!
2. When linking to fixed dofs/nodes occurs, these dofs/nodes are plotted for the macro `boun`. 
LOAD

load,<n1,n2,n3>

Plot all nodal loads (forces and moments) currently specified. The maximum length will be automatically
scaled. All other loads will be scaled in proportion to the maximum, accordingly some very small values
may scale to be too small to be visible.

Loads are plotted according to the following input:

- $n_1 = 0$ loads (1-ndm) (default)
- $n_1 = i$ load $i$
- $n_1 = ndf+1$ loads (1-ndm) (like $n_1=0$)
- $n_1 = ndf+2$ loads (4-ndf) = moments

If ‘$n_1$’ is negative loads will be positioned relative to the deformed mesh. If ‘$n_2$’ is nonzero the vector tip
will appear next to the node whereas if ‘$n_2$’ is zero the tail of the load vectors are on the nodes.

In addition the results can be scaled by the value of ‘$n_3$’.

If ‘$n_3$’ is negative all nodes with loads from load are shown.

Nodes with prescribed displacements are plotted using the pdis command.
The command redraws the FEAP–logo in color ’n1’ Consult the colo command for tables of available colors. Note that the logo vanishes for the black color (32). In a similar way the border of the plot–region is redrawn by the bord–command.
**MAGN**

`magn,<v1>`

With the macro `magn` the plot output can be magnified on the screen.

- 'v1' is the value of magnification of the mesh size. The default value for 'v1' is 0.05.
- 'v1' > 0 plots the undeformed mesh, 'v1' < 0 the deformed mesh.

- The process is controlled with the mouse buttons:
  * left button click : increase plot of mesh
  * right button click : decrease plot of mesh
  * both button click : finish the process

- The process starts with the actual values of the mesh size. The size can be set back to the initial value with the macro `init`.

- With the macro `move` the plot output can be moved on the screen.
The complete documentation is available using the macro `man`.
The undeformed or deformed mesh is plotted with respect to material properties. If \( n_1 \) and/or \( n_2 \) are omitted all element regions with different material properties are plotted. The program selects appropriate different colors for each set.

If \( n_3 \) is nonzero the mesh is added in black.

If \( n_1 \) is nonzero regions are filled belonging to properties set by \( n_1 \). A value of \( n_1 \) less than zero plots results on the deformed mesh.

Each element region can be filled in the color given by \( n_2 \). Consult the \texttt{colo} command for tables of available colors.
MATN

\texttt{matn},<n1,n2,n3>  

With the command \texttt{matn} one can define for which materials stress or contour plots should be performed. The macro can be used if there exists for example shell elements with the same coordinates. 'n1 – n3' defines the material numbers of plotted elements.

- \( n1 = 0 \) plot for all material numbers (default),
- \( n1 < 0 \) set all material numbers to zero,
- \( n1 > 0 \) plot for material numbers 'n1' to 'n2', inc='n3'.

The actual material numbers can be controlled by \texttt{show}.
\texttt{matn} acts on the macros \texttt{disp, velo, acce, dplo, eigv, eplo, evex, mate, hmsh, rplo, splo, stre, elem}

Remarks:

- For \texttt{stre,disp} etc.: Results and profile are plotted only for active materials. Other profile values could be chosen via \texttt{init,1}.
- For \texttt{stre}: No averaging is taken into account at material borders. Thus, the element code needs a special modification for isw=8.
MAXI

`maxi,<n1>`

Plot nodes with minimal/maximal values of last filled plot (e.g. `disp`, `stre`,...)

In case 'n1' is negative the nodes are plotted relative to the deformed mesh.
MESH

mesh,<n1,n2,n3>

Plot current mesh. Alternatively, in macro mode use of just `plot` will produce the undeformed plot. If 'n1' is negative the plot is in the deformed configuration at the current value of `scale`. For 'n2' zero the entire mesh is shown. If 'n2' is non-zero only material property set with the value of 'n2' is shown. On color terminals a different color will be used for each material type if 'n2' is input as the negative of the material number.

The mesh is plotted in color 'n3'. Consult the `colo` command for tables of available colors. The default value is 'n3' = 4 (cyan).
MONO

\texttt{mono,<n1>}

\texttt{mono,1} leads to a pure black and white screen with greyshading option for IBM, HP (screen and all plotfiles) and the Postscript-file under the macro 'prin'. For the VGA-screen there is no action.\texttt{mono,0} set all colors to the standard values, see the \texttt{colo}-macro.
MOVE

\texttt{move,v1,<v2, <v3>}

With the macro \texttt{move} the plot output can be moved on the screen.

- \texttt{v1} is the number of the chosen axis \([1,2]\).
- \texttt{v2} is the associated screen movement \([0-1]\). The default value for \texttt{v2} is 0.02.
- \texttt{v1} > 0 plots the undeformed mesh, \texttt{v1} < 0 the deformed mesh.
- \texttt{v3} \leq 0: The process is controlled with the mouse buttons:
  * left button click: move to left/down direction
  * right button click: move to right/up direction
  * both button click: finish the process

  This part is not available in the INTEL-Version.
- \texttt{v3} > 0: Only one step with the given values, this can be chosen e.g. in batch-mode.
- Typically the mesh is plotted. This can be suppressed via \texttt{v3}=2.
- One macro \texttt{mesh} - defining the initial values - is necessary before the process is ready to start. The position can be set back to the initial position with the macro \texttt{init}.
- With the macro \texttt{magn} the plot output can be magnified on the screen.
MOVI

This macro can be used to show the time dependency of a deformation process. For this purpose we have to calculate the complete process as a first step. Each deformed mesh has to be saved via the \texttt{writ}--macro (see the macro - manual) on the disk. In a second step we pop via the \texttt{movi} - macro all deformed meshes on the screen. The result is something which looks like a movie if you use it on a fast machine. The macro works only if the plot macros act on a separate window. Thus the following things have to be done.

1. Calculation of Problem (example)

\begin{verbatim}
 dt.,1
 prop
 writ,test
 loop.,100
 time
 tang.,1
 writ,disp
 next
 writ,clos
\end{verbatim}

2. Movie of Problem (example) (plot every fifth deformed mesh)

\begin{verbatim}
 read,test
 loop.,20
 loop.,5
 read,disp
 next
 plot,movi
 next
 read,clos
\end{verbatim}

The macro \texttt{movi} itself gives the same results as \texttt{defm}. 
**NDII**

`ndii, <n1>`

Plot locations of all nodes with problems within actual solution phase. The following options are possible:

<table>
<thead>
<tr>
<th>n1</th>
<th>plot result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>nodes with $D_{ii} &lt; 0$ (default)</td>
</tr>
<tr>
<td>2</td>
<td>nodes with $D_{ii} = 0$</td>
</tr>
<tr>
<td>3</td>
<td>nodes with $D_{ii} &lt;&lt;&lt; 1$</td>
</tr>
</tbody>
</table>

In case 'n1' is negative the nodes are plotted relative to the deformed mesh.
Plot locations of all nodes. Only nodes within the current plot region are displayed. Occasionally when a mesh is added no element edge will be plotted to each node. This is due to the fact that only elements which have all nodes visible are displayed.

In case 'n1' is negative the nodes are plotted relative to the deformed mesh.

In case 'n2' is greater than zero the node numbers are plotted too.

In case 'n2' is less than zero only the node with number n2 is plotted.

The nodes are plotted in color 'n3'. Consult the `colo` command for tables of available colors. The default value is 7 (yellow).
OUTL

\texttt{OUTL,n1,n2,n3}

Plot an outline of all parts with material number 'n2'. For 'n2' zero plot outline of entire problem. (default: n2=0).

If 'n1' is negative the outline of the deformed material number 'n2' is plotted. On color terminals the outline for each material 'n2' may be given in different colors if 'n2' is input as the negative material set number. \texttt{OUTL} will be plotted in color 'n3' (default=1).
PDIS

\texttt{pdis,<n1,n2,n3>}

Plot all prescribed nodal displacements currently in the mesh. The maximum length will be automatically scaled. All other displacements will be scaled in proportion to the maximum, accordingly some very small values may scale to be too small to be visible.

In addition the results can be scaled by the value of ‘n3’.

If ‘n1’ is negative displacements will be positioned relative to the deformed mesh. If ‘n2’ is nonzero the vector tip will appear next to the node whereas if ‘n2’ is zero the tail of the displacement vectors are on the nodes.

If ‘n3’ is negative all nodes with prescribed displacements from \texttt{pdis} are shown. Nodes with external loads are plotted using the \texttt{load} command.
PELE

pele,\langle n1 \rangle

- Pick up element numbers via left mouse button click and show material number (center of the element should be used for pick up.)
- In case ‘n1’ is negative the elements are searched relative to the deformed mesh.
- Data for that element can be found in Macro-modus with e.g. show,... .
- Admissible screen macros for view are move, rot, isom and pers. clip may ‘work’.
- Right mouse button click stops procedure.
PERS

pers,<n1>

Plot all views with an 'perspective' view. The other options are a 'cartesian' and an 'isometric' reference frame. Furthermore a rotation around axes can be done. For details see the plot commands cart, isom and rot.

After entering the command pers the user is asked interactively to specify the following:

- Coordinates of view point (X,Y,Z),
- Components of vertical vector (X,Y,Z).

After all parameters are specified the user can use the other plot commands to show mesh, stresses etc. However, the first command to be used should be zoom (without any parameters) to reset zoom options and scal to initiate pers properly.

With 'n1' = 1 hidden line views are set directly for 3D-structures (thus, the macro hide,1 is added automatically). With 'n1' = 2 same action is set for shell-structures (hids,1).
This macro is used to make plot files under PHIGS (IBM), GKS (HP) or PCX (VGA) via separate programs.

<table>
<thead>
<tr>
<th>Graphik</th>
<th>plotfile</th>
<th>n2</th>
<th>type of file</th>
<th>n3</th>
</tr>
</thead>
<tbody>
<tr>
<td>GKS</td>
<td>fplt_n1.ep</td>
<td>1 Postscript</td>
<td>1=landscape(default)</td>
<td></td>
</tr>
<tr>
<td>GKS</td>
<td>fplt_n1.cgm</td>
<td>2 CGM(bin)</td>
<td>2=portrait</td>
<td></td>
</tr>
<tr>
<td>PHIGS</td>
<td>fplt_n1.cgm</td>
<td>1 CGM(bin)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHIGS</td>
<td>fplt_n1.gdf</td>
<td>2 GDF / HPGL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VGA</td>
<td>fplt_n1.pcx</td>
<td>1 PCX</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **FPLT** is the name of the actual plot file (only characters 1-6 used).
- After opening the plotfile all of the following plot macros acts on screen and the plot–file too. This is shown by the prompt ‘Plof’.
- Up to 9 output files of each type are possible within a session.
- The plot–file can be closed via the macro plof,(0).
- Use before mono,1 to plot black on white!
- The CGM/HPGL/PS-files can be used directly or can be modified with certain programs, e.g. UNIRAS, CGMGKS, TEX(put!), PRESENTATION,...)
PNOD

\texttt{pnod, <n1,n2,n3>}

- Pick up nodal points via left mouse button click and show node number.
- In case 'n1' is negative or \texttt{scal} or \texttt{defo} is active the nodes are searched relative to the deformed mesh.
- 'n2' = 1 print in addition coordinates, displacement and stresses at that node on screen. More data may be found in Macro-modus with e.g. \texttt{disp}, \texttt{reac}, \texttt{show},... .
- 'n3' = 1 print data in addition to output file.
- Admissible screen macros for view are \texttt{move}, \texttt{rot}, \texttt{isom} and \texttt{pers}. \texttt{clip} may 'work'.
- Right mouse button click stops procedure.
Displacements, velocities, accelerations and eigenvectors can be plotted in cartesian coordinates ($n1 = 0$, default) or in polar coordinates ($n1 = ik = 12,13,23$). $i$ and $k$ describe the plane in which the transformation should be done. Afterwards the component $i$ describes the radial e.g. displacement whereas the component $k$ describes the e.g. tangential displacement.

The plot is performed in standard manner via e.g. the `disp` - macro.
This macro is used to make plot files.

<table>
<thead>
<tr>
<th>n1</th>
<th>plotfile</th>
<th>n2</th>
<th>type of file</th>
<th>n3</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1</td>
<td>fplt_n1.epi</td>
<td>1</td>
<td>Postscript</td>
<td>1=portrait(default)</td>
</tr>
<tr>
<td>n1</td>
<td>fplt_n1.pgl</td>
<td>2</td>
<td>HPGL</td>
<td>-</td>
</tr>
</tbody>
</table>

- Existing files are shown by `prin`, n1 with n1 > 99 (only for WIN).
- **FPLT** is the name of the actual plot file (Pname).
- Up to 99 output files of each type are possible within a session by different values of n1.
  - Postscript-file
    - n1 > 0 leads to a color plot
    - n1 < 0 leads to a black and white plot with grey shading
- After opening the plotfile all of the following plot macros acts on screen and the plot–file too. This is shown by the prompt `Prin`.
- The plot–file must be closed with the macro `prin,(0)`.
- The EPS/HPGL-files can be used directly or their source code can be modified with certain programs. Note that in the EPS-file all information is given for color and(!) grey shading.
PRIS

```
pris,n1,<n2,n3>,n4
```

Plot contours of principal/equivalent stresses, where 'n1' is the component to be plotted with respect to nptyp. The same parameter definitions as in stre are used.

Plots are only correct if one element type is used!

**Stresses with respect to 'n1'**

<table>
<thead>
<tr>
<th>'n1'</th>
<th>Stress for nptyp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$S_1$, $n_1$</td>
</tr>
<tr>
<td>2</td>
<td>$S_2$, $n_2$</td>
</tr>
<tr>
<td>3</td>
<td>$\alpha_s$, $\alpha_n$</td>
</tr>
<tr>
<td>4</td>
<td>$S_3$, $\alpha_n$</td>
</tr>
<tr>
<td>5</td>
<td>$m_2$</td>
</tr>
<tr>
<td>6</td>
<td>$m_2$</td>
</tr>
<tr>
<td>7</td>
<td>$\sigma_{v2D}(S)$, $\sigma_{v2D}(n)$</td>
</tr>
<tr>
<td>8</td>
<td>$\sigma_{v2D}(m)$</td>
</tr>
<tr>
<td>9</td>
<td>$\sigma_{v2,5D}$</td>
</tr>
<tr>
<td>10</td>
<td>$\sigma_{v3D}$</td>
</tr>
</tbody>
</table>

**Equivalent stresses**

\[
\begin{align*}
\sigma_{v2D} &= \sqrt{\sigma_x^2 + \sigma_y^2 - \sigma_x \sigma_y + 3 \tau_{xy}^2} \\
\sigma_{v2,5D} &= \sqrt{\sigma_x^2 + \sigma_y^2 - \sigma_x \sigma_y + 3 \tau_{xy}^2 + 3 \tau_{xz}^2 + 3 \tau_{yz}^2} \\
\sigma_{v3D} &= \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2 - \sigma_x \sigma_y - \sigma_x \sigma_z - \sigma_y \sigma_z + 3 \tau_{xy}^2 + 3 \tau_{xz}^2 + 3 \tau_{yz}^2} \\
\sigma_{v2D}(n) &= \sqrt{n_x^2 + n_y^2 - n_x n_y + 3 n_{xy}^2} \\
\sigma_{v2D}(m) &= \sqrt{m_x^2 + m_y^2 - m_x m_y + 3 m_{xy}^2} \\
\sigma_{v2,5D}(m) &= \sqrt{m_x^2 + m_y^2 - m_x m_y + 3 m_{xy}^2 + 3 q_z^2 + 3 q_y^2}
\end{align*}
\]

**Default position of Stresses in Elements**

<table>
<thead>
<tr>
<th>nptyp</th>
<th>stress position</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$S_x$, $S_{xy}$</td>
</tr>
<tr>
<td>2</td>
<td>$n_x$, $n_{xy}$</td>
</tr>
<tr>
<td>3</td>
<td>$m_x$, $m_{xy}$</td>
</tr>
<tr>
<td>4</td>
<td>$S_z$, $S_{xy}$</td>
</tr>
<tr>
<td>5</td>
<td>$n_z$, $n_{xy}$</td>
</tr>
</tbody>
</table>

Institut für Baustatik, KIT Cont. Mesh Macro Plot Mate3D Elem Nege Gmesh Cylt Isoge 332
Technical Information for User-Elements

Data have to be set on element level (isw=1) via include ’prisdat.h’ with common /prisdat/
npertyp,nprip(8).

- Titles and type has to be set via the parameter nptyp (default: nptyp=1).
- Stress positions have to be set via the array nprip default: (1,2,3,4,5,6,7,8).
PROF

prof,<n1>

The command plots the upper profile of the set of equations of the discussed problem. With \( n1 < 0 \) the lower profile is plotted too.
QUIT

quit

This macro terminates the plot modus. The same action is achieved with q or end.
Plot all nodal reactions currently calculated on the deformed mesh. Thus in an equilibrium state the boundary forces and e.g. the contact forces are plotted. At intermediate states the residual forces can be seen.

- Reactions are plotted according to the following input:
  - \( n1 = 0 \) reactions (1-ndm) (default)
  - \( n1 = i \) reaction i
  - \( n1 = ndf+1 \) reactions (1-ndm) (like \( n1=0 \))
  - \( n1 = ndf+2 \) reactions (4-ndf) = moments
- If \( n1 \) is negative reactions will be positioned relative to the deformed mesh.
- If \( n2 \) is nonzero the vector tip will appear next to the node whereas if \( n2 \) is zero the tail of the reaction vectors are on the nodes.
- In addition the results can be scaled by the value of \( n3 \).
- If \( n3 \) is negative all nodes with reactions are shown.
- **reac** prints the following values:
  - \( t=0 \): loads from mate + single loads
  - after tang: reactions
- In case of a dynamic analysis dynamic forces are included.
**RMSH**

\texttt{rmsh\langle n1,n2,n3\rangle}

Plot current mesh refinement, if the macro 'reme' has been used.

If 'n1' is negative the plot is in the deformed configuration at the current value of \texttt{scal}.

For 'n2' zero the entire mesh is shown. If 'n2' is non-zero only material property set with the value of 'n2' is shown. On color terminals a different color will be used for each material type if 'n2' is input as the negative of the material number.

The mesh is plotted in color 'n3'. Consult the \texttt{colo} command for tables of available colors. The default value is 'n3' = 5 (green).
RESI

resi,n1,n2,n3,n4

Plot contours for residuum of degree of freedom 'n1'. For negative values of 'n1' the residuum is plotted on the deformed mesh.

The macro leads to a filled plot for 'n2'= 0. On the other hand contour lines are plotted for 'n2'> 0.

**Filled/Contour plots**

FEAP calculates 14 contour lines automatically between the extremal values. Values can be changed by `init,2`.

In interactive mode (by `init,1`) FEAP presents a profile of results with maximum and minimum values. Input is the number of used colors and the maximum and minimum value. Up to 14 (default) colors are possible. As default FEAP calculates the contour lines automatically between the extremal values, otherwise between the given input values.

For contour line plots, a non–zero 'n3' value will suppress numbers near each contour line. With 'n4' the mesh is plotted in color 'n4' additionally.

For a further description see the macro `stre`. 
Besides the macros \texttt{isom} and \texttt{pers} it is possible to make a 3-D-view of a system with the \texttt{rot} macros. Thus, rotations about the 1,2,3-axis can be performed by the macros \texttt{rot1}, \texttt{rot2}, \texttt{rot3}.

'\(n1\)' is the rotation angle about the chosen axis in degree. The positive rotation vector shows in the direction of the coordinate axis.

A special 3-D view of a system can be performed by a number of \texttt{rot} commands. Thus, a sequence \texttt{rot2,90 , rot2,-40 , rot2,55} leads to a rotation angle 105 degree about axis 2.

By the macro \texttt{rot0} all rotation angles are set to zero. Thus we return to a cartesian projection.

The values of all angles can be seen by typing the macro \texttt{show}.

The rotated axis are plotted on screen for 'n2' > 0.

A combination with the \texttt{isom} and the \texttt{pers} macro may work but is not reliable. Thus before use these macros return to the cartesian view by the macro \texttt{rot0}. 

ROTM

\( \text{rotm}, <v1,v2> \)

With this macro 3-D-views of a mesh or a deformed mesh can be calculated for each mouse click.

- 'v1' is the number of the chosen axis.
- 'v1' > 0 plots the undeformed mesh, 'v1' < 0 the deformed mesh.
- 'v2' is the rotation angle about the chosen axis in degree. (def. value = 5 deg.) The positive rotation vector shows in the direction of the coordinate axis.

The process is controlled with the mouse buttons:
- left button click : rotation in positive direction
- right button click : rotation in negative direction
- both button click : finish the process

The process starts with the actual values of the rotation angles. They can be set back to zero with the macro \text{ rot0}. 
The `rplo` command may be used to construct plots of reactions along any specified line in the mesh for 2D-problems. 3D-problems can be calculated too if `pers` and `hide` is used. The line is than on the 3d-element faces.

- The line is scaled to 0-1 for the intersected part of mesh, not between the line ends.

- To define the ends of the line two options are possible:
  
  # Input of cartesian coordinates via the Macro `splo`, set in `Macro-modus`.
  
  # If the first option is not active, the screen coordinates of the ends of the line are defined via the mouse cursor.
  
  For INTEL: Mark the the first point(1) with a left button click of the mouse. Do the same for the second point(2).
  
  For FTN: Mark the the first point(1) by pressing the left button. Hold the button down until the mouse cursor is located at the position of the second point(2). Here, release the left button. If this option does not work set coordinates to zero via `splo` in `Macro-modus`.

- A 3-d formulation is possible, but note that now hidden line technic is used. Thus all visible intersections are calculated.

- The intersection bases only on the 4 corner nodes, thus 8/9 nodes are possible but the additional nodes are not used.

- Admissible screen macros for view are `move`, `rot`, `isom` and `pers`, `clip` may ‘work’.

- The ’n1’ parameter is used to specify the reaction component to be plotted. A negative value of ’n1’ acts on the deformed mesh.

- The ’n2’ parameter is used to specify the scaling process.

  
  n2 = 0 plot new value in diagram
  n2 = 1 plot other value in same diagram with rescaling
  n2 = 2 plot other value in same diagram without rescaling

- If ’n2’ is < 0 the coordinates are not plotted but printed to screen and output–file.

- The ’n3’ parameter is not used.
RSUM

\texttt{rsum,<n1,n2,n3>}

Plot locations of all nodes for sum of reaction forces. Only nodes within the current plot region are displayed.

In case ‘n1’ is negative the nodes are plotted relative to the deformed mesh.

In case ‘n2’ is greater than zero the node numbers are plotted too.

In case ‘n2’ is less than zero only the node with number n2 is plotted.

The nodes are plotted in color ‘n3’. Consult the \texttt{colo} command for tables of available colors. The default value is 2 (red).
SCAL

scal,v1

Displacements are scaled by value 'v1' (generally this operates only on elements where first two dof are in the $x_1$ and $x_2$ directions). The default value of 'v1' is 1.0 and need not be specified. After the displacement components are scaled by 'v1' the plot region is resized to permit both the undeformed and the deformed plot to appear on the screen.

There is no way to ensure that subsequent loading steps will appear in the window. Any element which is not in the plot window will not appear. For problems which undergo large motions it is possible to force the plot region to be well defined by using additional “dummy” nodes to define the expected region. These “dummy” nodes have specified coordinates but are not connected to any element. It is desirable to also restrain all degree–of–freedoms for the “dummy” nodes using appropriate boundary restraints.
**SECT**

`sect,n1,<n2,n3,n4>`

This macro is used to plot all quantities for cross sections.

n3 describes the number of cross section (default=1). (not valid for sect,12 and 13)

<table>
<thead>
<tr>
<th>n1</th>
<th>n2</th>
<th>n3</th>
<th>n4</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>&lt;n3&gt;</td>
<td></td>
<td>mesh on line element</td>
</tr>
<tr>
<td>2</td>
<td>n2</td>
<td>&lt;n3&gt;</td>
<td></td>
<td>nodes on line element (n2&gt;0 with numbers)</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>&lt;n3&gt;</td>
<td></td>
<td>element numbers on line element</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>&lt;n3&gt;</td>
<td></td>
<td>center of gravity</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>&lt;n3&gt;</td>
<td></td>
<td>center of shear</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>&lt;n3&gt;</td>
<td></td>
<td>reference point</td>
</tr>
<tr>
<td>7</td>
<td>n2</td>
<td>&lt;n3&gt;</td>
<td></td>
<td>plot function w on line element; scaling factor n2 (default = 1)</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>&lt;n3&gt;</td>
<td></td>
<td>mesh on area element</td>
</tr>
<tr>
<td>9</td>
<td>n2</td>
<td>&lt;n3&gt;</td>
<td></td>
<td>nodes on area element (n2&gt;0 with numbers)</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>&lt;n3&gt;</td>
<td></td>
<td>element numbers on area element</td>
</tr>
<tr>
<td>11</td>
<td>n2</td>
<td>&lt;n3&gt;</td>
<td></td>
<td>plot function w on area element, (n2&gt;0 lines, n2&lt;0 fill)</td>
</tr>
<tr>
<td>12</td>
<td>n2</td>
<td>n3</td>
<td>n4</td>
<td>plot stress n2 at element n3 and gauss point n4</td>
</tr>
<tr>
<td>13</td>
<td>n2</td>
<td>n3</td>
<td>n4</td>
<td>plot fluxes at element and gauss point n4, scale = n4</td>
</tr>
</tbody>
</table>
SHOW

show

This command shows the user the currently chosen parameters for the plot output.
SIZE

size,n1

Specify the size of text, numbers and labels to be plotted.

<table>
<thead>
<tr>
<th>n1</th>
<th>Text size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>small (default)</td>
</tr>
<tr>
<td>2</td>
<td>medium</td>
</tr>
<tr>
<td>3</td>
<td>large</td>
</tr>
<tr>
<td>4</td>
<td>super large</td>
</tr>
</tbody>
</table>
SLEEP

`sleep.n1`

With this macro command the program waits for `n1` seconds. This can be used within demos or presentations.
• The **splo** command may be used to construct plots of stresses along any specified line in the mesh for 2D-problems. 3D-problems can be calculated too if **pers** and **hide** is used. The line is than on the 3d-element faces.

• The line is scaled to 0-1 for the intersected part of mesh, not between the line ends.

• The line is scaled to 0-1 for the intersected part of mesh, not between the line ends.

• To define the ends of the line two options are possible:

  # Input of cartesian coordinates via the Macro **splo.set** in **Macro**-modus.
  
  # If the first option is not active, the screen coordinates of the ends of the line are defined via the mouse cursor.
  
  For INTEL: Mark the the first point(1) with a left button click of the mouse. Do the same for the second point(2).
  
  For FTN: Mark the the first point(1) by pressing the left button. Hold the button down until the mouse cursor is located at the position of the second point(2). Here, release the left button.
  
  If this option does not work set coordinates to zero via **splo** in **Macro**-modus.

• A 3-d formulation is possible, but note that now hidden line technic is used. Thus all visible intersections are calculated.

• The intersection bases only on the 4 corner nodes, thus 8/9 nodes are possible but the additional nodes are not used.

• Admissible screen macros for view are **move**, **rot**, **isom** and **pers**. **clip** may `work`.

• The `'n1'` parameter is used to specify the stress component to be plotted. A negative value of `'n1'` acts on the deformed mesh.

• The `'n2'` parameter is used to specify the scaling process.

  \[
  \begin{align*}
  n2 &= 0 \quad \text{plot new value in diagram} \\
  n2 &= 1 \quad \text{plot other value in same diagram with rescaling} \\
  n2 &= 2 \quad \text{plot other value in same diagram without rescaling}
  \end{align*}
  \]

• If `'n2'` is < 0 the coordinates are not plotted but printed to screen and output–file.

• The `'n3'` parameter is used to define the layer number for layered elements. With negative values `-n3` the stress `'n1'` for layer `'-n3'` is plotted.
STRE

\texttt{stre,n1,n2,v3,n4}

Plot contours of stresses, where ’n1’ is the component to be plotted. For negative values of ’n1’ the stresses are plotted on the deformed mesh.

The macro leads to a filled plot for ’n2= 0’. On the other hand contour lines are plotted for ’n2’\gt 0.

\textbf{Filled/Contour plots}

FEAP calculates 14 contour lines automatically between the extremal values. Values can be changed by \texttt{init,2}.

In interactive mode (by \texttt{init,1}) FEAP presents a profile of results with maximum and minimum values. Input is the number of used colors and the maximum and minimum value. Up to 14 (default) colors are possible. As default FEAP calculates the contour lines automatically between the extremal values, otherwise between the given input values.

\textbf{Layered beam/shell elements and layered solid/solid-shell elements with one element in thickness direction}

For layered elements, e.g. composite elements or elements with plastic material behavior, it is possible to specify the layer and the position in the layer for which stresses are plotted by a negative value ’v3’, the result is a filled plot. E.g. ’v3=-3.2’ plots stresses in layer 3 at position 2. For the definition of layer and position see the associated element manual or formulation.

Solids: This makes \textbf{no sense} for more than one element in thickness direction!!

For contour line plots, a non–zero ’v3’ value will suppress numbers near each contour line.

With ’n4’ the mesh is plotted in color ’n4’ additionally.

\textbf{Principal stresses}

The principal stresses can be plotted as filled or contour plots using the command \texttt{pris}.

The principal stresses can be plotted by ’load’-vectors. The associated command is \texttt{stre,27,<n2,v3,v4>}. ’n2=2’(default) plots for 2D-elements, ’n2=3’ plots for 3D-elements

With ’v3’ the length of the vectors can be multiplied by a factor.

With ’v4’ the length of the tip can be multiplied by a factor (only 2D).

Stresses must be stored in elements (see manual)

2D: $S_{xx}$, $S_{xy}$, $S_{yy}$

3D: $S_{xx}$, $S_{yy}$, $S_{zz}$, $S_{xy}$, $S_{xz}$, $S_{yz}$

A similar command is \texttt{flux,2}.

Which stresses are plotted depends on the used element. Originally \texttt{FEAP} plot the following stresses.
Details which stresses are plotted are specified in the element descriptions.

<table>
<thead>
<tr>
<th>n1</th>
<th>Component (istv&gt;0)</th>
<th>Component (istv&lt;0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11–stress</td>
<td>User generated values</td>
</tr>
<tr>
<td>2</td>
<td>12–stress</td>
<td>&quot;</td>
</tr>
<tr>
<td>3</td>
<td>22–stress</td>
<td>&quot;</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>&quot;</td>
</tr>
<tr>
<td>5</td>
<td>1–principal stress</td>
<td>&quot;</td>
</tr>
<tr>
<td>6</td>
<td>2–principal stress</td>
<td>&quot;</td>
</tr>
<tr>
<td>7</td>
<td>$\alpha$</td>
<td>&quot;</td>
</tr>
<tr>
<td>8-25</td>
<td>User generated values</td>
<td>&quot;</td>
</tr>
<tr>
<td>16-25</td>
<td>if 3D-Material library is used:</td>
<td>if 3D-Material library is used:</td>
</tr>
<tr>
<td></td>
<td>Internal variables</td>
<td>Internal variables</td>
</tr>
<tr>
<td>26</td>
<td>actual component of Principle stress</td>
<td>actual component of Principle stress</td>
</tr>
<tr>
<td>27</td>
<td>Principle stresses as vectors</td>
<td>Principle stresses as vectors</td>
</tr>
</tbody>
</table>
STR1

str1,n1

Plot elementwise constant stresses on undeformed mesh calculated from element center, where ‘n1’ is the component to be plotted. Thus, jumps occur at each element border. Which stresses are plotted depends on the used element. Stresses must be calculated in element for isw=14.

Currently str1 is implemented in elmt05, elmt06, elmt07 and elmt09.

Filled/Contour plots

FEAP calculates 14 contour lines automatically between the extremal values. Values can be changed by init,2.

In interactive mode (by init,1) FEAP presents a profile of results with maximum and minimum values. Input is the number of used colors and the maximum and minimum value. Up to 14 (default) colors are possible. As default FEAP calculates the contour lines automatically between the extremal values, otherwise between the given input values.

For contour line plots, a non-zero ’n3’ value will suppress numbers near each contour line.

With ’n4’ the mesh is plotted in color ’n4’ additionally.

For a further description see the macro stre.
SYMM

`symm, n1, n2`

It is possible to plot all quantities in additional quadrants in case of symmetric problems with the `symm` macro.

The following symmetry conditions are implemented:

<table>
<thead>
<tr>
<th>n1</th>
<th>symmetry action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>reset to first quadrant (default)</td>
</tr>
<tr>
<td>1</td>
<td>symmetry with respect to x-axis</td>
</tr>
<tr>
<td>2</td>
<td>symmetry with respect to y-axis</td>
</tr>
<tr>
<td>3</td>
<td>symmetry with respect to x + y-axis</td>
</tr>
<tr>
<td>4</td>
<td>symmetry with respect to z in xy-plane (bases on 1.Quadr.)</td>
</tr>
<tr>
<td>5</td>
<td>symmetry with respect to z in xy-plane (bases on 1.-4.Quadr.)</td>
</tr>
</tbody>
</table>

For $n2 \neq 0$ the window will not be rescaled and cleared by this macro.
TEXT

\text{<n1,n2,v3>}

Enter text to be placed in plot region. Type the desired text in the text window. To position the text move the mouse cursor in the view window at the desired location. The cursor is represented by an arrow (VGA) or cross-hairs (IBM). Press the left button to print the given text.

Use the 'n1' parameter to specify the color of each text item or use the \texttt{colo} command. The default value is 1 (white).

The 'n2' parameter is not used.

The text size can be changed using the macro \texttt{size}.

With values 'n3'> 0 it is possible to write 15 textlines in the right window directly without mouse. Here 'n3' is the desired line number. Between 1 and 15 lines are possible.

For the IBM-Version the following length of text is available

<table>
<thead>
<tr>
<th>size</th>
<th>number of characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
</tbody>
</table>

Institut für Baustatik, KIT Cont. Mesh Macro Plot Mate3D Elem Nege Gmesh Cygt Isogeo 353
TIE

tie ,<n1,n2,n3>

Plot locations of all tied nodes. Only nodes within the current plot region are displayed. Occasionally when a mesh is added no element edge will be plotted to each node. This is due to the fact that only elements which have all nodes visible are displayed.

In case 'n1' is negative the nodes are plotted relative to the deformed mesh.

In case 'n2' is greater than zero the node numbers are plotted too.

In case 'n2' is less than zero only the node with number n2 is plotted.

The nodes are plotted in color 'n3'. Consult the colo command for tables of available colors. The default value is 2 (red).
The `tplo` macro is used to plot load deflection curves from plot mode. The initialization has to be done with the macro command `tplo`. Several options are available.

The parameter `n1` may be defined as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>load plot load vs displacement</td>
</tr>
<tr>
<td>2</td>
<td>disp plot displacement vs time</td>
</tr>
<tr>
<td>3</td>
<td>phas plot velocity vs displacement (phase diagram)</td>
</tr>
<tr>
<td>4</td>
<td>reac plot reaction vs displacement 1)</td>
</tr>
<tr>
<td>5</td>
<td>detd plot determinant vs displacement 2)</td>
</tr>
<tr>
<td>6</td>
<td>velo plot velocity vs time</td>
</tr>
<tr>
<td>7</td>
<td>acce plot acceleration vs time</td>
</tr>
<tr>
<td>8</td>
<td>loat plot load vs time</td>
</tr>
<tr>
<td>9</td>
<td>reat plot reaction vs time 1)</td>
</tr>
<tr>
<td>10</td>
<td>dett plot determinant vs time 2)</td>
</tr>
<tr>
<td>11</td>
<td>tdis plot time vs displacement 4)</td>
</tr>
<tr>
<td>12</td>
<td>strd plot stress i at node k vs displacement 3)</td>
</tr>
<tr>
<td>13</td>
<td>strt plot stress i at node k vs time 3)</td>
</tr>
<tr>
<td>14</td>
<td>us1d plot value1 vs displacement 4)</td>
</tr>
<tr>
<td>15</td>
<td>us2d plot value2 vs displacement 4)</td>
</tr>
<tr>
<td>16</td>
<td>us1t plot value1 vs time 4)</td>
</tr>
<tr>
<td>17</td>
<td>us2t plot value2 vs time 4)</td>
</tr>
</tbody>
</table>

1. The nodal reactions have to be calculated within each time step via `reac`.
2. Only within arc-length method, extended system, or using SM-solver or if calculated by the macro `detk`.
3. Stress i at node k has to be defined with the macro `tplo,sset`. Furthermore stress has to be calculated via `stre,node` for each time step.
4. Values have to be provided by user. More informations are given in the chapter Adding elements.

Further options are available using the macro command `tplo`. 
**TRAJ**

\[ \text{traj,<n1,n2,n3>} \]

This macro plots the trajectories of main stresses by crosses in the nodes.
If 'n1' is not equal zero, abs('n1') is the number of main stress, i.e. 1,2,(3)
If 'n1' is negative the plot acts on the deformed mesh.
'n2' multiplies the length of the lines by a factor.
'n3=2'(default) plots for 2D-elements, 'n3=3' plots for 3D-elements

Stresses must be stored in elements (see manual)
2D: \( S_{xx}, S_{xy}, S_{yy} \)
3D: \( S_{xx}, S_{yy}, S_{zz}, S_{xy}, S_{xz}, S_{yz} \)
UEIG

\texttt{ueig<n1,n2,n3>}

This macro is up to now not documented.
 Plot contours for degree of freedom (i.e. velocity) ’n1’. For negative values of ’n1’ the velocity are plotted on the deformed mesh.

The macro leads to a filled plot for ’n2’= 0. On the other hand contour lines are plotted for ’n2’> 0.

**Filled plots**

In interactive mode **FEAP** presents a profile of results with maximum and minimum values. Then **FEAP** asks for the number of used colors and the maximum and minimum value. Up to 14 colors are possible. The default value for the number of colors is 14. As default **FEAP** calculates the contour lines automatically between the extremal values, otherwise between the given input values.

**Contour plots**

In interactive mode **FEAP** presents a profile of results with maximum and minimum values. Then **FEAP** asks for the number of used contours and the maximum and minimum value. Up to 14 contours are possible. The default value for the number of contours is 14. As default **FEAP** calculates the contour lines automatically between the extremal values, otherwise between the given input values.

For contour line plots, a non-zero ’n3’ value will suppress numbers near each contour line.

With ’n4’ the mesh is plotted in color ’n4’ additionally.

Compare the handling of this macro with **stre**.

In batch mode, the requisite data described above must appear after the **end** macro command in the order that the program needs inputs.
**WIPE**

`wipe, n1`

`wipe` clears the screen. With ‘n1’ it is possible to decide which part of the screen is cleared. The following values are possible for ‘n1’

<table>
<thead>
<tr>
<th>n1</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>entire screen</td>
</tr>
<tr>
<td>1</td>
<td>clear part of screen between points 1 and 2</td>
</tr>
<tr>
<td>2</td>
<td>clear part of screen outside points 1 and 2</td>
</tr>
</tbody>
</table>

The default value is n1 = 0.

**Remark:**
The input of points 1 and 2 is done via the mouse cursor. For ‘n1’ = 1,2 the screen is not really cleared, but filled with the background color (usually black=32).
XSCA(L)

\texttt{xsc}(l),v1,v2,v3

The system (including deformations) can be scaled independently in the directions $x_1$, $x_2$ (and $x_3$). The default values are 1.0 for all directions.

After the system is scaled by ‘v1, v2, v3 ’ the plot region is resized to permit both the undeformed and the deformed plot to appear on the screen by the macro \texttt{scal}.

There is no way to ensure that subsequent loading steps will appear in the window. Any element which is not in the plot window will not appear. For problems which undergo large motions it is possible to force the plot region to be well defined by using additional “dummy” nodes to define the expected region. These “dummy” nodes have specified coordinates but are not connected to any element. It is desirable to also restrain all degree–of–freedoms for the “dummy” nodes using appropriate boundary restraints.
The **zoom** feature permits the user to plot part of mesh and or results. If the node numbers are known then the region to be plotted may be identified by specifying:

<table>
<thead>
<tr>
<th>n1</th>
<th>– one node defining region to zoom on.</th>
</tr>
</thead>
<tbody>
<tr>
<td>n2</td>
<td>– other node for zoom.</td>
</tr>
<tr>
<td>n1 = n2 = 0</td>
<td>– gives entire mesh.</td>
</tr>
</tbody>
</table>

If coordinates are known then the region to be plotted may be identified by specifying: **zoom,,,1**

and add the following values: \((x, y, z)_1, (x, y, z)_2\).

The **zoom** without numbers is required to return to the entire mesh as the plot region. This option is useful to rescale the plot to scale = 1.0 after it has been scaled to ‘v1’ by the **scal** macro.
Chapter 5

3D-Material library

- Theory

For some elements, e.g. ELMT21 exist a library of different 3D-material models. Currently the following models are implemented.

<table>
<thead>
<tr>
<th>Material</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>linear elastic isotropic</td>
</tr>
<tr>
<td>2</td>
<td>linear elastic orthotropic</td>
</tr>
<tr>
<td>3</td>
<td>linear elastic transversal isotropic</td>
</tr>
<tr>
<td>4</td>
<td>small elasto-(visco-)plastic strains</td>
</tr>
<tr>
<td>5</td>
<td>finite elasto-plastic strains $F = F_e F_p$</td>
</tr>
<tr>
<td>6</td>
<td>finite elastic strains, Ogden</td>
</tr>
<tr>
<td>7</td>
<td>Method of Cells (MOC), Aboudi</td>
</tr>
<tr>
<td>8</td>
<td>FE$^2$</td>
</tr>
<tr>
<td>9</td>
<td>small strain isotropic damage</td>
</tr>
<tr>
<td>10</td>
<td>concrete model, Schütt</td>
</tr>
<tr>
<td>11</td>
<td>small strain visco-elastic</td>
</tr>
<tr>
<td>12</td>
<td>linear piezoelectric</td>
</tr>
<tr>
<td>13</td>
<td>dielectric elastomers</td>
</tr>
<tr>
<td>14</td>
<td>finite elastic strains, Blatz-Ko</td>
</tr>
<tr>
<td>15</td>
<td>transversal isotropic with damage</td>
</tr>
<tr>
<td>16</td>
<td>functionally graded linear elastic</td>
</tr>
</tbody>
</table>

Details on the derivation of these material models can be found in the Theory Manual. Details on the implementation in elements can be found in the Manual for adding elements.
• Material input data/ output plot

Material 1: linear elastic isotropic, (Theory)

<table>
<thead>
<tr>
<th>Record 1</th>
<th>linear elastic isotropic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$</td>
<td>$[F/L^2]$ Young’s modulus</td>
</tr>
<tr>
<td>$\nu$</td>
<td>– Poisson’s ratio</td>
</tr>
</tbody>
</table>

Material 2: linear elastic orthotropic, (Theory)

<table>
<thead>
<tr>
<th>Record 1</th>
<th>linear elastic orthotropic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{11}$</td>
<td>$[F/L^2]$ Young’s modulus for direction 1</td>
</tr>
<tr>
<td>$E_{22}$</td>
<td>$[F/L^2]$ Young’s modulus for direction 2</td>
</tr>
<tr>
<td>$E_{33}$</td>
<td>$[F/L^2]$ Young’s modulus for direction 3</td>
</tr>
<tr>
<td>$\nu_{12}$</td>
<td>– Poisson’s ratio</td>
</tr>
<tr>
<td>$\nu_{13}$</td>
<td>– Poisson’s ratio</td>
</tr>
<tr>
<td>$\nu_{23}$</td>
<td>– Poisson’s ratio</td>
</tr>
<tr>
<td>$G_{12}$</td>
<td>$[F/L^2]$ Shear modulus</td>
</tr>
<tr>
<td>$G_{13}$</td>
<td>$[F/L^2]$ Shear modulus</td>
</tr>
<tr>
<td>$G_{23}$</td>
<td>$[F/L^2]$ Shear modulus</td>
</tr>
<tr>
<td>$\varphi$</td>
<td>$[\degree]$ Rotation angle in 1–2 plane, a layered element resets this value for each layer.</td>
</tr>
</tbody>
</table>

Material 3: linear elastic transversal isotropic, (Theory)

<table>
<thead>
<tr>
<th>Record 1</th>
<th>linear elastic transversal isotropic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_1$</td>
<td>$[F/L^2]$ Young’s modulus for direction 1</td>
</tr>
<tr>
<td>$E_2$</td>
<td>$[F/L^2]$ Young’s modulus for transverse directions 2/3 $(E_3 = E_2)$</td>
</tr>
<tr>
<td>$\nu_{12}$</td>
<td>– Poisson’s ratio $(\nu_{13} = \nu_{12})$</td>
</tr>
<tr>
<td>$G_{12}$</td>
<td>$[F/L^2]$ Shear modulus $(G_{13} = G_{12})$</td>
</tr>
<tr>
<td>$G_{23}$</td>
<td>$[F/L^2]$ Shear modulus $(\nu_{23} = E_2/(2G_{23}) - 1)$</td>
</tr>
<tr>
<td>$\varphi$</td>
<td>$[\degree]$ Rotation angle in 1–2 plane, a layered element resets this value for each layer.</td>
</tr>
</tbody>
</table>

Material 4: elasto-(visco-)plastic isotropic small strains, (Theory)

<table>
<thead>
<tr>
<th>Record 1</th>
<th>elasto-(visco-)plastic isotropic small strains</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$</td>
<td>$[F/L^2]$ Young’s modulus</td>
</tr>
<tr>
<td>$\nu$</td>
<td>– Poisson’s ratio $(\nu_{13} = \nu_{12})$</td>
</tr>
<tr>
<td>$Y_0$</td>
<td>$[F/L^2]$ initial yield stress</td>
</tr>
<tr>
<td>$Y_\infty$</td>
<td>$[F/L^2]$ yield stress at $t = \infty$</td>
</tr>
<tr>
<td>$xh$</td>
<td>$[F/L^2]$ linear hardening modulus</td>
</tr>
<tr>
<td>$xd$</td>
<td>– exponential hardening value</td>
</tr>
<tr>
<td>$\eta$</td>
<td>$[FT/L^2]$ viscosity</td>
</tr>
</tbody>
</table>

yield condition: $f = \sqrt{3/2}S_D : S_D - Y_0 + xh \cdot a + (Y_\infty - Y_0) \cdot (1 - e^{-xd \cdot a})$
with $S_D$: deviatoric stresses and $a$: equivalent plastic strains
### Material 5: elasto-plastic isotropic finite strains, (Theory)

<table>
<thead>
<tr>
<th>Record 1</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$</td>
<td>$[F/L^2]$</td>
</tr>
<tr>
<td>$\nu$</td>
<td>$-$</td>
</tr>
<tr>
<td>$Y_0$</td>
<td>$[F/L^2]$</td>
</tr>
<tr>
<td>$Y_\infty$</td>
<td>$[F/L^2]$</td>
</tr>
<tr>
<td>$x_h$</td>
<td>$[F/L^2]$</td>
</tr>
<tr>
<td>$x_d$</td>
<td>$-$</td>
</tr>
</tbody>
</table>

yield condition: $f = \sqrt{3/2} S_D : S_D - Y_0 + x_h \cdot a + (Y_\infty - Y_0) \cdot (1 - e^{-x_d a})$

with $S_D$: deviatoric stresses and $a$: equivalent plastic strains

### Material 6: finite strain elastic, Ogden, (Theory)

<table>
<thead>
<tr>
<th>Record 1</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_1$</td>
<td>$[F/L^2]$</td>
</tr>
<tr>
<td>$\mu_2$</td>
<td>$[F/L^2]$</td>
</tr>
<tr>
<td>$\mu_3$</td>
<td>$[F/L^2]$</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>$-$</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>$-$</td>
</tr>
<tr>
<td>$\alpha_3$</td>
<td>$-$</td>
</tr>
<tr>
<td>$\Lambda$</td>
<td>$[F/L^2]$</td>
</tr>
</tbody>
</table>

$\sum_{i=1}^{3} (\mu_i \cdot \alpha_i) = 2\mu$
Material 7: Method of Cells (MOC), Aboudi, *(Theory)*

<table>
<thead>
<tr>
<th>Record 1</th>
<th>Method of Cells (MOC), Aboudi</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{11}$</td>
<td>$[F/L^2]$ Young’s modulus fiber direction 1</td>
</tr>
<tr>
<td>$E_{22}$</td>
<td>$[F/L^2]$ Young’s modulus fiber direction 2</td>
</tr>
<tr>
<td>$E_{33}$</td>
<td>$[F/L^2]$ Young’s modulus fiber direction 3</td>
</tr>
<tr>
<td>$\nu_{12}$</td>
<td>$-$ Poisson’s ratio fiber</td>
</tr>
<tr>
<td>$\nu_{13}$</td>
<td>$-$ Poisson’s ratio fiber</td>
</tr>
<tr>
<td>$\nu_{23}$</td>
<td>$-$ Poisson’s ratio fiber</td>
</tr>
<tr>
<td>$G_{12}$</td>
<td>$[F/L^2]$ Shear modulus fiber</td>
</tr>
<tr>
<td>$G_{13}$</td>
<td>$[F/L^2]$ Shear modulus fiber</td>
</tr>
<tr>
<td>$G_{23}$</td>
<td>$[F/L^2]$ Shear modulus fiber</td>
</tr>
<tr>
<td>$\alpha_{11}^{t}$</td>
<td>$[1/K]$ coefficient of thermal expansion fiber direction 1</td>
</tr>
<tr>
<td>$\alpha_{22}^{t}$</td>
<td>$[1/K]$ coefficient of thermal expansion fiber direction 2</td>
</tr>
<tr>
<td>$\alpha_{33}^{t}$</td>
<td>$[1/K]$ coefficient of thermal expansion fiber direction 3</td>
</tr>
<tr>
<td>$E$</td>
<td>$[F/L^2]$ Young’s modulus matrix</td>
</tr>
<tr>
<td>$\nu$</td>
<td>$-$ Poisson’s ratio matrix</td>
</tr>
<tr>
<td>$G$</td>
<td>$[F/L^2]$ Shear modulus matrix</td>
</tr>
<tr>
<td>$\alpha_{t}$</td>
<td>$[1/K]$ coefficient of thermal expansion matrix</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Record 2</th>
<th>Method of Cells (MOC), Aboudi</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{11}$</td>
<td>$[F/L]$ spring stiffness direction 1 (Penalty parameter)</td>
</tr>
<tr>
<td>$C_{22}$</td>
<td>$[F/L]$ spring stiffness direction 2</td>
</tr>
<tr>
<td>$C_{33}$</td>
<td>$[F/L]$ spring stiffness direction 3</td>
</tr>
<tr>
<td>$\Delta T$</td>
<td>$[K]$ Temperature change</td>
</tr>
<tr>
<td>$V_F$</td>
<td>$-$ fiber volume fraction</td>
</tr>
<tr>
<td>$Y_{tF}$</td>
<td>$[F/L^2]$ tensile strength fiber</td>
</tr>
<tr>
<td>$Y_{cF}$</td>
<td>$[F/L^2]$ compressive strength fiber</td>
</tr>
<tr>
<td>$Y_{tM}$</td>
<td>$[F/L^2]$ tensile strength matrix</td>
</tr>
<tr>
<td>$Y_{cM}$</td>
<td>$[F/L^2]$ compressive strength matrix</td>
</tr>
<tr>
<td>$Y_{sM}$</td>
<td>$[F/L^2]$ shear strength matrix</td>
</tr>
<tr>
<td>$n_C$</td>
<td>$-$ number of cells, at present only 4</td>
</tr>
<tr>
<td>$\varphi$</td>
<td>$^\circ$ Rotation angle in 1–2 plane, a layered element resets this value for each layer.</td>
</tr>
</tbody>
</table>
Material 8: FE$^2$, (Theory)

<table>
<thead>
<tr>
<th>Record 1</th>
<th>FE$^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fmicro</td>
<td>Path and name of used RVE (micro problem) (name without extensions .i)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Record 2</th>
<th>FE$^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fbatch</td>
<td>Path and name of batch file to initiate data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Record 3</th>
<th>FE$^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ittyp</td>
<td>0 = restart files for each GP and element of micro-RVE 1 = one restart file for micro-RVE</td>
</tr>
<tr>
<td>irecl</td>
<td>Length of restart file for each GP of micro-RVE in BYTES (only ittyp=1)</td>
</tr>
</tbody>
</table>

Up to 10 different RVE’s could be used. For that it is necessary to use material numbers 1-10 in MATE!

irecl can be calculated using one TANG of the macro-problem. An error will occur, but the length of the restart file could be seen in the output file (irecl) and e.g. the Windows Explorer. FEAP, INTEL- and SALFORD-Compiler produce slightly different file lengths. The maximum value should be used.

Alternatively the MICRO-Problem could be started with one TANG and EXIT.

Material 9: small strain isotropic damage, (Theory)

<table>
<thead>
<tr>
<th>Record 1</th>
<th>small strain isotropic damage</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>$[F/L^2]$ Young’s modulus</td>
</tr>
<tr>
<td>$\nu$</td>
<td>$[-]$ Poisson’s ratio</td>
</tr>
<tr>
<td>$\sigma_u$</td>
<td>$[F/L^2]$ ultimate stress</td>
</tr>
<tr>
<td>$G_f$</td>
<td>$[F/L]$ fracture energy</td>
</tr>
<tr>
<td>implexx</td>
<td>$[-]$ 0=implizit, 1=implizit-explizit</td>
</tr>
<tr>
<td>$\eta$</td>
<td>$[FT/L^2]$ viscosity</td>
</tr>
<tr>
<td>$\alpha = 1$</td>
<td>$[-]$ backward Euler method</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>stre,i</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>r</td>
<td>internal variable</td>
</tr>
<tr>
<td>17</td>
<td>d</td>
<td>damage variable</td>
</tr>
<tr>
<td>18</td>
<td>$</td>
<td>\varepsilon</td>
</tr>
</tbody>
</table>

For 2D-problems modify characteristic length $\ell_c$ in subroutine Mate3d09.for.
Material 10: concrete model, (Theory)

<table>
<thead>
<tr>
<th>Record 1</th>
<th>concrete model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$</td>
<td>$[F/L^2]$</td>
</tr>
<tr>
<td>$\nu$</td>
<td>$-$</td>
</tr>
<tr>
<td>ipla</td>
<td>$-$</td>
</tr>
<tr>
<td>$f_{ctm}$</td>
<td>$[F/L^2]$</td>
</tr>
<tr>
<td>$f_{cm}$</td>
<td>$[F/L^2]$</td>
</tr>
<tr>
<td>$G_f$</td>
<td>$[F/L]$</td>
</tr>
<tr>
<td>$G_c$</td>
<td>$[F/L]$</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>$-$</td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>$-$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\text{stre}_i$</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>$\alpha_1$</td>
<td>equivalent plastic strain 1</td>
</tr>
<tr>
<td>17</td>
<td>$\alpha_2$</td>
<td>equivalent plastic strain 2</td>
</tr>
</tbody>
</table>

Material 11: small strain visco-elastic, (Theory)

<table>
<thead>
<tr>
<th>Record 1</th>
<th>small strain visco-elastic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$</td>
<td>$[F/L^2]$</td>
</tr>
<tr>
<td>$\nu$</td>
<td>$-$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>$[1/K]$</td>
</tr>
<tr>
<td>$n v$</td>
<td>$-$</td>
</tr>
<tr>
<td>$n m$</td>
<td>$-$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$n v$ Records</th>
<th>small strain visco-elastic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_i$</td>
<td>$-$</td>
</tr>
<tr>
<td>$\tau_i$</td>
<td>$[T]$</td>
</tr>
</tbody>
</table>

$\mu_0 = 1 - \sum_{i=1}^{n v} \mu_i \geq 1$ with $0 \leq \mu_i \leq 1$

<table>
<thead>
<tr>
<th>$\text{stre}_i$</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>$\varepsilon_{v1}$</td>
<td>viscoelastic strain 1</td>
</tr>
<tr>
<td>17</td>
<td>$\varepsilon_{v2}$</td>
<td>viscoelastic strain 2</td>
</tr>
<tr>
<td>18</td>
<td>$\varepsilon_{v3}$</td>
<td>viscoelastic strain 3</td>
</tr>
<tr>
<td>19</td>
<td>$\varepsilon_{v4}$</td>
<td>viscoelastic strain 4</td>
</tr>
<tr>
<td>20</td>
<td>$\varepsilon_{v5}$</td>
<td>viscoelastic strain 5</td>
</tr>
<tr>
<td>21</td>
<td>$\varepsilon_{v6}$</td>
<td>viscoelastic strain 6</td>
</tr>
<tr>
<td>22</td>
<td>$\varepsilon_{v7}$</td>
<td>viscoelastic strain 7</td>
</tr>
<tr>
<td>23</td>
<td>$\varepsilon_{v8}$</td>
<td>viscoelastic strain 8</td>
</tr>
<tr>
<td>24</td>
<td>$\varepsilon_{v9}$</td>
<td>viscoelastic strain 9</td>
</tr>
<tr>
<td>25</td>
<td>$\varepsilon_{v10}$</td>
<td>viscoelastic strain 10</td>
</tr>
</tbody>
</table>
Material 12: linear piezoelectric, *(Theory)*

<table>
<thead>
<tr>
<th>Record 1</th>
<th>linear piezoelectric</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$ [F/L$^2$]</td>
<td>Young’s modulus</td>
</tr>
<tr>
<td>$\nu$</td>
<td>Poisson’s ratio</td>
</tr>
<tr>
<td>$\epsilon_{13}$ [Q/L$^2$]</td>
<td>piezoelectric coupling modulus</td>
</tr>
<tr>
<td>$\epsilon_{33}$ [Q/L$^2$]</td>
<td>piezoelectric coupling modulus</td>
</tr>
<tr>
<td>$\epsilon_{15}$ [Q/L$^2$]</td>
<td>piezoelectric coupling modulus</td>
</tr>
<tr>
<td>$\varepsilon$ [Q$^3$/F L$^4$]</td>
<td>dielectric constant</td>
</tr>
</tbody>
</table>

Material 13: dielectric elastomers, *(Theory)*

Material 14: finite elastic strains, Blatz-Ko, *(Theory)*

<table>
<thead>
<tr>
<th>Record 1</th>
<th>Blatz-Ko</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$ [F/L$^2$]</td>
<td>Shear modulus</td>
</tr>
<tr>
<td>$\nu$</td>
<td>Poisson’s ratio ($0 \leq \nu &lt; 0.5$)</td>
</tr>
<tr>
<td>$f$</td>
<td>Interpolation parameter ($0 \leq f \leq 1$)</td>
</tr>
</tbody>
</table>

$f = 1$: compressible NEO-HOOKE, $f < 1$: compressible MONEY-RIVLIN

Material 15: Transversal isotropic with damage, *(Theory)*

<table>
<thead>
<tr>
<th>Record 1</th>
<th>transversal isotropic with damage</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{11}$ [F/L$^2$]</td>
<td>Young’s modulus for direction 1</td>
</tr>
<tr>
<td>$E_{22}$ [F/L$^2$]</td>
<td>Young’s modulus for transverse directions 2/3 ($E_3 = E_2$)</td>
</tr>
<tr>
<td>$\nu_{12}$</td>
<td>Poisson’s ratio ($\nu_{13} = \nu_{12}$)</td>
</tr>
<tr>
<td>$G_{12}$ [F/L$^2$]</td>
<td>Shear modulus ($G_{13} = G_{12}$)</td>
</tr>
<tr>
<td>$G_{23}$ [F/L$^2$]</td>
<td>Shear modulus ($\nu_{23} = E_2/(2G_{23}) - 1$)</td>
</tr>
<tr>
<td>$\varphi$ [$^\circ$]</td>
<td>Rotation angle in 1–2 plane, a layered element resets this value for each layer.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Record 2 Pos 1-6</th>
<th>transversal isotropic with damage</th>
</tr>
</thead>
<tbody>
<tr>
<td>iftyp</td>
<td>Failure criterion</td>
</tr>
<tr>
<td></td>
<td>1 = Tsai-Wu 2 = Hashin (extended) 3 = Puck 4 = Cuntze</td>
</tr>
<tr>
<td>$R_{11}$ [F/L$^2$]</td>
<td>Tensile strength fiber direction</td>
</tr>
<tr>
<td>$R_{12}$ [F/L$^2$]</td>
<td>Compressive strength fiber direction</td>
</tr>
<tr>
<td>$R_{22}$ [F/L$^2$]</td>
<td>Tensile strength matrix direction</td>
</tr>
<tr>
<td>$R_{23}$ [F/L$^2$]</td>
<td>Compressive strength matrix direction</td>
</tr>
<tr>
<td>$R_{12}$ [F/L$^2$]</td>
<td>Shear strength direction 12</td>
</tr>
</tbody>
</table>

with respect to failure criterion:

<table>
<thead>
<tr>
<th>Record 2 Pos 7-9</th>
<th>iftyp 1-Tsai-Wu, 2-Hashin</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{13}$ [F/L$^2$]</td>
<td>Shear strength direction 13</td>
</tr>
<tr>
<td>$R_{23}$ [F/L$^2$]</td>
<td>Shear strength direction 23</td>
</tr>
</tbody>
</table>
### 3D-MATERIAL LIBRARY

**Record 2 Pos 7-9**

<table>
<thead>
<tr>
<th>iftyp 3-Puck</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{i\perp}^+$ [-] Curve fitting parameter: 0.30-0.35</td>
</tr>
<tr>
<td>$p_{i\perp}$ [-] Curve fitting parameter: 0.25-0.30</td>
</tr>
</tbody>
</table>

**Record 2 Pos 7-9**

<table>
<thead>
<tr>
<th>iftyp 4-Cuntze</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_{i\perp}$ [-] Curve fitting parameter: 0.05-0.15</td>
</tr>
<tr>
<td>$m_i$ [-] Interaction parameter: 2.5-3.5</td>
</tr>
</tbody>
</table>

**Record 3 Pos1**

<table>
<thead>
<tr>
<th>Degradation model typ</th>
</tr>
</thead>
<tbody>
<tr>
<td>idtyp [-] =1: Constant(CRC-ACS model): for 1-Tsai-Wu</td>
</tr>
<tr>
<td>=2: Constant(Chang &amp; Lessard model): for 2-Hashin</td>
</tr>
<tr>
<td>=3: Constant: for 3-Puck and 4-Cuntze</td>
</tr>
<tr>
<td>=4: Gradual: for 3-Puck</td>
</tr>
<tr>
<td>=5: Gradual: for 4-Cuntze</td>
</tr>
</tbody>
</table>

### IFTYP=1

#### TSAI-WU

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stre,1</td>
<td></td>
</tr>
<tr>
<td>16 $iv_1$ [-] damaged</td>
<td></td>
</tr>
</tbody>
</table>

### IFTYP=2

#### HASHIN

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stre,1</td>
<td></td>
</tr>
<tr>
<td>16 $M^t$ [-] tensile matrix failure</td>
<td></td>
</tr>
<tr>
<td>17 $M^c$ [-] compressive matrix failure</td>
<td></td>
</tr>
<tr>
<td>18 $FM$ [-] matrix shear failure</td>
<td></td>
</tr>
<tr>
<td>19 $MFM$ [-] $= M^t + FM$ or $= M^c + FM$</td>
<td></td>
</tr>
<tr>
<td>20 $FF$ [-] Fiber fracture</td>
<td></td>
</tr>
</tbody>
</table>

### IFTYP=3

#### PUCK

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stre,1</td>
<td></td>
</tr>
<tr>
<td>16 $\sum \text{Mode}_i$ [-] Matrix cracking</td>
<td></td>
</tr>
<tr>
<td>17 Mode A [-] tensile matrix failure</td>
<td></td>
</tr>
<tr>
<td>18 Mode B [-] matrix shear failure</td>
<td></td>
</tr>
<tr>
<td>19 Mode C [-] compressive matrix failure</td>
<td></td>
</tr>
<tr>
<td>20 $FF$ [-] Fiber fracture</td>
<td></td>
</tr>
</tbody>
</table>

### IFTYP=4

#### CUNTZE

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stre,1</td>
<td></td>
</tr>
<tr>
<td>16 $\sum \text{IFF}_i$ [-] Matrix cracking</td>
<td></td>
</tr>
<tr>
<td>17 IFF 1 [-] tensile matrix failure</td>
<td></td>
</tr>
<tr>
<td>18 IFF 2a [-] matrix shear failure $\sigma_{22} \geq 0$</td>
<td></td>
</tr>
<tr>
<td>19 IFF 2b [-] matrix shear failure $\sigma_{22} &lt; 0$</td>
<td></td>
</tr>
<tr>
<td>20 IFF 3 [-] compressive matrix failure</td>
<td></td>
</tr>
<tr>
<td>21 $FF$ [-] Fiber fracture</td>
<td></td>
</tr>
</tbody>
</table>

For all damage variables: $0 \leq d \leq 1$
Material 16: functionally graded linear elastic, (Theory)

\[ E(x, y, z) = E(z) = E_0 \cdot (c_0 + c_1 \cdot z + c_2 \cdot z^2) \]

<table>
<thead>
<tr>
<th>Record 1</th>
<th>funct. graded linear elastic</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_0 )</td>
<td>( F/L^2 )</td>
</tr>
<tr>
<td>( \nu )</td>
<td>-</td>
</tr>
<tr>
<td>( c_0 )</td>
<td>-</td>
</tr>
<tr>
<td>( c_1 )</td>
<td>-</td>
</tr>
<tr>
<td>( c_2 )</td>
<td>-</td>
</tr>
</tbody>
</table>
## Chapter 6

### Elements in Student Version

#### 6.1 Available elements

<table>
<thead>
<tr>
<th>El. Nr.</th>
<th>Aufgabe</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2D/3D-Fachwerkelement</td>
</tr>
<tr>
<td>2</td>
<td>2D-Timoshenko - Balkenelement</td>
</tr>
<tr>
<td>3</td>
<td>2D-Bernoulli - Balkenelement</td>
</tr>
<tr>
<td>4</td>
<td>3D-Bernoulli - Balkenelement</td>
</tr>
<tr>
<td>5</td>
<td>3-9 Knoten Scheibenelement (ESZ, EVZ, Axisym)</td>
</tr>
<tr>
<td>6</td>
<td>4 Knoten Scheibenelement (ESZ, Pian/Sumihara)</td>
</tr>
<tr>
<td>7</td>
<td>4 Knoten DKQ-Platten element</td>
</tr>
<tr>
<td>8</td>
<td>Achssymmetrisches Schalenelement</td>
</tr>
<tr>
<td>9</td>
<td>3D Schalenelement</td>
</tr>
<tr>
<td>10</td>
<td>Punktelement fuer Federn/Massen</td>
</tr>
<tr>
<td>11</td>
<td>3D-Timoshenko - Balkenelement - exzentrisch</td>
</tr>
<tr>
<td>12</td>
<td>Wärmeleitung o. Grundwasserströmung</td>
</tr>
<tr>
<td>13</td>
<td>flache Schale</td>
</tr>
<tr>
<td>14</td>
<td>2D-Knoten-Knoten-Kontaktelement</td>
</tr>
<tr>
<td>15</td>
<td>3D-Timoshenko - Balkenelement linear/nichtlinear</td>
</tr>
<tr>
<td>16</td>
<td>3D Seilelement</td>
</tr>
<tr>
<td>17</td>
<td>4 Knoten Bathe/Dvorkin-Plattenelement</td>
</tr>
<tr>
<td>18</td>
<td>4/8/9 Knoten SRI-Plattenelement</td>
</tr>
<tr>
<td>19</td>
<td>3 Knoten DKT-Plattenelement</td>
</tr>
<tr>
<td>20</td>
<td>Trägerrost Bernoulli/St.Venant Element</td>
</tr>
<tr>
<td>21</td>
<td>3D Volumenelement</td>
</tr>
<tr>
<td>22</td>
<td>4/8/9 Knoten Scheibenelement</td>
</tr>
<tr>
<td>23</td>
<td>4 Knoten IDKQ-Plattenelement</td>
</tr>
<tr>
<td>24</td>
<td>Schubspannungen aus Q/M_T in dünnwandigen Querschnitten</td>
</tr>
<tr>
<td>25</td>
<td>2D-Bernoulli - Voutenelement</td>
</tr>
<tr>
<td>26</td>
<td>4 Knoten Scheibenelement (EVZ, Axisym) (B-Formulierung)</td>
</tr>
<tr>
<td>27</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>4/8/9 Oberflächen-Lastelement</td>
</tr>
<tr>
<td>31</td>
<td>4 Knoten Plattenelement mit Stabilisierung</td>
</tr>
<tr>
<td>El. Nr.</td>
<td>Aufgabe</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>34</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>3D Solidschalenelement</td>
</tr>
<tr>
<td>46</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
</tr>
</tbody>
</table>
## 6.2 Element numbers

<table>
<thead>
<tr>
<th>Aufgabe</th>
<th>El. Nr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balken 2D-Timoshenko</td>
<td>2</td>
</tr>
<tr>
<td>Balken 2D-Bernoulli</td>
<td>3</td>
</tr>
<tr>
<td>Balken 2D-Bernoulli-Voute</td>
<td>25</td>
</tr>
<tr>
<td>Balken 3D-Timoshenko</td>
<td>15</td>
</tr>
<tr>
<td>Balken 3D-Timoshenko exzentris</td>
<td>11</td>
</tr>
<tr>
<td>Balken 3D-Bernoulli</td>
<td>4</td>
</tr>
<tr>
<td>Fachwerke 2D/3D</td>
<td>1</td>
</tr>
<tr>
<td>Flache Schale linear/nichtlinear</td>
<td>13</td>
</tr>
<tr>
<td>Knoten-Knoten-Kontakt 2D</td>
<td>14</td>
</tr>
<tr>
<td>Oberflächenlasten</td>
<td>30</td>
</tr>
<tr>
<td>Platten Bathe/Dvorkin</td>
<td>17</td>
</tr>
<tr>
<td>Platten DKT</td>
<td>19</td>
</tr>
<tr>
<td>Platten DKQ</td>
<td>7</td>
</tr>
<tr>
<td>Platten IDKQ</td>
<td>23</td>
</tr>
<tr>
<td>Platten SRI</td>
<td>18</td>
</tr>
<tr>
<td>Platten SRI+Stabilisierung</td>
<td>33</td>
</tr>
<tr>
<td>Punktfedern</td>
<td>10</td>
</tr>
<tr>
<td>Punktmassen</td>
<td>10</td>
</tr>
<tr>
<td>Scheiben (ESZ, EVZ, Axysym)</td>
<td>5</td>
</tr>
<tr>
<td>Scheiben (ESZ, Pian/Sumihara)</td>
<td>6</td>
</tr>
<tr>
<td>Scheiben</td>
<td>22</td>
</tr>
<tr>
<td>Scheiben (EVZ, Axysym) (B)</td>
<td>26</td>
</tr>
<tr>
<td>Schalen Achsensymmetrie</td>
<td>8</td>
</tr>
<tr>
<td>Schalen 3D</td>
<td>9</td>
</tr>
<tr>
<td>Seil 3D</td>
<td>16</td>
</tr>
<tr>
<td>Wärmeleitung</td>
<td>12</td>
</tr>
<tr>
<td>Grundwasserströmung</td>
<td>12</td>
</tr>
<tr>
<td>Trägerrost</td>
<td>20</td>
</tr>
<tr>
<td>Volumen, 3D-Körper</td>
<td>21</td>
</tr>
<tr>
<td>Schub aus $Q/M_T$ in dünnwandigen Querschnitten</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>26</td>
</tr>
<tr>
<td>Scheiben - B-Formulierung)</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>29</td>
</tr>
<tr>
<td>Oberflächen-Lasten</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>32</td>
</tr>
<tr>
<td>Platten URI mit Stabilisierung</td>
<td>33</td>
</tr>
<tr>
<td>Aufgabe</td>
<td>El. Nr.</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>3D-Solidschalen</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
</tr>
</tbody>
</table>
6.3 Short description of elements

<table>
<thead>
<tr>
<th>Element No.</th>
<th>Element type</th>
<th>Input Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>2D/3D Truss element</td>
<td>$E, A, \rho$</td>
</tr>
<tr>
<td>02</td>
<td>non-linear 2D Timoshenko-Beam Element</td>
<td>$E, G, A, I, \rho, \eta, \alpha_t, h, ityp$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$q_1, q_2, n_1, n_2$</td>
</tr>
<tr>
<td>03</td>
<td>2D Bernoulli beam element</td>
<td>$E, A, I, h, q_1, q_2, n_1, n_2, T2O, num$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\rho, c, b, \alpha_t, t_N, t_M$</td>
</tr>
<tr>
<td>04</td>
<td>3D Bernoulli beam element</td>
<td>$E, G, A, I_x, I_y, I_z, q_xL, q_yL, q_zL, q_xG, q_yG, q_zG, \alpha$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T2O, \rho$</td>
</tr>
<tr>
<td>05</td>
<td>Plane stress/strain element</td>
<td>$E, \nu, \rho, &lt;i, l, k&gt;$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$h, b_1, b_2, \alpha, T_0$ or $T_0 - T_1$</td>
</tr>
<tr>
<td>06</td>
<td>P/S plain stress/strain element</td>
<td>$E, \nu, \rho, h, i$</td>
</tr>
<tr>
<td>07</td>
<td>DKQ plate element</td>
<td>$E, \nu, h, q, \rho, c, f_{cd}, f_{yd}, d_x, d_y$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\alpha_t, \Delta T$</td>
</tr>
<tr>
<td>08</td>
<td>Axisymmetric shell element</td>
<td>$E, \nu, h, \rho, \alpha_t, \kappa_s$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$g, p_c, p_0, z_0, idir, s$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\alpha_t, t_b, t_t$</td>
</tr>
<tr>
<td>09</td>
<td>General shell element</td>
<td>$E, \nu, \rho, h$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_1, b_2, p, b_x, b_y, b_z, t_c, c$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>igeo,a1,a2,a3</td>
</tr>
<tr>
<td>10</td>
<td>Spring/mass/damping element</td>
<td>$K(i) i = 1, ndf$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$M(i) i = 1, ndf$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$D(i) i = 1, ndf$</td>
</tr>
<tr>
<td>11</td>
<td>3D eccentric Timoshenko beam element</td>
<td>$E, G, A, I_x, I_y, I_z, q_x, q_y, q_z, \alpha$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\text{lin}, \rho, I_{yz}, y_s, z_s, y_M, z_M, y_0, z_0, \text{scf}$</td>
</tr>
<tr>
<td>12</td>
<td>Heat transfer element</td>
<td>$\text{ityp, kat, ksym}$</td>
</tr>
<tr>
<td></td>
<td>(and other steady state problems)</td>
<td>$K_1, K_2, \text{angl1} - x, \rho \cdot h, \rho \cdot c, \rho \cdot g, \text{elev}$</td>
</tr>
<tr>
<td>13</td>
<td>shallow shell element</td>
<td>$E, \nu, h, q, \rho, \text{lin}$</td>
</tr>
<tr>
<td>14</td>
<td>node to node contact element</td>
<td>$\text{idof, pen, tol}$</td>
</tr>
<tr>
<td>15</td>
<td>2–5 node non-linear 3D Timoshenko-Beam Element</td>
<td>$E, G, A, I_y, I_z, I_{yz}, I_T, y_s, z_s, y_m, z_m, \alpha$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\text{iGeo, iPl, iscf, \rho}$</td>
</tr>
<tr>
<td>Element No.</td>
<td>Element type</td>
<td>Input Data</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>16</td>
<td>3D cable element</td>
<td>$E, A, \gamma, \alpha_t, s_0/\ell_0, S_0, q_x, q_y, q_z, \Delta t$</td>
</tr>
<tr>
<td>17</td>
<td>Bathe/Dvorkin plate element</td>
<td>$E, \nu, h, q, \rho, SCF$</td>
</tr>
<tr>
<td>18</td>
<td>SRI plate element</td>
<td>$E, \nu, h, q, \rho, nb, ns, SCF$</td>
</tr>
<tr>
<td>19</td>
<td>DKT plate element</td>
<td>$E, \nu, h, q$</td>
</tr>
<tr>
<td>20</td>
<td>2D Bernoulli/St.Venant Grid beam element</td>
<td>$E, G, I_t, I_y, q_1, q_2$</td>
</tr>
<tr>
<td>21</td>
<td>3D solid element</td>
<td>$matn, lin, isym, istr, ngpk, ngps, q_x, q_y, q_z, \Delta T, \alpha_t, \rho$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>matn=1: $E, \nu$</td>
</tr>
<tr>
<td>22</td>
<td>Plane stress element</td>
<td>$E, \nu, h, \alpha_T, \Delta T, lin, Y_0, cp$</td>
</tr>
<tr>
<td>23</td>
<td>Improved DKQ plate element</td>
<td>$E, \nu, h, q$</td>
</tr>
<tr>
<td>24</td>
<td>Shear stresses from forces/torsional moments</td>
<td>$E, G, b, Q_y, Q_z, ns, nsp, ityp, M_{T0}, ic$</td>
</tr>
<tr>
<td>25</td>
<td>2D-Bernoulli-Hauch Element</td>
<td>$Q - No, E, \ell_1, \ell_2, \ell_3, \ell_4, \ell_5, n, q$</td>
</tr>
<tr>
<td>26</td>
<td>B-plain strain/axisym element</td>
<td>$imat, ityp, II, \rho, K, G$</td>
</tr>
<tr>
<td>30</td>
<td>Surface load element</td>
<td>–</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>SRI plate element with stabilization</td>
<td>$E, \nu, h, q, \rho, r_w, r_\beta$</td>
</tr>
<tr>
<td>Element No.</td>
<td>Element type</td>
<td>Input Data</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>3D solid shell element EAS</td>
<td><code>matn, nlay, lin, ielas1, ibd, ibs, ielas2, h, ngp, scf</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>q_x, q_y, q_z, ΔT, α_t, ρ</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>E, ν (for matn=1)</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>for i=1,nlay: phi_i,h_i</code></td>
</tr>
<tr>
<td>46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6.4 Detailed description of elements

6.4.1 ELMT01

$E, A, \rho$

- Theory

ELMT01 is a general 2D/3D truss element. \([\text{ndm}=2/3, \text{ndf}=2/3, \text{nel}=2]\)

- Material input data

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$</td>
<td>$F/L^2$</td>
<td>Young’s modulus</td>
</tr>
<tr>
<td>$A$</td>
<td>$L^2$</td>
<td>Area</td>
</tr>
<tr>
<td>$\rho$</td>
<td>$F/L^3/[L/T^2]$</td>
<td>density $\rho = \gamma/g$</td>
</tr>
</tbody>
</table>

- Displacements and normal forces

- Output of results

Normal forces, strains and stresses are calculated for each element in the local axial direction.

- Plot of results

Results are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity plotted</td>
<td>$N [F]$</td>
<td>$\varepsilon [-]$</td>
<td>$\sigma [F/L^2]$</td>
</tr>
</tbody>
</table>
### Theory

**ELMT02** is a 2–node 2D Timoshenko–Beam Element with linear shape functions. Thus more elements are necessary for a correct solution. Geometrical nonlinear calculations are possible for different nonlinear theories. For shear correction $\kappa = 5/6$ together with $GA_s = \frac{\kappa \cdot GA}{(1 + \kappa \cdot \ell^2/12 \cdot GA/EI)}$ is used.

[ndm=2, ndf=3, nel=2]  

<table>
<thead>
<tr>
<th>(E)</th>
<th>(F/L^2)</th>
<th>Elastic modulus</th>
</tr>
</thead>
<tbody>
<tr>
<td>(G)</td>
<td>(F/L^2)</td>
<td>Shear modulus</td>
</tr>
<tr>
<td>(A)</td>
<td>(L^2)</td>
<td>Area</td>
</tr>
<tr>
<td>(I)</td>
<td>(L^4)</td>
<td>Moment of Inertia</td>
</tr>
<tr>
<td>(\rho)</td>
<td>(F/L^3)</td>
<td>(L/T^2)</td>
</tr>
<tr>
<td>(\eta)</td>
<td>(F/L^3)</td>
<td>(L/T)</td>
</tr>
<tr>
<td>(\alpha_t)</td>
<td>(1/K)</td>
<td>coefficient of thermal expansion</td>
</tr>
<tr>
<td>(h)</td>
<td>(L)</td>
<td>thickness for temperature loads</td>
</tr>
</tbody>
</table>

**Material input data**

- **ityp**
  - 1 = finite rotations (Green)(default)
  - 2 = finite rotations (Reissner)
  - 3 = moderate rotations (consistent, Green)
  - 4 = moderate rotations (simplified, Green)
  - 5 = linear

| \(q_1\) | \(F/L\) | load in local z-dir. at left node |
| \(q_2\) | \(F/L\) | load in local z-dir. at right node |
| \(n_1\) | \(F/L\) | load in local x-dir. at left node |
| \(n_2\) | \(F/L\) | load in local x-dir. at right node |

**Loads via macro qloa.**

| \(ma\) | [-] | Material number in Inputfile |
| \(q_1\) | \(F/L\) | load in local z-dir. at left node |
| \(q_2\) | \(F/L\) | load in local z-dir. at right node |
| \(n_1\) | \(F/L\) | load in local x-dir. at left node |
| \(n_2\) | \(F/L\) | load in local x-dir. at right node |
| \(ifol\) | [-] | follower load 0/1, only for q |
| \(T_u\) | \(K\) | Temperature at bottom |
| \(T_o\) | \(K\) | Temperature at top |

**Displacements and stress resultants**
• **Output of stress resultants**

Stress resultants (N–Axial Force, Q–Shear Force, M–Bending Moment) are calculated at left and right node of the element in local directions for the reference configuration (ityp=5) or the current configuration (ityp≤4).

• **Plot of stress resultants**

The above defined stress resultants are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress resultant</td>
<td>N [F]</td>
<td>Q [F]</td>
<td>M [FL]</td>
</tr>
</tbody>
</table>
6.4.3 ELMT03

$E, A, I, h, q_1, q_2, n_1, n_2, T2O, num$

$\rho, c, b, \alpha_t, t_t, t_b$

- **Theory**

**ELMT03** is a 2–node 2D Bernoulli–Beam Element with Hermitean shape functions (cubic). Thus at least one element leads for a part of a structure in most cases to the correct solution. Symmetry with respect to local z-axis is assumed. \([\text{ndm}=2, \text{ndf}=3, \text{nel}=2]\)

- **Material input data**

<table>
<thead>
<tr>
<th>Record 1</th>
<th>Record 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E[F/L^2]$</td>
<td>Elastic modulus</td>
</tr>
<tr>
<td>$h[L]$</td>
<td>thickness</td>
</tr>
<tr>
<td>$q_1[F/L]$</td>
<td>load in local z-dir / left node</td>
</tr>
<tr>
<td>$q_2[F/L]$</td>
<td>load in local z-dir / right node</td>
</tr>
<tr>
<td>$n_1[F/L]$</td>
<td>load in local x-dir / left node</td>
</tr>
<tr>
<td>$n_2[F/L]$</td>
<td>load in local x-dir / right node</td>
</tr>
<tr>
<td>$T2O$</td>
<td>II.Order theory(0=F,1=T,2=T(Resi))</td>
</tr>
<tr>
<td>$\text{num}$</td>
<td>Plotnumbers (0=F,1=T)</td>
</tr>
</tbody>
</table>

| $ma$ | Material number in Inputfile |
| $q_1$ | $F/L$ | load in local z-dir. at left node |
| $q_2$ | $F/L$ | load in local z-dir. at right node |
| $n_1$ | $F/L$ | load in local x-dir. at right node |
| $n_2$ | $F/L$ | load in local x-dir. at right node |

- **Loads** via macro $qloa$. 

- **Displacements and stress resultants**
• Output of stress resultants
Stress resultants (N–Axial Force, Q–Shear Force, M–Bending Moment and Foundation Pressure) are calculated at left and right node of the element in local directions due to the standard sign convention.

• Plot of stress resultants
The above defined stress resultants are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress resultant</td>
<td>N [F]</td>
<td>Q [F]</td>
<td>M [F · L]</td>
<td>P(c) [F/L²]</td>
</tr>
</tbody>
</table>

• Remark on II. Order Theory

\#T20

= 1: Full iteration
\[ (K_e + K_g)\Delta v^i = P - (K_e + K_g)v^i \]

= 2: Iteration only on right hand side
\[ K_e\Delta v^i = P - (K_e + K_g)v^i \]

# iteration until convergence, e.g. 10 steps:

```
loop,,10
\text{tang},,1
next
```

# results: disp,all stre,all reac,all

• For the elastic foundation two models are implemented:

\[ c > 0 \text{ el. found. always (tension and compression)} \]
\[ c < 0 \text{ el. found. only for compression (Solution only available after equilibrium iteration!)} \]

```
loop,,10
\text{tang},,1
next
```
6.4.4 ELMT04

---

**Theory**

ELMT04 is a 2(+1)–node 3D Bernoulli–Beam Element. The element bases on a Bernoulli-theory for the bending terms whereas for the torsional terms a St.–Venant theory is used. The local axis are chosen as follows: $x_L$ is the axial axis through the center of gravity, $y_L$ and $z_L$ are the principal axes of the beam. [ndm=3, ndf=6, nel=2(+1)]

The base vectors can be defined in two ways:

- **Element with 3 nodes:** introduce third node for each element (generation of elements via macro `el3b`)
  
  $e_{xL}$ in element direction
  
  $e_{yL} = [x_3 - x_1] \times e_{xL}$
  
  $e_{zL} = e_{xL} \times e_{yL}$

- **Element with 2 nodes:** global z-axis is used for definition of the base vectors, macro `elem` can be used.
  
  $e_{xL}$ in element direction
  
  $e_{yL} = e_{xL} \times e_{zG}$ with $e_{zG} = [0, 0, 1]$  
  
  $e_{zL} = e_{xL} \times e_{yL}$

  For the special case that the element direction is $\pm$ the global z-axis $e_{yL}$ is defined as $e_{yL} = [0, -1, 0]$.

A rotation of the local $y_L$ and $z_L$-axis can be introduced by the angle $\alpha$, which starts positive from the $y_L$-axis in direction to the $z_L$-axis. Input of $\alpha$ in degree! Thus un-symmetric cross sections can be used.

---

**Material input data**

<table>
<thead>
<tr>
<th>Record 1</th>
<th>Record 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E[F/L^2]$</td>
<td>Elastic modulus</td>
</tr>
<tr>
<td>$G[F/L^2]$</td>
<td>Shear modulus</td>
</tr>
<tr>
<td>$A[L^2]$</td>
<td>Area</td>
</tr>
<tr>
<td>$I_x[L^4]$</td>
<td>Moment of Inertia / local x-axis</td>
</tr>
<tr>
<td>$I_y[L^4]$</td>
<td>Moment of Inertia / local y-axis</td>
</tr>
<tr>
<td>$I_z[L^4]$</td>
<td>Moment of Inertia / local y-axis</td>
</tr>
<tr>
<td>$q_{xL}[F/L]$</td>
<td>const. load / local x-direction</td>
</tr>
<tr>
<td>$q_{yL}[F/L]$</td>
<td>const. load / local y-direction</td>
</tr>
<tr>
<td>$q_{zL}[F/L]$</td>
<td>const. load / local z-direction</td>
</tr>
<tr>
<td>$q_{xG}[F/L]$</td>
<td>const. load / global x-direction</td>
</tr>
<tr>
<td>$q_{yG}[F/L]$</td>
<td>const. load / global y-direction</td>
</tr>
<tr>
<td>$q_{zG}[F/L]$</td>
<td>const. load / global z-direction</td>
</tr>
<tr>
<td>$\alpha[DEG.]$</td>
<td>Rot. angle for local coor.system</td>
</tr>
</tbody>
</table>
**Loads** via macro \texttt{qloa}.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ma</td>
<td>Material number in Inputfile</td>
</tr>
<tr>
<td>( q_{xL} )</td>
<td>const. load / local ( x )-direction</td>
</tr>
<tr>
<td>( q_{yL} )</td>
<td>const. load / local ( y )-direction</td>
</tr>
<tr>
<td>( q_{zL} )</td>
<td>const. load / local ( z )-direction</td>
</tr>
<tr>
<td>( q_{xG} )</td>
<td>const. load / global ( x )-direction</td>
</tr>
<tr>
<td>( q_{yG} )</td>
<td>const. load / global ( y )-direction</td>
</tr>
<tr>
<td>( q_{zG} )</td>
<td>const. load / global ( z )-direction</td>
</tr>
</tbody>
</table>

**Displacements and base vectors for element**

\[
e_{xL} = (x_3 - x_1) e_{xL} \\
e_{zL} = e_{xL} \times e_{yL}
\]

**Output of stress resultants**

Stress resultants (\( N_x \)-Axial Force, \( Q_y/Q_{dy} \)-Shear Force, \( Q_z/Q_{dz} \)= \( R_z \)-Shear Force, \( M_x \)-Torsional Moment, \( M_y \)-Bending Moment, \( M_z \)-Bending Moment) are calculated at left and right node of the element in local directions.

**Plot of stress resultants [FORC]**

The above defined stress resultants are plotted due to following numbers.

<table>
<thead>
<tr>
<th>Force number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress resultant</td>
<td>( N_x ) [\textit{F}]</td>
<td>( Q_y/Q_{dy} ) [\textit{F}]</td>
<td>( Q_z/Q_{dz} ) [\textit{F}]</td>
<td>( M_x ) [\textit{FL}]</td>
<td>( M_y ) [\textit{FL}]</td>
<td>( M_z ) [\textit{FL}]</td>
</tr>
</tbody>
</table>

**Remark:** The local base vectors can be visualized for control of input data with \texttt{Forc,7}.

**Remark:** Stress resultants are plotted in the local 1-3 direction. The plot plane can be modified with \texttt{Forc,\pm12} or \texttt{\pm13} (default 13).

**Remark on II. Order Theory**

see remark for \texttt{ELMT03}
6.4.5 ELMT05

\[ E, \nu, \rho, <i>, <l>, <k> \]
\[ h, b_1, b_2, \alpha, T_0 \text{ or } T_0 - T_1 \]

- **Theory**

ELMT05 is a general 3–9 node plane stress, plane strain, and (torsionless) axisymmetric element. The 3–node version is a triangle, the 4–node is a quadrilateral with linear shape functions, whereas the 9–node element has quadratical shape functions. The 5–7 node elements have some midside nodes. The 8–node element is of Serendipity–type. [ndm=2, ndf=2, nel=3-9]

- **Material input data**

| \( E[F/L^2] \) | Young’s modulus |
| \( \nu[-] \) | Poisson ratio |
| \( \rho[F/L^3]/[L/T^2] \) | density \( \rho = \gamma/g \) |
| \( <i> \) (def.=1) | 1=plane stress(ESZ), 2=plane strain(EVZ), 3=axisymmetric calculation(ROTS) |
| \( <l> \) (def.=2) | number of Gauss quadrature points/direction for computing the FE arrays (e.g., stiffness) |
| \( <k> \) (def.=1) | number of Gauss quadrature points/direction for stress outputs |
| \( h[L] \) | thickness of plane stress slice (set to 1.0 for plane strain, axisymmetric) |
| \( b_1[F/L^3] \) | body force value (constant) for 1(x)–direction |
| \( b_2[F/L^3] \) | body force value (constant) for 2(y)–direction |
| \( \alpha[1/K] \) | coefficient of linear thermal expansion |
| \( T_0[K] \) | base temperature for this material or |
| \( T_0 - T_1[K] \) | temperature difference for this material |

- **Loads** via macro `qloa`.

| \( ma \) | Material number in Inputfile |
| \( b_1 F/L^3 \) | body force value (constant) for 1(x)–direction |
| \( b_2 F/L^3 \) | body force value (constant) for 2(y)–direction |

- **Temperature**

If the input is the base temperature \( T_0 \), then macros `temp` or `btem` have to be used for input of actual temperature \( T_1 \). Thus a complicated temperature distribution may be possible. In case of an input \( T_0 - T_1 \) the system is loaded by a constant temperature difference. Thus the use of `temp` or `btem` is not necessary. Note that \( T_0 \) is the base temperature and \( T_1 \) is the actual temperature.
**Displacements (u,v) and stresses**

The coordinate system is (x,y,z) for ESZ, EVZ. In the axisymmetric case (ROTS) the coordinate system is (r,z,\(\varphi\))=(x,y,\(\varphi\)).

**Output of stresses and strains**

Stresses and strains are calculated at each quadrature point in global directions x, y, and z. The stresses \(\sigma_1\) and \(\sigma_2\) are the principal stresses with the angle \(\varphi_1\) between x–stress and x–coordinate direction (in degrees).

**Plot of stresses**

Stresses and strains are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress ESZ</td>
<td>(\sigma_{xx})</td>
<td>(\tau_{xy})</td>
<td>(\sigma_{yy})</td>
<td>(\sigma_z)</td>
<td>(\sigma_1)</td>
<td>(\sigma_2)</td>
<td>(\varphi_1)</td>
<td>(\varepsilon_{xx})</td>
<td>(\gamma_{xy})</td>
<td>(\varepsilon_{yy})</td>
<td>(\varepsilon_{zz})</td>
</tr>
<tr>
<td>Stress EVZ</td>
<td>(\sigma_{xx})</td>
<td>(\tau_{xy})</td>
<td>(\sigma_{yy})</td>
<td>(\sigma_{zz})</td>
<td>(\sigma_1)</td>
<td>(\sigma_2)</td>
<td>(\varphi_1)</td>
<td>(\varepsilon_{xx})</td>
<td>(\gamma_{xy})</td>
<td>(\varepsilon_{yy})</td>
<td>(\varepsilon_{zz})</td>
</tr>
<tr>
<td>Stress ROTS</td>
<td>(\sigma_{rr})</td>
<td>(\tau_{rz})</td>
<td>(\sigma_{zz})</td>
<td>(\sigma_{\varphi\varphi})</td>
<td>(\sigma_1)</td>
<td>(\sigma_2)</td>
<td>(\varphi_1)</td>
<td>(\varepsilon_{rr})</td>
<td>(\gamma_{rz})</td>
<td>(\varepsilon_{zz})</td>
<td>(\varepsilon_{\varphi\varphi})</td>
</tr>
</tbody>
</table>

Element-wise constant stresses can be plotted with macro **str1**.

Plot-results for the 8–node serendipity–element could not be used due negative weighting functions.

**Remarks**

Loads (in **load,eloa**, ...) must be multiplied by the term \(2\pi r\) in case of axisymmetric problems!

Especially for inplane bending problems in case of plane stress/strain **ELMT06** should be used which leads to better results.

The element routine will compute any combination of nodes, up to 9, provided all vertex nodes are specified. Triangles may be specified by either giving the same node number for contiguous vertices, or for a 3–node element, by specifying only the first 3 nodes (i.e., node 4 has a zero value).
6.4.6 ELMT06

$E, \nu, \rho, h, i$

- **Theory**

ELMT06 is a 4 node plane stress, plane strain element. It is called PIAN/SUMIHARA element and is based on a mixed formulation. This element is superior to ELMT05 and should be used instead of ELMT05, especially for inplane bending problems. [ndm=2, ndf=2, nel=4]

- **Material input data**

| $E[F/L^2]$ | Young’s modulus |
| $\nu[-]$ | Poisson ratio |
| $\rho[F/L^3]/[L/T^2]$ | density $\rho = \gamma/g$ |
| $h[L]$ | thickness (set to 1.0 for plane strain) |
| $i$ | 1=plane stress, 2=plane strain |

- **Displacements (u,v) and stresses**

- **Output of stresses**

Stresses are calculated at the element center in global directions x and y. The stresses $\sigma_1$ and $\sigma_2$ are the principal stresses with the angle $\varphi_1$ between 1–stress and x–coordinate direction (in degrees).

- **Plot of stresses**

Stresses are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress plotted</td>
<td>$\sigma_{xx}$</td>
<td>$\sigma_{xy}$</td>
<td>$\sigma_{yy}$</td>
<td>$\sigma_{zz}$</td>
<td>$\sigma_1$</td>
<td>$\sigma_2$</td>
<td>$\varphi_1$</td>
</tr>
</tbody>
</table>

Element-wise constant stresses can be plotted with macro `str1`. 
6.4.7 ELMT07

\[ E, \nu, h, q, \rho, c, f_{cd}, f_{yd}, d_x, d_y, \alpha_t, \Delta T \]

- **Theory**
  ELMT07 is a general 4-node thin plate bending element based on the Discrete Kirchhoff Theory. [ndm=3, ndf=3, nel=4]

- **Material input data**

<table>
<thead>
<tr>
<th>Record 1</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E[F/L^2] )</td>
<td>Modulus of elasticity</td>
</tr>
<tr>
<td>( \nu[-] )</td>
<td>Poisson ratio</td>
</tr>
<tr>
<td>( h[L] )</td>
<td>Plate thickness</td>
</tr>
<tr>
<td>( q[F/L^2] )</td>
<td>Uniform normal load in z-dir.</td>
</tr>
<tr>
<td>( \rho[F/L^3]/[L/T^2] )</td>
<td>density: ( \rho = \gamma/g )</td>
</tr>
<tr>
<td>( c[F/L^3] )</td>
<td>Elastic foundation in z-dir.</td>
</tr>
<tr>
<td>( f_{cd}[F/L^2] )</td>
<td>Strength concrete</td>
</tr>
<tr>
<td>( f_{yd}[F/L^2] )</td>
<td>Strength steel</td>
</tr>
<tr>
<td>( d_x[L] )</td>
<td>Thickness until steel in x-dir.</td>
</tr>
<tr>
<td>( d_y[L] )</td>
<td>Thickness until steel in y-dir.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Record 2</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_t[1/K] )</td>
<td>Coefficient of thermal expansion</td>
</tr>
<tr>
<td>( \Delta T[K] )</td>
<td>Temperature difference (( T_{top} - T_{bottom} ))</td>
</tr>
</tbody>
</table>

- For the elastic foundation two models are implemented:
  - \( c > 0 \) el. found. always (tension and compression)
  - \( c < 0 \) el. found. only for compression (Solution only available after equilibrium iteration!)

- **Loads** via macro qloa.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( ma )</td>
<td>Material number in Inputfile</td>
</tr>
<tr>
<td>( q )</td>
<td>[( F/L^2 )] uniform normal load in z-direction</td>
</tr>
<tr>
<td>( \Delta T )</td>
<td>[( K )] Temperature difference (( T_{top} - T_{bottom} ))</td>
</tr>
</tbody>
</table>

- Definition of stress resultants
  \[ m_{xx} = - \int \sigma_{xx} z \, dz, \quad m_{yy} = - \int \sigma_{yy} z \, dz, \quad m_{xy} = - \int \sigma_{xy} z \, dz = m_{yx} \]
• Displacements \((w, \varphi_x, \varphi_y)\) and moments per length

\[ m_{xx}, m_{xy}, m_{yy} \]

• Output of moments

The moments per unit length are calculated at each center of element in global directions \(x\) and \(y\). The moments \(m_1\) and \(m_2\) are the principal moments with the angle \(\varphi_1\) between 1–moment and \(x\)-coordinate direction (in degrees). Furthermore the foundation pressure is calculated at each center (Positive Pressure is in -z direction).

The reinforcements are calculated based on the general design diagram due to DIN 1045-1 (new). Note that ultimate(!) loads are input and that minimum values of reinforcements and constructive reinforcements have to be added.

• Plot of moments

The moments per unit length, the foundation pressure and the reinforcements are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>Type</th>
<th>value plotted</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Moment</td>
<td>(m_{xx})</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>2</td>
<td>Moment</td>
<td>(m_{xy})</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>3</td>
<td>Moment</td>
<td>(m_{yy})</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>4</td>
<td>Pressure</td>
<td>(\text{Press}(c))</td>
<td>([F/L^2])</td>
</tr>
<tr>
<td>5</td>
<td>Moment</td>
<td>(m_1)</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>6</td>
<td>Moment</td>
<td>(m_2)</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>7</td>
<td>Angle</td>
<td>(\varphi_1)</td>
<td>([\text{rad}])</td>
</tr>
<tr>
<td>8</td>
<td>Shear force</td>
<td>(q_x)</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>9</td>
<td>Shear force</td>
<td>(q_y)</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>10</td>
<td>Reinforcement</td>
<td>(as_x \text{ bot})</td>
<td>([L^2/L])</td>
</tr>
<tr>
<td>11</td>
<td>Reinforcement</td>
<td>(as_x \text{ top})</td>
<td>([L^2/L])</td>
</tr>
<tr>
<td>12</td>
<td>Reinforcement</td>
<td>(as_y \text{ bot})</td>
<td>([L^2/L])</td>
</tr>
<tr>
<td>13</td>
<td>Reinforcement</td>
<td>(as_y \text{ top})</td>
<td>([L^2/L])</td>
</tr>
</tbody>
</table>

Element-wise constant values can be plotted with macro \texttt{str1}. 
6.4.8 ELMT08

\[ E, \nu, h \]
\[ g, p_c, p_0, z_0, idir, s \]
\[ \alpha_t, t_b, t_t \]

- **Theory**
  ELMT08 is a shear–elastic plane 2–node shell–element for axisymmetric shells with axisymmetric loading. Due to linear shape functions C⁰-continuity is achieved. The geometry is described with a two-dimensional mesh. [ndm=2, ndf=3, nel=2]

- **Material input data**

<table>
<thead>
<tr>
<th>Record 1</th>
<th>Youngs modulus</th>
</tr>
</thead>
<tbody>
<tr>
<td>E[F/L²]</td>
<td></td>
</tr>
<tr>
<td>ν[−]</td>
<td>Poison’s ratio</td>
</tr>
<tr>
<td>h[L]</td>
<td>thickness</td>
</tr>
<tr>
<td>ρ[F/L³]/[L/T²]</td>
<td>density ( \rho = \gamma/g )</td>
</tr>
<tr>
<td>c_b[F/L³]</td>
<td>elastic foundation</td>
</tr>
<tr>
<td>SCF[−]</td>
<td>shear correction factor, see Elmt18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Record 2</th>
<th>deadload, positive in global negative 2–direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>g[F/L³]</td>
<td></td>
</tr>
<tr>
<td>p_c[F/L²]</td>
<td>const. pressure load, positive in positive local 2–direction</td>
</tr>
<tr>
<td>p_0[F/L³]</td>
<td>slope value for linear pressure</td>
</tr>
<tr>
<td>z_0[L]</td>
<td>coordinate of zero pressure, only for linear pressure loading</td>
</tr>
<tr>
<td>idir</td>
<td>direction of linear pressure loading, 1=r, 2=z (Default=2)</td>
</tr>
<tr>
<td>s[F/L²]</td>
<td>snowload, positive in global negative 2–direction</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Record 3</th>
<th>coefficient of thermal expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>α_t[1/K]</td>
<td></td>
</tr>
<tr>
<td>t_b[K]</td>
<td>temperature at shell bottom</td>
</tr>
<tr>
<td>t_t[K]</td>
<td>temperature at shell top</td>
</tr>
</tbody>
</table>
Displacements and stress resultants

\[ \begin{align*}
  2 = z_1 & \quad 1 = r \\
  2 \quad 1 & \quad p_0 \\
  n_s & \quad m_s \\
  n_t & \quad m_t \\
  q_s & \quad q_s
\end{align*} \]

Output of stress resultants

Stress resultants are calculated at the center of the element:

<table>
<thead>
<tr>
<th>$n_s$</th>
<th>Axial Force direction s</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_t$</td>
<td>Axial Force direction t</td>
</tr>
<tr>
<td>$m_s$</td>
<td>Bending Moment direction s</td>
</tr>
<tr>
<td>$m_t$</td>
<td>Bending Moment direction t</td>
</tr>
<tr>
<td>$q_s$</td>
<td>Shear Force direction s</td>
</tr>
</tbody>
</table>

Plot of stress resultants

The above defined stress resultants are plotted due to following numbers

<table>
<thead>
<tr>
<th>Force number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress resultant</td>
<td>$n_s$</td>
<td>$n_t$</td>
<td>$m_s$</td>
<td>$m_t$</td>
<td>$q_s$</td>
</tr>
</tbody>
</table>

Remark: Input of loads is also possible with the macros load and eloa. Here, the factor $2\pi r$ has to be added for all terms! Furthermore eloa can be used for loads $q$, if the term $q \cdot 2\pi r$ fits in the schemes of load cases ($r$ may be not constant!).
6.4.9 ELMT09

- Theory

ELMT09 is a 4-node general shell-Element. The element is based on a DKQ-theory for the bending terms with modifications due to the element warping whereas the membrane part is modified due to the use of the rotation around the local z-axis which is called drilling degree of freedom. [ndm=3, ndf=6, nel=4]

- Input data

# Record 1: Material input data

| $E [F/L^2]$ | Youngs modulus |
| $\nu [-]$ | Poisson’s ratio |
| $\rho [F/L^2]/[L/T^2]$ | density $\rho = \gamma/g$ |
| $h [L]$ | thickness |

# Record 2: Load data

| $b_1 [F/L^2]$ | uniform loading in 1-direction |
| $b_2 [F/L^2]$ | uniform loading in 2-direction |
| $p [F/L^2]$ | uniform loading in 3-direction |
| $b_x [F/L^2]$ | gravity loading in x-direction |
| $b_y [F/L^2]$ | gravity loading in y-direction |
| $b_z [F/L^2]$ | gravity loading in z-direction |
| $t_c$ | load type (0=lumped; 1=consistent) |
| $c [F/L^2]$ | elastic found. constant in local 3-dir. |

# Record 3: Local base vectors

| igeo | 1 | 2 | 3-17 |
| a1 | $t_{qz} [L]$ | see macro base |
| a2 | $t_{qy} [L]$ |
| a3 | $t_{qx} [L]$ |

- Loads via macro qloa.

| ma | Material number in Inputfile |
| $b_1 [F/L^2]$ | uniform loading in 1-direction |
| $b_2 [F/L^2]$ | uniform loading in 2-direction |
| $b_3 [F/L^2]$ | uniform loading in 3-direction |
| $b_x [F/L^2]$ | gravity loading in x-direction |
| $b_y [F/L^2]$ | gravity loading in y-direction |
| $b_z [F/L^2]$ | gravity loading in z-direction |
• Base Systems

The local coordinate system is assumed in the center of the element.

# igeo = 0,1 (default option)

\[ t_3 \] is the normal vector. The local base vector \( t_1 \) bisects the line between node 2 and 3 of the element, the base vector \( t_2 \) is calculated from \( t_2 = t_3 \times t_1 \).

# igeo = 2

Input is a vector \( t_q = [t_qx, t_qy, t_qz]^T \). Again \( t_3 \) is the normal vector, defined like for igeo=1. It follows: \( t_2 = t_3 \times t_q \) and \( t_1 = t_2 \times t_3 \).

# igeo = 3-17

Other base systems can be defined by using data from the macro base.

Remarks

# For regular meshes igeo=0,1 leads to correct results, whereas for distorted meshes each element has different directions. Thus stress resultants are calculated in different directions but added for e.g. stre,1. This is nonsense!

# It is strongly recommended to control the orientation of the local base system using the macro forc,1

# If the element is used in combination with adaptivity the input of \( t_q \) is mandatory!!

• Elastic foundation

For the elastic foundation two models are implemented: \( c > 0 \) el. found. always (tension and compression) \( c < 0 \) el. found. only for compression (Solution only available after equilibrium iteration!)

• Displacements and stress resultants
Output of stress/strain resultants of shell (stre,...) (at center/midsurface of element)

- Axial Force direction 1
- Bending Moment direction 1
- Axial Force direction 12
- Bending Moment direction 12
- Principal Force 1
- Principal Moment 1
- Principal Force 2
- Principal Moment 2
- Principal angle for forces
- Principal angle for moments
- Membrane strain in direction 1
- Curvature in direction 1
- Membrane strain in direction 12
- Curvature in direction 12
- Membrane strain in direction 2
- Curvature in direction 2
- Pressure in direction 3 (+p in -3dir.)

Plot of stress resultants (stre,...)

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress resultant</td>
<td>n_{11}</td>
<td>n_{12}</td>
<td>n_{22}</td>
<td>n_{1}</td>
<td>n_{2}</td>
<td>m_{11}</td>
<td>m_{12}</td>
<td>m_{22}</td>
<td>m_{1}</td>
<td>m_{2}</td>
<td>p</td>
</tr>
</tbody>
</table>

Output of stresses/strains (stre,lay,1,n,-1) (at center/top(t)+bottom(b) of element)

<table>
<thead>
<tr>
<th>Values at top (t)</th>
<th>Values at bottom (b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>σ_{11}(t)</td>
<td>σ_{11}(b)</td>
</tr>
<tr>
<td>σ_{12}(t)</td>
<td>σ_{12}(b)</td>
</tr>
<tr>
<td>σ_{22}(t)</td>
<td>σ_{22}(b)</td>
</tr>
<tr>
<td>σ_{1}(t)</td>
<td>σ_{1}(b)</td>
</tr>
<tr>
<td>σ_{2}(t)</td>
<td>σ_{2}(b)</td>
</tr>
<tr>
<td>α_{σ}(t)</td>
<td>α_{σ}(b)</td>
</tr>
<tr>
<td>ε_{11}(t)</td>
<td>ε_{11}(b)</td>
</tr>
<tr>
<td>ε_{12}(t)</td>
<td>ε_{12}(b)</td>
</tr>
<tr>
<td>ε_{22}(t)</td>
<td>ε_{22}(b)</td>
</tr>
<tr>
<td>p</td>
<td></td>
</tr>
</tbody>
</table>

Plot of stresses (stre,i,-1)

<table>
<thead>
<tr>
<th>Values at top (t)</th>
<th>Values at bottom (b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress number</td>
<td>1</td>
</tr>
<tr>
<td>Stress</td>
<td>σ_{11}</td>
</tr>
</tbody>
</table>

Plot of element-wise constant stresses/stress resultants

Use macro str1.i.
6.4.10 ELMT10

\[ K(i), i = 1, \text{ndf} \]
\[ M(i), i = 1, \text{ndf} \]
\[ D(i), i = 1, \text{ndf} \]

- **Theory**

ELMT10 is a point stiffness/mass/damping element for adding point (diagonal) entries to the stiffness/mass/damping matrix. An element with one node has to be defined in the Input-file. \([\text{ndm}=], \text{ndf}=1-6, \text{nel}=1]\)

Alternatively, two nodes can be defined and the stiffness/mass/damping terms are set between these two nodes for the defined dofs. Thus a coupling of elements is possible. Other dofs between these nodes may be coupled with `link`. \([\text{ndm}=1-3, \text{ndf}=1-6, \text{nel}=2]\)

- **Material input data**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(K(1)[F/L])</td>
<td>Stiffness value to be added to 1-dof</td>
</tr>
<tr>
<td>(K(2)[F/L])</td>
<td>Stiffness value to be added to 2-dof</td>
</tr>
<tr>
<td></td>
<td>... up to ndf</td>
</tr>
<tr>
<td>(M(1)[F]/[L/T^2])</td>
<td>Mass value to be added to 1-dof</td>
</tr>
<tr>
<td>(M(2)[F]/[L/T^2])</td>
<td>Mass value to be added to 2-dof</td>
</tr>
<tr>
<td></td>
<td>... up to ndf</td>
</tr>
<tr>
<td>(D(1)[F]/[L/T])</td>
<td>Damping value to be added to 1-dof</td>
</tr>
<tr>
<td>(D(2)[F]/[L/T])</td>
<td>Damping value to be added to 2-dof</td>
</tr>
<tr>
<td></td>
<td>... up to ndf</td>
</tr>
</tbody>
</table>

- **Material model for elastic foundation**:

  \(K(i) > 0\) el. found. always (tension and compression)
  
  \(K(i) < 0\) el. found. only for compression (Solution only available after equilibrium iteration!)

- **Output of internal forces/reactions**

Internal forces (=reactions) for each degree of freedom are calculated and printed via `stre`.

Reactions at nodes are calculated and printed via `reac` for two node version.

- **Plot of internal forces/reactions**

Internal forces are plotted via `forc` for two node version.

Reactions at nodes are plotted via `reac` for two node version.
6.4.11 ELMT11

\[ E, G, A, I_T, I_y, I_z, q_x, q_y, q_z, \alpha \]
\[ \text{lin}, \rho, I_{yz}, z_S, y_M, z_M, y_O, z_O, \text{scf} \]

- **Theory**

ELMT11 is a geometrically nonlinear 2(+1)-node 3D Timoshenko–Beam Element. The element bases on a Timoshenko-theory for the bending terms whereas for the torsional terms a St-Venant theory is used. The element allows an eccentric formulation. Thus, it should be used especially in combination with plate and shell elements. If the eccentricity is not used the similar Bernoulli-element (ELMT04) is advantageous. \([\text{ndm}=3, \text{ndf}=6, \text{nel}=2(+1)]\)

The base vectors can be defined in two ways:

- **Element with 3 nodes**: introduce third node for each element (generation of elements via macro `el3b`)
  \[ \mathbf{e}_xL \text{ in element direction} \quad \mathbf{e}_yL = [x_3 - x_1] \times \mathbf{e}_xL \quad \mathbf{e}_zL = \mathbf{e}_xL \times \mathbf{e}_yL \]

- **Element with 2 nodes**: global z-axis is used for definition of the base vectors, macro `elem` can be used.
  \[ \mathbf{e}_xL \text{ in element direction} \quad \mathbf{e}_yL = \mathbf{e}_xL \times \mathbf{e}_zG \quad \text{with} \quad \mathbf{e}_zG = [0, 0, 1] \quad \mathbf{e}_zL = \mathbf{e}_xL \times \mathbf{e}_yL \]

  For the special case that the element direction is ± the global z-axis \( \mathbf{e}_yL \) is defined as \( \mathbf{e}_yL = [0, -1, 0] \).

A rotation of the local \( y_L \) and \( z_L \)-axis can be introduced by the angle \( \alpha \), which starts positive from the \( y_L \)-axis in direction to the \( z_L \)-axis. Input of \( \alpha \) in degree!

- **Material input data**

  Record 1

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E[F/L^2] )</td>
<td>Elastic modulus</td>
</tr>
<tr>
<td>( G[F/L^2] )</td>
<td>Shear modulus</td>
</tr>
<tr>
<td>( A[L^2] )</td>
<td>Area</td>
</tr>
<tr>
<td>( I_x[L^4] )</td>
<td>Moment of inertia-x / local x-axis</td>
</tr>
<tr>
<td>( I_y[L^4] )</td>
<td>Moment of inertia-y / local x-axis</td>
</tr>
<tr>
<td>( I_z[L^4] )</td>
<td>Moment of inertia-z / local x-axis</td>
</tr>
<tr>
<td>( q_x[F/L] )</td>
<td>Constant load in local x-direction / local x-axis</td>
</tr>
<tr>
<td>( q_y[F/L] )</td>
<td>Constant load in local y-direction / local x-axis</td>
</tr>
<tr>
<td>( q_z[F/L] )</td>
<td>Constant load in local z-direction / local x-axis</td>
</tr>
<tr>
<td>( \alpha[\text{DEG.}] )</td>
<td>Rotation angle for local coordinate system</td>
</tr>
</tbody>
</table>
Record 2

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lin[-]</td>
<td>0[default]=linear, 1=moderate rotations</td>
</tr>
<tr>
<td>$\rho[F/L^3]/[L/T^2]$</td>
<td>density $\rho = \gamma$ (only for LMAS)</td>
</tr>
<tr>
<td>$I_{yz}[L^4]$</td>
<td>Moment of inertia $yz = + \int yz , da$ local x-axis</td>
</tr>
<tr>
<td>$y_S[L]$</td>
<td>$y$-coordinate of center of gravity point</td>
</tr>
<tr>
<td>$z_S[L]$</td>
<td>$z$-coordinate of center of gravity point</td>
</tr>
<tr>
<td>$y_M[L]$</td>
<td>$y$-coordinate of center of shear point</td>
</tr>
<tr>
<td>$z_M[L]$</td>
<td>$z$-coordinate of center of shear point</td>
</tr>
<tr>
<td>$y_O[L]$</td>
<td>$y$-coordinate of center of output point</td>
</tr>
<tr>
<td>$z_O[L]$</td>
<td>$z$-coordinate of center of output point</td>
</tr>
<tr>
<td>scf[-]</td>
<td>shear correction 0[def.]=on, 1=off</td>
</tr>
</tbody>
</table>

• Loads via macro qloa.

<table>
<thead>
<tr>
<th>ma [-]</th>
<th>Material number in Inputfile</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_x[F/L]$</td>
<td>constant load in local x-direction</td>
</tr>
<tr>
<td>$q_y[F/L]$</td>
<td>constant load in local y-direction</td>
</tr>
<tr>
<td>$q_z[F/L]$</td>
<td>constant load in local z-direction</td>
</tr>
</tbody>
</table>

• Displacements and base vectors for element

\[
\begin{align*}
\textbf{\text{\phi}_x} & = (\text{\textbf{x}_1\textbf{X}_2}) \times \textbf{\text{\phi}_L} \\
\textbf{\text{\phi}_y} & = \textbf{\text{\phi}_L} \\
\textbf{\text{\phi}_z} & = (\text{\textbf{x}_3\textbf{X}_4}) \times \textbf{\text{\phi}_L} \\
\textbf{\text{\phi}_w} & = \textbf{\text{\phi}_L} \\
\textbf{\text{\phi}_v} & = \textbf{\text{\phi}_L} \\
\textbf{\text{\phi}_u} & = (\text{\textbf{x}_2\textbf{X}_3}) \times \textbf{\text{\phi}_L} \\
\end{align*}
\]

• Output of stress resultants

Stress resultants ($N_x$-Axial Force, $Q_y$-Shear Force, $Q_z$-Shear Force, $M_x$-Torsional Moment, $M_y$-Bending Moment, $M_z$-Bending Moment) are calculated at left and right node of the element in local directions at the defined output point.

• Plot of stress resultants [FORC]

The above defined stress resultants are plotted due to following numbers

<table>
<thead>
<tr>
<th>Force number</th>
<th>Stress resultant</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$N_x$ [F]</td>
</tr>
<tr>
<td>2</td>
<td>$Q_y$ [F]</td>
</tr>
<tr>
<td>3</td>
<td>$Q_z$ [F]</td>
</tr>
<tr>
<td>4</td>
<td>$M_x$ [FL]</td>
</tr>
<tr>
<td>5</td>
<td>$M_y$ [FL]</td>
</tr>
<tr>
<td>6</td>
<td>$M_z$ [FL]</td>
</tr>
</tbody>
</table>

• Remark: The local base vectors can be visualized for control of input data with forc,7.

• Remark: Stress resultants are plotted in the local 1-3 direction. The plot plane can be modified with forc,±12 or ±13 (default 13).
### 6.4.12 ELMT12

**ityp, kat, sym**

$K_1, K_2, \text{angl}1 - x, \rho \cdot h, \rho \cdot c, \rho \cdot g, \text{elev}$

**Theory**

ELMT12 is a general plane and axisymmetric element for use in the analysis of problems modeled by a **Laplace equation**. In general 3–9 nodes are allowed. The element has been written with options which use parameters defined by the linear heat transfer analysis, the linear flow in porous isotropic media and the calculation of shear stresses and warping functions for cross sections.

The element requires a two dimensional mesh and uses the first two coordinate values for x and y (r and z for axisymmetry), respectively. Similarly, the element uses the **first** degree–of–freedom at each node for the temperature or head nodal values. Thus with the macro **disp** for example the nodal **temperature** values are printed.

The element routine will compute any combination of nodes, up to 9, provided all vertex nodes are specified. Triangles may be specified by either giving the same node number for contiguous vertices, or for the 3–node element, by specifying only the first 3 nodes (i.e., node 4 has a zero value).

[ndm=2, ndf=2, nel=3-9]]

**Material input data** Record 1 define the type of problem:

<table>
<thead>
<tr>
<th>ityp</th>
<th>Problem type, see below</th>
</tr>
</thead>
<tbody>
<tr>
<td>kat</td>
<td>1=plane, 2=axisymmetric (not for ityp=5,6)</td>
</tr>
<tr>
<td>sym</td>
<td>Symmetry conditions (only for ityp=5,6)</td>
</tr>
<tr>
<td></td>
<td>(1: to 1–axis, 2: to 2–axis, 3: to 1+2–axis)</td>
</tr>
</tbody>
</table>

Record 2 depends on which problem has been chosen:

**Linear anisotropic heat transfer (ityp = 1):**

<table>
<thead>
<tr>
<th>$K_1$</th>
<th>Thermal conductivity in 1–direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_2$</td>
<td>Thermal conductivity in 2–direction</td>
</tr>
<tr>
<td>angl1 – x</td>
<td>Angle 1–axis makes with x(or r)–axis</td>
</tr>
<tr>
<td>$\rho \cdot h$</td>
<td>density · heat source</td>
</tr>
<tr>
<td>$\rho \cdot c$</td>
<td>density · specific heat</td>
</tr>
<tr>
<td>$\rho \cdot g$</td>
<td>not used</td>
</tr>
<tr>
<td>elev</td>
<td>not used</td>
</tr>
</tbody>
</table>
• Flow in horizontal oriented porous media (ityp = 2):

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1$</td>
<td>Flow permeability in 1–direction</td>
</tr>
<tr>
<td>$K_2$</td>
<td>Flow permeability in 2–direction</td>
</tr>
<tr>
<td>$\text{angl}_1 - x$</td>
<td>Angle 1–axis makes with x(or r)–axis</td>
</tr>
<tr>
<td>$\rho \cdot h$</td>
<td>density · flow source</td>
</tr>
<tr>
<td>$\rho \cdot c$</td>
<td>density · specific flow</td>
</tr>
<tr>
<td>$\rho \cdot g$</td>
<td>not used</td>
</tr>
<tr>
<td>elev</td>
<td>not used</td>
</tr>
</tbody>
</table>

• Flow in vertically oriented porous media (ityp = 3):

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1$</td>
<td>Flow permeability in 1–direction</td>
</tr>
<tr>
<td>$K_2$</td>
<td>Flow permeability in 2–direction</td>
</tr>
<tr>
<td>$\text{angl}_1 - x$</td>
<td>Angle 1–axis makes with x(or r)–axis</td>
</tr>
<tr>
<td>$\rho \cdot h$</td>
<td>density · flow source</td>
</tr>
<tr>
<td>$\rho \cdot c$</td>
<td>density · specific flow</td>
</tr>
<tr>
<td>$\rho \cdot g$</td>
<td>density · $g$</td>
</tr>
<tr>
<td>elev</td>
<td>Surface elevation of zero pressure when head = 0</td>
</tr>
</tbody>
</table>

• Simulated free surface in a horizontal porous media (ityp = 4):

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1$</td>
<td>Flow permeability in 1–direction</td>
</tr>
<tr>
<td>$K_2$</td>
<td>Flow permeability in 2–direction</td>
</tr>
<tr>
<td>$\text{angl}_1 - x$</td>
<td>Angle 1–axis makes with x(or r)–axis</td>
</tr>
<tr>
<td>$\rho \cdot h$</td>
<td>density · flow source</td>
</tr>
<tr>
<td>$\rho \cdot c$</td>
<td>density · specific flow</td>
</tr>
<tr>
<td>$\rho \cdot g$</td>
<td>not used</td>
</tr>
<tr>
<td>elev</td>
<td>not used</td>
</tr>
</tbody>
</table>

• Cross section properties, warping function and shear stresses from $M_T$ (ityp = 5):

Input data only on ksym, see above.
A boundary condition has to be set only at one (unloaded) node.

• Cross section properties, stress function and shear stresses from $M_T$, also center of stress function $(y_0, z_0)$ (ityp = 6):

Input data only on ksym, see above.
For such problems zero boundary conditions have to be defined at the surface of the system. If more than one surface exists all nodes at a certain surface must have the same boundary values. This can be achieved by using the macro `link`.

There exist no external load case.

- **warping function and shear stresses from** \( Q_y, Q_z \) \((ityp = 7)\):

| \( A \) | area of cross section |
| \( I_y \) | moment of inertia |
| \( I_z \) | moment of inertia |
| \( I_{yz} \) | moment of inertia = \( \int yz \, da \) |
| \( y_s \) | \( y \)-coordinate of center of gravity |
| \( z_s \) | \( z \)-coordinate of center of gravity |
| \( Q_y \) | shear force: 0 or 1 |
| \( Q_z \) | shear force: 1 or 0 |
| \( \nu \) | Poisson’s ratio |
| \( y_0 \) | \( y \)-coordinate of center of stress function (only for \( \nu \neq 0 \)) |
| \( z_0 \) | \( z \)-coordinate of center of stress function (only for \( \nu \neq 0 \)) |

\((y_0 \text{ and } z_0 \text{ can be calculated under } ityp = 6)\)

A boundary condition has to be set only at one node.

- **Print of quantities**

The output quantities for each option are printed using the basic form or the `stre`, `n1`, `n2`, `n3` macro command and are reported as follows:

- **Linear anisotropic heat transfer** \((ityp = 1)\):

| \( q_1 \) | Heat flow in 1–direction (1=x in plane, 1=r in axisym) |
| \( q_2 \) | Heat flow in 1–direction (2=y in plane, 2=z in axisym) |
| \( |q| \) | Maximum heat flow \((\sqrt{q_1^2 + q_2^2})\) |
| \( \Theta \) | Temperature at element output point |

- **Flow in a horizontal porous media** \((ityp = 2)\):

| \( q_1 \) | Fluid flow in 1–direction (1=x in plane, 1=r in axisym) |
| \( q_2 \) | fluid flow in 1–direction (2=y in plane, 2=z in axisym) |
| \( |q| \) | Maximum fluid flow \((\sqrt{q_1^2 + q_2^2})\) |
| \( p \) | Pressure at element output point |
• **Flow in a vertically oriented porous media (ityp = 3):**

| $q_1$ | Fluid flow in 1–direction (1=x in plane, 1=r in axisym) |
| $q_2$ | Fluid flow in 1–direction (2=y in plane, 2=z in axisym) |
| $|q|$ | Maximum fluid flow ($\sqrt{q_1^2 + q_2^2}$) |
| $p$ | Pressure at element output point ($p = \text{head} - \rho \times (y - y_0)$) |

• **Simulated free surface in a horizontal porous media (ityp = 4):**

| $q_1$ | Fluid flow in 1–direction (1=x in plane, 1=r in axisym) |
| $q_2$ | Fluid flow in 1–direction (2=y in plane, 2=z in axisym) |
| $|q|$ | Maximum fluid flow ($\sqrt{q_1^2 + q_2^2}$) |
| $h$ | Head at element output point ($h = \sqrt{2 \times u - 1}$) |

(N.B. $u-1$ is the dependent nodal variable used in the analysis.

• **Cross section properties, warping function and shear stresses from $M_T$ (ityp = 5):**

The first call of the macro `stre,all` leads to the cross sectional values $A$, $I_{11}$, $I_{22}$, $I_{12}$ for the reference axis, for parallel axis and main axis through the center of gravity and the coordinates of the center of gravity.

The second call of the macro `stre,all` leads to the coordinates of the center of shear.

Finally the third call of the macro `stre,all` leads to the cross sectional values $I_T$ and $C_M$.

After these three calls has been done (for simplicity define a procedure [ `tang,.1 ; loop,.3 ; stre,all ; next` ] ) shear stresses are available using the standard call `stre,,n1,n2,n3`. Note that plots of all values are only available after the above mentioned three calls.

Values of the warping functions can be printed with the macro `stre,node,n1,n2,n3`.

• **Cross section properties, stress function and shear stresses from $M_T$ (ityp = 6):**

see ityp = 5

• **Warping function and shear stresses from $Q_y, Q_z$ (ityp = 7):**

The macro `stre,all` has all to be used 3 times to get all results.

After these three calls has been done (for simplicity define a procedure [ `tang,.1 ; loop,.3 ; stre,all ; next` ] ) shear stresses are available using the standard call `stre,,n1,n2,n3`. Note that plots of all values are only available after the above mentioned three calls.

Values of the warping functions can be printed with the macro `stre,node,n1,n2,n3`.

• **Plot of quantities**

Basic quantities are plotted via the macro `disp` due to following numbers
| disp number          |       |       |
|----------------------|-------|
| disp (ityp = 1)      | 1     |
| disp (ityp = 2)      | temperature |
| disp (ityp = 3)      | pressure |
| disp (ityp = 4)      | pressure |
| disp (ityp = 5)      | warping function $w$ from $M_T$ (not $w_Q$, $w_T$) |
| disp (ityp = 6)      | stress function $\Phi$ |
| disp (ityp = 7)      | warping function $w$ from $Q_y, Q_z$ |

All quantities are plotted via the macro `stre` due to following numbers

<table>
<thead>
<tr>
<th>'Stress' number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>'stress' (ityp = 1)</td>
<td>$q_1$</td>
<td>$q_2$</td>
<td>$</td>
<td>q</td>
<td>$</td>
</tr>
<tr>
<td>'stress' (ityp = 2)</td>
<td>$q_1$</td>
<td>$q_2$</td>
<td>$</td>
<td>q</td>
<td>$</td>
</tr>
<tr>
<td>'stress' (ityp = 3)</td>
<td>$q_1$</td>
<td>$q_2$</td>
<td>$</td>
<td>q</td>
<td>$</td>
</tr>
<tr>
<td>'stress' (ityp = 4)</td>
<td>$q_1$</td>
<td>$q_2$</td>
<td>$</td>
<td>q</td>
<td>$</td>
</tr>
<tr>
<td>'stress' (ityp = 5)</td>
<td>$T_{13}$</td>
<td>$T_{23}$</td>
<td>$</td>
<td>T</td>
<td>$</td>
</tr>
<tr>
<td>'stress' (ityp = 6)</td>
<td>$T_{13}$</td>
<td>$T_{23}$</td>
<td>$</td>
<td>T</td>
<td>$</td>
</tr>
<tr>
<td>'stress' (ityp = 7)</td>
<td>$T_{13}$</td>
<td>$T_{23}$</td>
<td>$</td>
<td>T</td>
<td>$</td>
</tr>
</tbody>
</table>

Remark: Note that for ityp=5,6,7 plots of all values are only available after calling three times `stre`, all in the macro mode.

The resultant flows (stresses) can be plotted with arrows using the macro `flux`. 
6.4.13 ELMT13

$E$, $\nu$, $h$, $q$, $\rho$, $lin$

- **Theory**

ELMT13 is a general 4/8/9 - node thin shallow shell element based on the Reissner–Mindlin Theory with selective reduced integration (SRI). Be careful, this element may lead to hourglass-modes. The element formulation is based on the Green Lagrangian strain tensor and the 2. Piola-Kirchhoff stress tensor. No transformation to local coordinates has to be done. Thus, all results are stated in the 1-2 plane.

[ndm=3, ndf=5, nel=4, 8, 9]

- **Material input data**

  | $E[F/L^2]$ | Modulus of elasticity |
  | $\nu[-]$ | Poisson ratio |
  | $h[L]$ | Plate thickness |
  | $q[F/L^2]$ | Uniform normal load in z-dir. |
  | $\rho[F/L^3]/[L/T^2]$ | density : $\rho = \gamma/g$ |
  | $lin$ | linear/nonlinear = 0/1 |

- **Loads** via macro $qloa$.

  | $ma$ | Material number in Inputfile |
  | $q[F/L^2]$ | uniform normal load in z-direction |

- **Definition of stress resultants**

  $n_{xx} = \int \sigma_{xx} \, dz$, \hspace{0.5cm} $n_{yy} = \int \sigma_{yy} \, dz$, \hspace{0.5cm} $n_{xy} = \int \sigma_{xy} \, dz = n_{yx}$

  $m_{xx} = \int \sigma_{xx} \, z \, dz$, \hspace{0.5cm} $m_{yy} = \int \sigma_{yy} \, z \, dz$, \hspace{0.5cm} $m_{xy} = \int \sigma_{xy} \, z \, dz = m_{yx}$

  $q_{xz} = \int \tau_{xz} \, dz$, \hspace{0.5cm} $q_{yz} = \int \tau_{yz} \, dz$

- **Displacements ($u_x, u_y, w, \varphi_x, \varphi_y$) and stress resultants per length**

  Shear forces show on positive side in positive direction.
• **Output of moments/shear forces**

The normal forces/moment/shear forces per unit length are calculated at each center of element in global directions x and y. The moments $m_1$ and $m_2$ are the principal moments with the angle $\varphi_1$ between 1–moment and x-coordinate direction (in degrees).

• **Plot of moments/shear forces**

The normal forces/moment/shear forces per unit length are plotted due to following numbers:

<table>
<thead>
<tr>
<th>Stress number</th>
<th>Type</th>
<th>value plotted</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Force</td>
<td>$n_{xx}$</td>
<td>$[F/L]$</td>
</tr>
<tr>
<td>2</td>
<td>Force</td>
<td>$n_{yy}$</td>
<td>$[F/L]$</td>
</tr>
<tr>
<td>3</td>
<td>Force</td>
<td>$n_{xy}$</td>
<td>$[F/L]$</td>
</tr>
<tr>
<td>4</td>
<td>Moment</td>
<td>$m_{xx}$</td>
<td>$[FL/L]$</td>
</tr>
<tr>
<td>5</td>
<td>Moment</td>
<td>$m_{yy}$</td>
<td>$[FL/L]$</td>
</tr>
<tr>
<td>6</td>
<td>Moment</td>
<td>$m_{xy}$</td>
<td>$[FL/L]$</td>
</tr>
<tr>
<td>7</td>
<td>Shear force</td>
<td>$q_{xz}$</td>
<td>$[F/L]$</td>
</tr>
<tr>
<td>8</td>
<td>Shear force</td>
<td>$q_{yz}$</td>
<td>$[F/L]$</td>
</tr>
<tr>
<td>9</td>
<td>Moment</td>
<td>$m_1$</td>
<td>$[FL/L]$</td>
</tr>
<tr>
<td>10</td>
<td>Moment</td>
<td>$m_2$</td>
<td>$[FL/L]$</td>
</tr>
<tr>
<td>11</td>
<td>thickness</td>
<td>$h$</td>
<td>$[L]$</td>
</tr>
</tbody>
</table>
6.4.14 ELMT14

ido, pen, tol

- Theory
ELMT14 is a 2D/3D node-to-node contact element. Contact is possible in one direction. The gap is defined by

\[(x_2 + u_2) - (x_1 + u_1) > 0\]

where x is the chosen direction. A penalty and an augmented lagrange (augm) formulation is available.

- Material input data

<table>
<thead>
<tr>
<th>idof</th>
<th>coordinate direction for contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipen</td>
<td>penalty value for imposing contact constraint</td>
</tr>
<tr>
<td>tol</td>
<td>tolerance on gap (def. = 0.0)</td>
</tr>
</tbody>
</table>

- Displacements and normal forces

- Output of results
Contact force N.

- Plot of results
Results are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity plotted</td>
<td>N [F]</td>
</tr>
</tbody>
</table>
6.4.15 ELMT15

ELMT15 is a 2–5 node geometrically nonlinear 3D-Timoshenko–Beam Element with Reissner strains and finite rotations. 

[ndm=3, ndf=6, nel={2,3,4,5}]

### Theory

**ELMT15** is a 2–5 node geometrically nonlinear 3D-Timoshenko–Beam Element with Reissner strains and finite rotations.

\[\begin{align*}
E & \quad [F/L^2] \quad \text{Elastic modulus} \\
G & \quad [F/L^2] \quad \text{Shear modulus} \\
A & \quad [L^2] \quad \text{Area} \\
I_y & \quad [L^4] \quad \text{Moment of Inertia w.r.t. y-axis} \\
I_z & \quad [L^4] \quad \text{Moment of Inertia w.r.t. z-axis} \\
I_{yz} & \quad [L^4] \quad \text{Moment of Inertia w.r.t. yz-axis} \\
I_T & \quad [L^4] \quad \text{Moment of Inertia w.r.t. x-axis} \\
y_s & \quad [L] \quad \text{y-coordinate of center of gravity} \\
zt & \quad [L] \quad \text{z-coordinate of center of gravity} \\
y_m & \quad [L] \quad \text{y-coordinate of center of shear} \\
y_m & \quad [L] \quad \text{z-coordinate of center of shear} \\
\alpha & \quad [DEG.] \quad \text{Rotation angle for local coordinate system} \\
iGeo & \quad \text{geometrically nonlinearity} \\
iPlo & \quad 0/1 \text{ Stress resultants related to reference(0)/current(1) configuration} \\
iscf & \quad 0/1 \text{ FE shear correction off(0)/on(1)} \\
\rho & \quad [F/L^3]/[L/T^2] \quad \text{density } \rho = \gamma/g
\end{align*}\]

- **Material input data**

- **Loads** via macro `qloa`.

- **Displacements and stress resultants**
• **Output of stress resultants**

Stress resultants (N₁–Axial Force, Q₂,Q₃–Shear Forces, M₁ Torsional Moment, M₂,M₃–Bending Moments) are calculated at left and right node of the element with respect to iPlo=0/1 related to reference(0) or current(1) configuration.

• **Plot of stress resultants**

The above defined stress resultants are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress resultant</td>
<td>N₁</td>
<td>Q₂</td>
<td>Q₃</td>
<td>M₁</td>
<td>M₂</td>
<td>M₃</td>
</tr>
<tr>
<td></td>
<td>[F]</td>
<td>[F]</td>
<td>[F]</td>
<td>[FL]</td>
<td>[FL]</td>
<td>[FL]</td>
</tr>
</tbody>
</table>
6.4.16  ELMT16

\[ E, A, \rho, \alpha_t, s_0/\ell_0 \]
\[ S_0, q_x, q_y, q_z, \Delta T \]

- **Theory**
ELMT16 is a general 3D cable element. \([\text{ndm}=3, \text{ndf}=3, \text{nel}=2]\)

- **Material input data**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(E)</td>
<td>([F/L^2])</td>
<td>Young's modulus</td>
</tr>
<tr>
<td>(A)</td>
<td>([L^2])</td>
<td>Area</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>([F/L^3])</td>
<td>Specific weight for dead load in global -z direction</td>
</tr>
<tr>
<td>(\alpha_t)</td>
<td>([1/K])</td>
<td>Coefficient of thermal expansion</td>
</tr>
<tr>
<td>(s_0/\ell_0)</td>
<td>[-]</td>
<td>Cable length/chord length ((\ell_0 = X_2 - X_1))</td>
</tr>
<tr>
<td>(S_0)</td>
<td>([F])</td>
<td>Prestress force in chord dir.</td>
</tr>
<tr>
<td>(q_x)</td>
<td>([F/L])</td>
<td>Load in global x-direction</td>
</tr>
<tr>
<td>(q_y)</td>
<td>([F/L])</td>
<td>Load in global y-direction</td>
</tr>
<tr>
<td>(q_z)</td>
<td>([F/L])</td>
<td>Load in global z-direction</td>
</tr>
<tr>
<td>(\Delta T)</td>
<td>([K])</td>
<td>Temperature difference</td>
</tr>
</tbody>
</table>

- **Displacements and normal forces**

- **Output of results**
Cable forces are calculated for each element in the local axial direction at beginning and end of element as well as for the chord direction.

- **Plot of results**
Results are plotted due to following numbers

<table>
<thead>
<tr>
<th>Force number</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cable Forest</td>
<td>(S \ [F])</td>
</tr>
</tbody>
</table>

Remark: Cable forces are plotted in the local 1-3 direction. The plot plane can be modified with \text{forc} ,±12 or ±13 (default 13).

- **Remarks**
- Each cable can be modelled with one element.
- In case of large deflections more than one element should be used. The position of all nodal points is on the chord-line! 4 elements are recommended.
- The problem to find the right cable position is nonlinear. Thus an iteration is necessary!
- The deformed configuration of the cable could only be seen in case of more than one element describing the cable!
6.4.17 ELMT17

$E, \nu, h, q, \rho$

- **Theory**
ELMT17 is a general 4-node thin plate bending element based on the Reissner-Mindlin Theory with special assumptions for the shear terms (Bathe/Dvorkin approach).

$[\text{ndm}=3, \text{ndf}=3, \text{nel}=4]$

- **Material input data**

| $E[F/L^2]$ | Modulus of elasticity |
| $\nu[-]$ | Poisson ratio |
| $h[L]$ | Plate thickness |
| $q[F/L^2]$ | Uniform normal load in z-dir. |
| $\rho[F/L^3]/[L/T^2]$ | density : $\rho = \gamma/g$ |
| $SCF[-]$ | Shear correction factor (def=5/6) |

- **Loads** via macro `qloa`.

| $ma$ | Material number in Inputfile |
| $q[F/L^2]$ | uniform normal load in z-direction |

- **Definition of stress resultants**

$m_{xx} = \int \sigma_{xx} z \, dz$,  
$m_{yy} = \int \sigma_{yy} z \, dz$,  
$m_{xy} = \int \sigma_{xy} z \, dz = m_{yx}$

$q_{xz} = \int \tau_{xz} \, dz$,  
$q_{yz} = \int \tau_{yz} \, dz$

- **Displacements ($w, \varphi_x, \varphi_y$) and moments per length**

Shear forces show on positive side in positive direction.

- **Output of moments/shear forces**

The moments/shear forces per unit length are calculated at each center of element in global directions x and y. The moments $m_1$ and $m_2$ are the principal moments with the angle $\varphi_1$ between 1-moment and x-coordinate direction (in degrees).
• Plot of moments/shear forces

The moments/shear forces per unit length are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>Type</th>
<th>value plotted</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Moment</td>
<td>(m_{xx})</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>2</td>
<td>Moment</td>
<td>(m_{xy})</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>3</td>
<td>Moment</td>
<td>(m_{yy})</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>5</td>
<td>Moment</td>
<td>(m_1)</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>6</td>
<td>Moment</td>
<td>(m_2)</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>7</td>
<td>Angle</td>
<td>(\varphi_1)</td>
<td>([rad])</td>
</tr>
<tr>
<td>8</td>
<td>Shear force</td>
<td>(q_{xz})</td>
<td>([F/L])</td>
</tr>
<tr>
<td>9</td>
<td>Shear force</td>
<td>(q_{yz})</td>
<td>([F/L])</td>
</tr>
</tbody>
</table>

Element-wise constant values can be plotted with macro `str1`.

• Shear correction factor \(\kappa\)

<table>
<thead>
<tr>
<th>input value SCF</th>
<th>(\kappa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(SCF &lt; 0)</td>
<td>(\kappa = SCF/(1 + \frac{1}{2}(1 + \nu)\frac{L^2}{h^2})) with (L = \max(L_x, L_y)), see Tessler, Hughes</td>
</tr>
<tr>
<td>(SCF = 0)</td>
<td>(\kappa = \frac{5}{6})</td>
</tr>
<tr>
<td>(SCF &gt; 0)</td>
<td>(\kappa = SCF)</td>
</tr>
</tbody>
</table>
6.4.18 ELMT18

\[ E, \nu, h, q, \rho, nb, ns, \kappa \]

- **Theory**

ELMT18 is a general 4/8/9 - node thin/thick plate bending element based on the Reissner–Mindlin Theory with selective reduced integration (SRI). Be careful, this element may lead to hourglass-modes for thin plates using a 2/1 or 3/2 integration. \([\text{ndm}=3, \text{ndf}=3, \text{nel}=4,8,9]\)

- Definition: thin plate: \(L/h > 10\), thick plate: \(L/h \leq 10\)

- Integration: thin plate 2/1 or 3/2, thick plate 2/2 or 3/3

- **Material input data**

| \(E[F/L^2]\) | Modulus of elasticity |
| \(\nu[-]\) | Poisson ratio |
| \(h[L]\) | Plate thickness |
| \(q[F/L^2]\) | Uniform normal load in z-dir. |
| \(\rho[F/L^3]/[L/T^2]\) | density: \(\rho = \gamma/g\) |
| \(nb\) | No. of Gauss points bending (def=2) |
| \(ns\) | No. of Gauss points shear (def=1) |
| \(SCF[-]\) | Shear correction factor (def=5/6) |

- **Loads via macro** \(qloa\).

| \(ma\) | [-] | Material number in Inputfile |
| \(q\) | \([F/L^2]\) | uniform normal load in z-direction |

- Definition of stress resultants

\[ m_{xx} = \int \sigma_{xx} \, z \, dz, \quad m_{yy} = \int \sigma_{yy} \, z \, dz, \quad m_{xy} = \int \sigma_{xy} \, z \, dz = m_{yx} \]

\[ q_{xx} = \int \tau_{xx} \, dz, \quad q_{yz} = \int \tau_{yz} \, dz \]

- **Displacements** \((w, \varphi_x, \varphi_y)\) and moments per length

Shear forces show on positive side in positive direction.
• Output of moments/shear forces
The moments/shear forces per unit length are calculated at each center of element in global directions x and y. The moments $m_1$ and $m_2$ are the principal moments with the angle $\varphi_1$ between 1–moment and x-coordinate direction (in degrees).

• Plot of moments/shear forces
The moments/shear forces per unit length are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>Type</th>
<th>value plotted</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Moment</td>
<td>$m_{xx}$</td>
<td>[F/L]</td>
</tr>
<tr>
<td>2</td>
<td>Moment</td>
<td>$m_{xy}$</td>
<td>[F/L]</td>
</tr>
<tr>
<td>3</td>
<td>Moment</td>
<td>$m_{yy}$</td>
<td>[F/L]</td>
</tr>
<tr>
<td>5</td>
<td>Moment</td>
<td>$m_1$</td>
<td>[F/L]</td>
</tr>
<tr>
<td>6</td>
<td>Moment</td>
<td>$m_2$</td>
<td>[F/L]</td>
</tr>
<tr>
<td>7</td>
<td>Angle</td>
<td>$\varphi_1$</td>
<td>[rad]</td>
</tr>
<tr>
<td>8</td>
<td>Shear force</td>
<td>$q_{xz}$</td>
<td>[F/L]</td>
</tr>
<tr>
<td>9</td>
<td>Shear force</td>
<td>$q_{yz}$</td>
<td>[F/L]</td>
</tr>
</tbody>
</table>

Element-wise constant values can be plotted with macro str1.

• Shear correction factor $\kappa$
input value SCF $\kappa$

$\begin{array}{c|c}
SCF < 0 & \kappa = \frac{SCF}{1 + \frac{1}{2} \left( \frac{1}{1 + \nu} \frac{L^2}{h^2} \right)} \\
SCF = 0 & \kappa = 5 \\
SCF > 0 & \kappa = SCF \\
\end{array}$

with $L = \max(L_x, L_y)$, see Tessler, Hughes
### 6.4.19 ELMT19

**Theory**

ELMT19 is a general 3-node thin plate bending element based on the Discrete Kirchhoff Theory, compare ELMT07 for a 4-node element. \([\text{ndm}=3, \text{ndf}=3, \text{nel}=3]\)

**Material input data**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(E[F/L^2])</td>
<td>Modulus of elasticity</td>
</tr>
<tr>
<td>(\nu[-])</td>
<td>Poisson ratio</td>
</tr>
<tr>
<td>(h[L])</td>
<td>Plate thickness</td>
</tr>
<tr>
<td>(q[F/L^2])</td>
<td>Uniform normal load in z-dir.</td>
</tr>
</tbody>
</table>

**Loads** via macro `qloa`.

<table>
<thead>
<tr>
<th>Material number in Inputfile</th>
<th>uniform normal load in z-direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ma)</td>
<td>(q[F/L^2])</td>
</tr>
</tbody>
</table>

**Definition of stress resultants**

\[ m_{xx} = - \int \sigma_{xx} z \, dz, \quad m_{yy} = - \int \sigma_{yy} z \, dz, \quad m_{xy} = - \int \sigma_{xy} z \, dz = m_{yx} \]

**Displacements** \((w, \varphi_x, \varphi_y)\) and moments per length

![Displacement and moment diagram](https://via.placeholder.com/150)

**Output of moments**

The moments per unit length are calculated at each center of element in global directions \(x\) and \(y\). The moments \(m_1\) and \(m_2\) are the principal moments with the angle \(\varphi_1\) between 1-moment and \(x\)-coordinate direction (in degrees).

**Plot of moments**

The moments per unit length are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>Type</th>
<th>value plotted</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Moment</td>
<td>(m_{xx})</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>2</td>
<td>Moment</td>
<td>(m_{xy})</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>3</td>
<td>Moment</td>
<td>(m_{yy})</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>5</td>
<td>Moment</td>
<td>(m_1)</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>6</td>
<td>Moment</td>
<td>(m_2)</td>
<td>([FL/L])</td>
</tr>
<tr>
<td>7</td>
<td>Angle</td>
<td>(\varphi_1)</td>
<td>([\text{rad}])</td>
</tr>
</tbody>
</table>
6.4.20 ELMT20

\[ E, G, I_T, I_y, q_1, q_2 \]

- **Theory**

**ELMT20** is a 2–node Bernoulli/St.Venant Grid–Beam Element. \([\text{ndm}=3, \text{ndf}=3, \text{nel}=2]\)

- **Material input data**

| \( E[F/L^2] \) | Elastic modulus |
| \( G[F/L^2] \) | Shear modulus |
| \( I_T[L^4] \) | Torsional rigidity |
| \( I_y[L^4] \) | Moment of Inertia |
| \( q_1[F/L] \) | load in z-dir. at left node |
| \( q_2[F/L] \) | load in z-dir. at right node |

- **Loads** via macro \( \text{qloa} \).

| \( ma \) | Material number in Inputfile |
| \( q_1[F/L] \) | load in z-dir. at left node |
| \( q_2[F/L] \) | load in z-dir. at right node |

- **Displacements and stress resultants**

\[ v = [w, \ \varphi_x, \ \varphi_y]^T \]
\[ S = [Q_z, M_T, M_y]^T \]

- **Output of stress resultants**

Stress resultants (\( Q_z \)–Shear Force, \( M_T \)–Bending Moment, \( M_y \)–Bending Moment) are calculated at left and right node of the element in local directions.

- **Plot of stress resultants**

The above defined stress resultants are plotted due to following numbers:

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress resultant</td>
<td>( Q_z ) ([F])</td>
<td>( M_T ) ([F \cdot L])</td>
<td>( M_T ) ([F \cdot L])</td>
</tr>
</tbody>
</table>
6.4.21 ELMT21

matn, < lin >, < isym, istr, ngpk, ngps >

$q_x, q_y, q_z, \Delta T, \alpha_t, \rho$

$E, \nu$ (for matn=1)

• Theory

ELMT21 is a 4/8/20/21/27/64-node geometrical linear/nonlinear 3D-Solid Element. The definition of the nodes is similar to the macro bloc. The 20 node element can be used with nen=20 or nen=21. In the latter case an additional unused node is generated. Hidden line plots are only available for the 8 node element.

[ndm=3, ndf=3, nel=4/8/20/21/27/64]

Different material models are implemented and can be used with the 3D-Material library.

• Material input data

<table>
<thead>
<tr>
<th>matn</th>
<th>[-]</th>
<th>material number</th>
</tr>
</thead>
<tbody>
<tr>
<td>lin</td>
<td>[-]</td>
<td>0: geometrical linear, 1: geometrical nonlinear</td>
</tr>
<tr>
<td>isym</td>
<td>[-]</td>
<td>0: symmetric, 1: nonsymmetric</td>
</tr>
<tr>
<td>istr</td>
<td>[-]</td>
<td>stresses(lin=0): 1&lt;0&gt;=linear stresses(lin=1): 1&lt;0&gt;=2.P.K, 2=Cauchy</td>
</tr>
<tr>
<td>ngpk</td>
<td>[-]</td>
<td>default = 4/3/3/3/3: No. of Gauss-Points for $K$</td>
</tr>
<tr>
<td>ngps</td>
<td>[-]</td>
<td>default = 4/3/3/3/3: No. of Gauss-Points for $\sigma$</td>
</tr>
</tbody>
</table>

$q_x \ [F/L^3]$ load in global x-direction

$q_y \ [F/L^3]$ load in global y-direction

$q_z \ [F/L^3]$ load in global z-direction

$\Delta T \ [K]$ temperature difference

$\alpha_t \ [1/K]$ coefficient of thermal expansion

$\rho \ [F/L^3]/[L^2/T]$ density : $\rho = \gamma/g$

Material input data, see the 3D-Material library

Details on the derivation of the material models can be found in the Theory Manual.

Details on the implementation of elements can be found in the Manual for adding elements.

• Options for integration

The above mentioned default values for ngpk and ngps are used. Sometimes other options may lead to better results or a faster calculation.

ngpk = ngps – 1 may prevent oscillations of shear stresses in the elastic case.

ngpk = ngps = 6 (in total 6 integration points) may lead to a faster solution for the 8-node element.

ngpk = ngps = 14 (in total 14 integration points) may lead to a faster solution for the 20-27-node elements.

• Loads via macro qloa.

<table>
<thead>
<tr>
<th>ma</th>
<th>[-]</th>
<th>Material number in Inputfile</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_x \ [F/L^3]$</td>
<td>load in global x-direction</td>
<td></td>
</tr>
<tr>
<td>$q_y \ [F/L^3]$</td>
<td>load in global y-direction</td>
<td></td>
</tr>
<tr>
<td>$q_z \ [F/L^3]$</td>
<td>load in global z-direction</td>
<td></td>
</tr>
<tr>
<td>$\Delta T \ [K]$</td>
<td>temperature difference</td>
<td></td>
</tr>
</tbody>
</table>
• Displacements and stress/strain resultants

\[ S = \begin{bmatrix} S_{11} \\ S_{22} \\ S_{33} \\ S_{12} \\ S_{13} \\ S_{23} \end{bmatrix}, \quad E = \begin{bmatrix} E_{11} \\ E_{22} \\ E_{33} \\ 2E_{12} \\ 2E_{13} \\ 2E_{23} \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \]

• Output of stresses
Normal and shear stresses are calculated at the Gauss points of the element in global directions.

• Plot of stresses
The above defined stresses are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress plotted [F/L^2]</td>
<td>S_{11}</td>
<td>S_{22}</td>
<td>S_{33}</td>
<td>S_{12}</td>
<td>S_{13}</td>
<td>S_{23}</td>
<td>S_1</td>
<td>S_2</td>
<td>S_3</td>
</tr>
</tbody>
</table>
• Plot problems with tetraeder
Macros disp, stre and similar macros do not work together with hide. Thus, use other post processing programs, e.g. TECPLOT for plotting of results. An associated interface to TECPLOT is implemented (tec).

• Element nodes
(see macro bloc)
6.4.22  ELMT22

\[ E, \nu, h, \alpha_T, \Delta T, \text{lin}, Y_0, c_p \]

- **Theory**

ELMT22 is a general 4/8/9 node plane stress element. The 4–node version is a quadrilateral with linear shape functions, whereas the 9–node element has quadratical shape functions. The 8–node element is of Serendipity–type. A geometrical nonlinear option and a material nonlinear option (*J*\textsubscript{2}-plasticity with isotropic hardening) are available.

\[ [\text{ndm}=2, \text{ndf}=2, \text{nel}=4/8/9] \]

- **Material input data**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E[F/L^2] )</td>
<td>Young’s modulus</td>
</tr>
<tr>
<td>( \nu[-] )</td>
<td>Poisson ratio</td>
</tr>
<tr>
<td>( h[L] )</td>
<td>thickness of plane stress slice</td>
</tr>
<tr>
<td>( \alpha_T[1/K] )</td>
<td>coefficient of linear thermal expansion</td>
</tr>
<tr>
<td>( \Delta T[K] )</td>
<td>temperature difference for this material</td>
</tr>
<tr>
<td>lin</td>
<td>0=linear, 1=nonlinear</td>
</tr>
<tr>
<td>( Y_0[F/L^2] )</td>
<td>initial yield stress</td>
</tr>
<tr>
<td>( c_p[F/L^2] )</td>
<td>Hardening parameter</td>
</tr>
</tbody>
</table>

- **Loads via macro **qloa**.

<table>
<thead>
<tr>
<th>ma</th>
<th>[\text{-}]</th>
<th>Material number in Inputfile</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b_1[F/L^3] )</td>
<td>body force value (constant) for 1(x)-direction</td>
<td></td>
</tr>
<tr>
<td>( b_2[F/L^3] )</td>
<td>body force value (constant) for 2(y)-direction</td>
<td></td>
</tr>
</tbody>
</table>

- **Displacements (u,v) and stresses (S=\( \sigma \), P)**

- **Output of stresses and strains**

Stresses and strains are calculated at each quadrature point in global directions 1(x) and 2(y). The stresses \( S_{\alpha\beta} \) and \( P_{\alpha\beta} \) are the linear stresses in case of \( \text{lin} = 0 \). For nonlinear problems \( S_{\alpha\beta} \) denotes the 2.Piola-Kirchhoff stresses and \( P_{\alpha\beta} \) the 1.Piola-Kirchhoff stresses.

- **Plot of stresses**

Stresses (\( S, P \)) and plastic strains \( E_p \) are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress plotted</td>
<td>( S_{xx} )</td>
<td>( S_{yy} )</td>
<td>( S_{xy} )</td>
<td>( E_{pxx} )</td>
<td>( E_{pyy} )</td>
<td>( E_{pxy} )</td>
<td>( E_{pq} )</td>
<td>( Y/Y_0 )</td>
<td>( \Delta\lambda_0 )</td>
<td>( P_{xx} )</td>
<td>( P_{yy} )</td>
<td>( P_{xy} )</td>
<td>( P_{yx} )</td>
</tr>
</tbody>
</table>

Plot-results for the 8–node serendipity–element could not be used due negative weighting functions.
6.4.23 ELMT23

Theory

ELMT023 is a general 4-node thin plate bending element based on the Discrete Kirchhoff Theory. This is an improved version of ELMT07.

\[ \text{ndm}=3, \text{ndf}=3, \text{nel}=4 \]

Material input data

Record 1

<table>
<thead>
<tr>
<th>E[F/L²]</th>
<th>Modulus of elasticity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ν[−]</td>
<td>Poisson ratio</td>
</tr>
<tr>
<td>h[L]</td>
<td>Plate thickness</td>
</tr>
<tr>
<td>q[F/L²]</td>
<td>Uniform normal load in z-dir.</td>
</tr>
</tbody>
</table>

Definition of stress resultants

\[
m_{xx} = -\int \sigma_{xx} z \, dz, \quad m_{yy} = -\int \sigma_{yy} z \, dz, \quad m_{xy} = -\int \sigma_{xy} z \, dz = m_{yx}
\]

Displacements \((w, \varphi_x, \varphi_y)\) and moments per length

Output of moments

The moments per unit length are calculated at each center of element in global directions \(x\) and \(y\). The moments \(m_1\) and \(m_2\) are the principal moments with the angle \(\varphi_1\) between 1–moment and \(x\)–coordinate direction (in degrees).

Plot of moments

The moments per unit length, the foundation pressure and the reinforcements are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>Type</th>
<th>Value plotted</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Moment</td>
<td>(m_{xx})</td>
<td>[FL/L]</td>
</tr>
<tr>
<td>2</td>
<td>Moment</td>
<td>(m_{xy})</td>
<td>[FL/L]</td>
</tr>
<tr>
<td>3</td>
<td>Moment</td>
<td>(m_{yy})</td>
<td>[FL/L]</td>
</tr>
</tbody>
</table>
6.4.24 ELMT24

\[ E, G, b, Q_y, Q_z, ns, nsp, ityp, M_{T0}, ic \]

- **Theory**

ELMT24 is a 2-node Element for shear stresses from shear forces or torsional moments in thin walled structures with different materials. All cross section values are calculated.

[ndm=2, ndf=1 nel=2]

For ityp=1(def) shear stresses from Q are calculated. Here, the center of shear can be found with two calculations:

1: \( y_M \) with \( Q_y = 0 \) and \( Q_z = 1 \)
2: \( z_M \) with \( Q_y = 1 \) and \( Q_z = 0 \)

For ityp=2 shear stresses for \( M_T \) are calculated for open or closed sections. Here the calculation of center of shear is done automatically. The total torsional moment is

\[ M_T = M_{T0} + Q_y \cdot z_{qm} - Q_z \cdot y_{qm} \ (Q_y, Q_z \text{ in } S) \]

For ic=0 only the closed part of the cross section is investigated (Bredt’ shear flux). For ic=1 the whole cross section is used. Thus pure open and mixed cross sections can be calculated. Note that in closed parts more than pure Bredt’ shear flux occur. In this case different stresses are calculated: Values at the border for open cross sections(linear over thickness) and constant values for closed parts (constant over thickness).

- **Material input data**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_i[F/L^2] )</td>
<td>Elastic modulus</td>
</tr>
<tr>
<td>( G_i[F/L^2] )</td>
<td>Shear modulus</td>
</tr>
<tr>
<td>( b_i[L] )</td>
<td>Thickness</td>
</tr>
<tr>
<td>( Q_y[F] )</td>
<td>Shear force</td>
</tr>
<tr>
<td>( Q_z[F] )</td>
<td>Shear force</td>
</tr>
<tr>
<td>( ns[-] )</td>
<td>Number of points for STRE (def.=4)</td>
</tr>
<tr>
<td>( nsp[-] )</td>
<td>Number of points for PLOT (def.=10)</td>
</tr>
<tr>
<td>( ityp[-] )</td>
<td>Typ: 1= Shear force, 2= Torsional moment</td>
</tr>
<tr>
<td>( M_{T0}[FL] )</td>
<td>Torsional moment</td>
</tr>
<tr>
<td>( ic[-] )</td>
<td>Typ: 0= only closed cross section, 1= open or mixed cross section</td>
</tr>
</tbody>
</table>
• Displacements and stress resultants

\[ \frac{z_2}{c_{68}} = \frac{z_1}{c_{97}} \]

• Output of stress resultants

Stress resultants (shear stress \( \tau \), shear flux \( t \) and main warping function \( \varphi_q \)) are calculated exactly at all points.

• Plot of stress resultants

The above defined stress resultants are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress resultant</td>
<td>( \tau [F/L^2] )</td>
<td>( t [F/L] )</td>
<td>( \varphi_q [-] )</td>
</tr>
</tbody>
</table>

• Necessary calculation procedure

- `chec` calculate cross section values
- `tang,.1` solve equations
- `stre,.all` calculate further cross section values
- `stre,.all` print stresses
6.4.25 ELMT25

\[ Q - N_{\text{No}}, E, \ell_1, \ell_2, \ell_3, \ell_4, \ell_5, n, q \]

- **Theory**

ELMT25 is a 2-node 2D Bernoulli–Beam Element for beams with cross-sections of variable height and width. Thus, Hermitean shape functions do not fulfill the homogeneous differential equation leading to approximate solutions. The formulations here still allow the use of one element for a haunch and base on special shape functions. Details on the theoretical and numerical background can be found in Herwig/Wagner [7]. In addition it is possible to calculate approximate solutions with Hermite shape functions and a numerical integration \( n_{\text{GP}}=3 \) for \( Q_i < 0 \).

[ndm=2, ndf=3, nel=2]

- **Material input data**

<table>
<thead>
<tr>
<th>Q-No,</th>
<th>E-modulus</th>
<th>length1</th>
<th>length2</th>
<th>length3</th>
<th>length4</th>
<th>length5</th>
<th>load n</th>
<th>load q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>E</td>
<td>( b_a )</td>
<td>( h_a )</td>
<td></td>
<td></td>
<td></td>
<td>n</td>
<td>q</td>
</tr>
<tr>
<td>2</td>
<td>E</td>
<td>( b_a )</td>
<td>( h_a )</td>
<td>( h_b )</td>
<td></td>
<td></td>
<td>n</td>
<td>q</td>
</tr>
<tr>
<td>3</td>
<td>E</td>
<td>( b_a )</td>
<td>( h_a )</td>
<td>( h_b )</td>
<td>( b_b )</td>
<td></td>
<td>n</td>
<td>q</td>
</tr>
<tr>
<td>4</td>
<td>E</td>
<td>( b_a )</td>
<td>( h_a )</td>
<td>( h_b )</td>
<td>( b_b )</td>
<td>( b_b )</td>
<td>t</td>
<td>n</td>
</tr>
<tr>
<td>5</td>
<td>E</td>
<td>( b_a )</td>
<td>( h_a )</td>
<td>( h_b )</td>
<td>( t )</td>
<td>( s )</td>
<td>n</td>
<td>q</td>
</tr>
<tr>
<td>6</td>
<td>E</td>
<td>( b_a )</td>
<td>( h_a )</td>
<td>( h_b )</td>
<td>( t )</td>
<td>( s )</td>
<td>n</td>
<td>q</td>
</tr>
</tbody>
</table>

Length-values are defined with respect to the following figures

- **Comments**

  # Q1 is only for comparison
  # Q2-Q6 only for \( h_a < h_b \) or \( b_a < b_b \)
  # Q-No. > 0: special shape functions, analytical solution
  # Q-No. < 0: Hermite shape functions, solution with numerical integration \( n_{\text{GP}}=3 \)
• Displacements and stress resultants

![](image)

- **Output of stress resultants**

Stress resultants (N–Axial Force, Q–Shear Force, M–Bending Moment) are calculated at left and right node of the element in local directions due to the standard sign convention.

- **Plot of stress resultants**

The above defined stress resultants are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress resultant</td>
<td>N [F]</td>
<td>Q [F]</td>
<td>M [F · L]</td>
</tr>
</tbody>
</table>
6.4.26 ELMT26

imat
ityp, II, ρ, K, G

- Theory ELMT26 is a 4 node plane strain/axisym element. It is called B-element and is based on a mixed formulation.
[ndm=3, ndf=3, nel=4]
- Material input data

<table>
<thead>
<tr>
<th>imat</th>
<th>1= isotropic linear elastic material</th>
</tr>
</thead>
<tbody>
<tr>
<td>ityp</td>
<td>0: plane strain, 1: axisymmetric</td>
</tr>
<tr>
<td>II</td>
<td>0: mixed hybrid approach, 1: displacement based approach</td>
</tr>
<tr>
<td>ρ</td>
<td>$[F/L^3]/[L/T^2]$ density $\rho = \gamma/g$</td>
</tr>
<tr>
<td>K</td>
<td>$[F/L^2]$ compression modulus $K = \frac{E}{3(1-2\nu)}$</td>
</tr>
<tr>
<td>G</td>
<td>$[F/L^2]$ shear modulus $G = \frac{E}{2(1+\nu)}$</td>
</tr>
</tbody>
</table>

- Displacements (u,v) and stresses

- Output of stresses
Stresses are calculated at the element center in global directions 1(x) and 2(y).

- Plot of stresses
Stresses are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress plotted</td>
<td>$\sigma_{xx}$</td>
<td>$\sigma_{xy}$</td>
<td>$\sigma_{yy}$</td>
<td>$\sigma_{zz}$</td>
<td>$\sigma_{??}$</td>
</tr>
</tbody>
</table>
6.4.27 ELMT27

6.4.28 ELMT28

6.4.29 ELMT29
6.4.30 ELMT30

- Theory

ELMT30 is a general 4/8/9 - node flat element to set surface loads e.g. on solid elements in arbitrary directions. Furthermore displacement depending (follower) loads are possible.

\[ \text{ndm}=3, \text{ndf}=3, \text{nel}=4/8/9 \]

- Material input data

- Loads via macro \texttt{qloa}.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{ma}</td>
<td>Material number in Input file</td>
</tr>
<tr>
<td>\textit{qx}</td>
<td>uniform normal load in x-direction</td>
</tr>
<tr>
<td>\textit{qy}</td>
<td>uniform normal load in y-direction</td>
</tr>
<tr>
<td>\textit{qz}</td>
<td>uniform normal load in z-direction</td>
</tr>
<tr>
<td>\textit{ltyp}</td>
<td>0=loads in global directions, 1=loads in local directions, 2=follower loads (only local \textit{qz})</td>
</tr>
</tbody>
</table>
6.4.31 ELMT31

6.4.32 ELMT32
6.4.33 ELMT33

\[ \begin{align*}
E, \nu, h, q, \rho, r_w, r_\beta
\end{align*} \]

- **Theory**
ELMT33 is a general 4-node thin plate bending element based on the Reissner–Mindlin Theory with uniform reduced integration (URI) and stabilization matrix (Belytschko, Tsay: IJNME 19(83)405-419). Be careful, this element may lead to hourglass-modes for thin plates. [ndm=3, ndf=3, nel=4]

- **Material input data**

<table>
<thead>
<tr>
<th>***</th>
<th>Formula</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E )</td>
<td>([F/L^2])</td>
<td>Modulus of elasticity</td>
</tr>
<tr>
<td>( \nu )</td>
<td>[-]</td>
<td>Poisson ratio</td>
</tr>
<tr>
<td>( h )</td>
<td>([L])</td>
<td>Plate thickness</td>
</tr>
<tr>
<td>( q )</td>
<td>([F/L^2])</td>
<td>Uniform normal load in z-dir.</td>
</tr>
<tr>
<td>( \rho )</td>
<td>([F/L^3]/[L/T^2])</td>
<td>density: ( \rho = \gamma/g )</td>
</tr>
<tr>
<td>( r_w )</td>
<td>[-]</td>
<td>stabilization parameter for w-terms</td>
</tr>
<tr>
<td>( r_\beta )</td>
<td>[-]</td>
<td>stabilization parameter for ( \beta )-terms</td>
</tr>
</tbody>
</table>

- **Loads via macro \textit{qloa}.**

| \( ma \) | [-] | Material number in Inputfile |
| \( q \) | \([F/L^2]\) | uniform normal load in z-direction |

- **Definition of stress resultants**
\[
\begin{align*}
    m_{xx} & = \int \sigma_{xx} z \, dz, \\
    m_{yy} & = \int \sigma_{yy} z \, dz, \\
    m_{xy} & = \int \sigma_{xy} z \, dz = m_{yx}, \\
    q_{xx} & = \int \tau_{xx} \, dz, \\
    q_{yz} & = \int \tau_{yz} \, dz
\end{align*}
\]

- **Displacements \((w, \varphi_x, \varphi_y)\) and moments per length**

Shear forces show on positive side in positive direction.

- **Output of moments/shear forces**
The moments/shear forces per unit length are calculated at each center of element in global directions \( x \) and \( y \). The moments \( m_1 \) and \( m_2 \) are the principal moments with the angle \( \varphi_1 \) between 1–moment and \( x \)-coordinate direction (in degrees).

- **Plot of moments/shear forces**
The moments/shear forces per unit length are plotted due to following numbers.
<table>
<thead>
<tr>
<th>Stress number</th>
<th>Type</th>
<th>value plotted</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Moment</td>
<td>$m_{xx}$</td>
<td>[FL/L]</td>
</tr>
<tr>
<td>2</td>
<td>Moment</td>
<td>$m_{xy}$</td>
<td>[FL/L]</td>
</tr>
<tr>
<td>3</td>
<td>Moment</td>
<td>$m_{yy}$</td>
<td>[FL/L]</td>
</tr>
<tr>
<td>5</td>
<td>Moment</td>
<td>$m_1$</td>
<td>[FL/L]</td>
</tr>
<tr>
<td>6</td>
<td>Moment</td>
<td>$m_2$</td>
<td>[FL/L]</td>
</tr>
<tr>
<td>7</td>
<td>Angle</td>
<td>$\varphi_1$</td>
<td>[rad]</td>
</tr>
<tr>
<td>8</td>
<td>Shear force</td>
<td>$q_{xz}$</td>
<td>[F/L]</td>
</tr>
<tr>
<td>9</td>
<td>Shear force</td>
<td>$q_{gz}$</td>
<td>[F/L]</td>
</tr>
</tbody>
</table>
6.4.34  ELMT34

6.4.35  ELMT35

6.4.36  ELMT36

6.4.37  ELMT37

6.4.38  ELMT38

6.4.39  ELMT39
6.4.40  ELMT40

6.4.41  ELMT41

6.4.42  ELMT42

6.4.43  ELMT43

6.4.44  ELMT44
### Theory

**ELMT45** is a 8–node geometrical linear/nonlinear 3D-Solid Shell Element. The definition of the nodes is similar to the macro *bloc* [ndm=3, ndf=3, nel=8].

#### Material input data

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>matn</td>
<td>material number</td>
</tr>
<tr>
<td>nlay</td>
<td>number of layers (def.=1)</td>
</tr>
<tr>
<td>lin</td>
<td>0: geometrical linear, 1: geometrical nonlinear</td>
</tr>
<tr>
<td>ieas1</td>
<td>Enhanced Assumed Strain terms 0/3/5/8/11/7/30</td>
</tr>
<tr>
<td>ibd</td>
<td>Bathe-Dvorkin 0/1</td>
</tr>
<tr>
<td>ibs</td>
<td>Betsch-Stein 0/1</td>
</tr>
<tr>
<td>ies2</td>
<td>1=reg. mesh, 2= distorted mesh</td>
</tr>
<tr>
<td>h</td>
<td>shell thickness=thickness of 1 Element</td>
</tr>
<tr>
<td>ngp</td>
<td>no. of Gauss Points/layer (def.=2)</td>
</tr>
<tr>
<td>scf</td>
<td>shear corr.factor: $&lt;0 =</td>
</tr>
<tr>
<td>$q_x$</td>
<td>volumetric load in global x-direction</td>
</tr>
<tr>
<td>$q_y$</td>
<td>volumetric load in global y-direction</td>
</tr>
<tr>
<td>$q_z$</td>
<td>volumetric load in global z-direction</td>
</tr>
<tr>
<td>$\Delta T$</td>
<td>temperature difference</td>
</tr>
<tr>
<td>$\alpha_t$</td>
<td>coefficient of thermal expansion</td>
</tr>
<tr>
<td>$\rho$</td>
<td>density $\rho = \gamma/g$</td>
</tr>
</tbody>
</table>

Different material models are implemented and can be used with the 3D-Material library.

- For $\text{matn} = k$ material model $k$ is used for all layers.
  
  Material input data, see the 3D-Material library

- For $\text{matn} = 0$ different material models can be assigned to each layer.

  Sequence of material models for all layers beginning at $z=-h/2$

  $\text{matn}_1, \text{matn}_2, \text{matn}_3, ..., \text{matn}_i, ..., \text{matn}_{\text{nlay}}$

  Input data for each layer $i$ (on separate input line) beginning at $z=-h/2$

  Material input data, see the 3D-Material library

Input of layer orientation and thickness has to be done for each layer in any case.

Input data for each layer $i$ (on separate input line) beginning at $z=-h/2$

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_i$</td>
<td>[$^\circ$] angle between global and local coordinate system</td>
</tr>
<tr>
<td>$h_i$</td>
<td>[$L$] thickness of layer</td>
</tr>
</tbody>
</table>

Details on the derivation of the material models can be found in the Theory Manual.
• Loads via macro $qloa$.

<table>
<thead>
<tr>
<th>ma</th>
<th>$[-]$</th>
<th>Material number in Inputfile</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_x$</td>
<td>$[F/L^2]$</td>
<td>load in global x-direction</td>
</tr>
<tr>
<td>$q_y$</td>
<td>$[F/L^2]$</td>
<td>load in global y-direction</td>
</tr>
<tr>
<td>$q_z$</td>
<td>$[F/L^2]$</td>
<td>load in global z-direction</td>
</tr>
</tbody>
</table>

• Displacements and stress/strain resultants

$$
S = \begin{bmatrix}
S_{11} \\
S_{22} \\
S_{33} \\
S_{12} \\
S_{13} \\
S_{23}
\end{bmatrix}
E = \begin{bmatrix}
E_{11} \\
E_{22} \\
E_{33} \\
2E_{12} \\
2E_{13} \\
2E_{23}
\end{bmatrix}
u = \begin{bmatrix}
u_1 \\
u_2 \\
u_3
\end{bmatrix}
$$

• Output of stresses

Normal and shear stresses are calculated at the Gauss points of the element in global directions.

• Plot of stresses

The above defined stresses are plotted due to following numbers

<table>
<thead>
<tr>
<th>Stress number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress plotted $[F/L^2]$</td>
<td>$S_{11}$</td>
<td>$S_{22}$</td>
<td>$S_{33}$</td>
<td>$S_{12}$</td>
<td>$S_{13}$</td>
<td>$S_{23}$</td>
</tr>
</tbody>
</table>
6.4.46  ELMT46

6.4.47  ELMT47

6.4.48  ELMT48
Chapter 7

Principal example

A principal example with extended documentation

- Problem

\[ F = 10 \]
\[ E = 100 \, 000 \]
\[ \nu = 0.3 \]
\[ t = 1.2 \]

- General Description using \texttt{FEAP}

The problem is modeled as plane stress problem using element Type 5 with 35 nodes and 24 elements. Thus 4-node quadrilaterals in an x,y coordinate system with 2-displacements at each node (u,v components in the x,y directions are used. Element Type Number 1 is used. The left end of the mesh is on vertical rollers except for the center point which is fully fixed.

The data required to analyze the problem using \texttt{FEAP} is given by the following sets.

- \texttt{Input File}

  \[ \texttt{firstrecord} : \begin{array}{l}
  1 > \texttt{feap * * example 1. Plane stress beam} \\
  2 > 35,24,1,2,2,4 \\
  \end{array} \]

  In the above record 1 is used to describe the start of a new problem. The first four characters must be \texttt{feap}. 
second record:
35 = numnp number of nodes in the model
24 = numel number of elements in the model
1 = nummat number of material property sets in the model
2 = ndm spatial dimension of the finite element mesh
2 = ndf maximum number of unknowns (dof) at any node
4 = nen maximum number of nodes connected to any element.

1.) Block generation of nodes and elements.

1> bloc
2> 4,4,6,1,1,1 (no.block nodes,NR,NS,1st node,1st elmt,matl.set )
3> 1,0.,16. (block node#, x,y-coordinates)
4> 2,0.,0.
5> 3,36.,0.
6> 4,36.,16.

2.) Boundary Condition Generation.

* edge with constant coordinate

1> ebou
2> 1,0.,1,0 (direction , value, 1-dof b.c., 2-dof b.c.)

Direction interpreted as follows
1 = 1st coordinate direction (i.e., x-dir)
2 = 2nd coordinate direction (i.e., y-dir).

Boundary codes interpreted as follows:
0 = unrestrained dof
1 = restrained degree-of-freedom

* node with constrained values

1> boun
2> 3,,1,1 (node#, , 1-dof fixed, 2-dof, fixed)
3>

3.) Forced Conditions (non-zero nodal loads/displacements)

1> load
2> 31,,,-10. (node#, , 1-dof load, 2-dof load)
3>

N.B. A negative load is in a direction opposite to coordinate direction.

4.) Material Property Sets

1> mate
2> 1,5
3> 100000.,0.3,0,0,1 (E,\nu,\rho,1=plane stress)
4> 1.2 (Thickness)

N.B. The last item on record 3 is the problem type. A one (1) is plane stress,
a two (2) plane strain and a three (3) is the axisymmetrical case.
5.) End of Inputfile

1> end (end of input data)
2> inte (execution in interactive mode)
3> stop (end of file)

6.) Summary of Complete Input File for Mesh

1> feap * * example 1. Plane stress beam
2> 35,24,1,2,2,4
3>
4> bloc
5> 4,4,6,1,1,1
6> 1,0.,16.
7> 2,0.,0.
8> 3,36.,0.
9> 4,36.,16.
10>
11> ebou
12> 1,0.,1,0
13>
14> boun
15> 3,,1,1
16>
17> load
18> 31,,,,,-10.
19>
20> mate
21> 1,5
22> 100000,,0.3,0.0,1
23> 1.2
24>
25> end
26> inte
27> stop

• Macro Executions in interactive mode

1> feap start program
2> tang,,1 form tangent, residual, solve eq.
3> disp,all output all displacements
4> stre,node,1,35 compute and output stresses at nodes
5> stre,all compute and output stresses in elmts
6> reac,all compute and output all nodal reactions
7> exit end macro execution
Results using PLOT–macro commands

- **System**

- **deformed mesh**

- **stress \( \sigma_{11} \)**
Das Programm **FEAP-ED** dient dazu, die Dateneingabe für das Finite-Element-Programm **FEAP** zu vereinfachen. Das Programm kann auch als ein einfacher Editor verwendet werden. Alternativ kann die Dateneingabe mit einem beliebigen Windows–Editor erfolgen. Dies ist die Standardeinstellung und vorzuziehen.


Daten werden entweder direkt eingegeben oder mit den entsprechenden Menüs erzeugt. Der Datenaustausch zwischen FEAP-ED und dem Editor erfolgt über die Zwischenablage. CTRL+V kopiert die in der jeweiligen Maske erzeugten Daten in das (zu aktivierende) Editorfenster.

Chapter 9

F E A P - NEGE Mesher

Vorbemerkungen
Dies ist die Kurzbeschreibung des Netzgenerierungsprogramms NEGE von J. Sienz, University College, Department of Civil Engineering, University of Swansea.

Das Programm wurde derart modifiziert, dass die Ein– und Ausgabe auf das FE–Programm FEAP abgestimmt und das Programm unter Windows9x/NT/2000/ME/XP lauffähig ist.

Kurzbeschreibung

Bild 1 Beispiel einer Beschreibung eines Gebietes.
Das Finite-Element Netz wird von NEGE automatisch erzeugt. Die mittlere Länge \( L_e \) des einzelnen Elementes kann mit der EingabevARIABLEN esiz vorgegeben werden. Hierzu werden vom Programm die maximalen Koordinaten \( x_{\text{min}}, x_{\text{max}}, y_{\text{min}}, y_{\text{max}} \) ermittelt.

Aus dem einschließenden Rechteck \( (\Delta x, \Delta y) \) wird die mittlere Elementlänge \( L_e \) wie folgt ermittelt:

\[
L_e = \frac{\text{esiz}}{100} \cdot \Delta s \quad \text{mit} \quad \Delta s = \sqrt{\Delta x^2 + \Delta y^2}
\]

Wird kein Wert esiz eingegeben, so rechnet NEGE mit esiz=5. Damit ist die Elementlänge als bestimmter Prozentsatz der Diagonale des einschließenden Rechtecks zu verstehen. Somit entstehen in etwa gleichmäßig verteilte Netze.

Lokale Netzverdichtungen können erzeugt werden, wenn Daten für das Hintergrundnetz angegeben werden.

**Eingabebeschreibung**


1) **nege** Kommentartext (Kopfzeile)

   \`
   \textbf{ntyp, node, ndf, ncorr, esiz}
   \`

   \textbf{ntyp} = 1: erzeugt nur das Hintergrundnetz
   2: erzeugt 1 und die Segmenten
   3: erzeugt 2 und die Gebiete
   4: erzeugt 3 sowie die Netzgenerierung und die FEAP-Eingabedatei

Hinweise: Es wird empfohlen, die Dateneingabe und Generierung schrittweise vorzunehmen (ntyp = 1 – 4). Im Plot-Modus funktionieren jeweils die Optionen zu Knoten, Elementen, Randbedingungen und Lasten.
node = 3, 4, 6, 8, 9 : zu erzeugende Elementtypen
ndf = 1, 2, 3 : Anzahl der Freiheitsgrade
   (1 – z.B. Temperatur, 2 – z.B. Scheibe, 3 – z.B. Platte)
ncorr = 0 : Gebiet ist geradlinig berandet
   1 : Gebiet hat auch gekrümmte Ränder
esiz : Vorgabe der mittleren Elementlänge in % der Diagonale
des einschließenden Rechtecks, etwa 10–0.5 (grob–fein)

[ 2) neco (Eingabe der Konstanten) ]
   a = ......
   b = ......
......
Hier sind Konstanten zu definieren, die dann anschließend bei den weiteren Macros verwendet
werden können. Die Regeln sind analog zum Macro para. Die Konstanten werden in den FEAP–
Macros übernommen.

[ 3) back (Hintergrundnetz) ]
   ib, lnode(ib,1), lnode(ib,2), lnode(ib,3)
......
......
Leerzeile
   jb, corb(jb,1), corb(jb,2), d(jb,1), d(jb,2), d(jb,3), d(jb,4)
......
......
Es wird ein Hintergrundnetz aus Dreieckelementen definiert, mit dessen Hilfe die Netzdichte beliebig
vorgegeben werden kann. Werden hier Daten eingegeben, ist der Parameter esiz (Kopfzeile) außer
Kraft gesetzt.
   ib : Element–Nummer
   lnode(ib,i), i = 1,3 : Knoten des Elements (entgegen dem Uhrzeigersinn)
   jb : Knoten–Nummer
   corb(jb,i), i = 1,2 : Koordinaten des Knotens
d(jb,1) : gewünschte Elementlänge
d(jb,2) : gewünschte Dehnung
d(jb,3) : X–Wert des Richtungskosinus der Dehnungsrichtung
d(jb,4) : Y–Wert des Richtungskosinus der Dehnungsrichtung

4) geom (Koordinaten der Knotenpunkte)
   ig, corg(ig,1), corg(ig,2)
   ig : Knotennummer
   corg(ig,i), i = 1,2 : Koordinaten des Knotens
Es werden die Koordinaten der Eckpunkte des Gebietes eingegeben.

5) segm (Eingabe der Randsegmente)
   is, lnode(is,1), lnode(is,2), [ntyp, g(is,1), g(is,2), g(is,3), g(is,4)]
is : Elementnummer
   lnode(is,1), lnode(is,2) : Anfangsknoten, Endknoten
**nur bei gekrümmten Rändern**

<table>
<thead>
<tr>
<th>mtyp</th>
<th>Beschreibung</th>
<th>g(is,1)</th>
<th>g(is,2)</th>
<th>g(is,3)</th>
<th>g(is,4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Kreis (M-Punkt $P_x, P_y$, Radius $R$)</td>
<td>$P_x$</td>
<td>$P_y$</td>
<td>$R$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Ellipse (M-Punkt $P_x, P_y$, Radien $R_x, R_y$)</td>
<td>$P_x$</td>
<td>$P_y$</td>
<td>$R_x$</td>
<td>$R_y$</td>
</tr>
<tr>
<td>6*</td>
<td>Kreis (Punkt $P_1, P_4$, Radius $R$)</td>
<td>$P_1$</td>
<td>$P_4$</td>
<td>$R$</td>
<td></td>
</tr>
</tbody>
</table>

Erläuterung siehe nächste Seite

Es werden die Segmente zur Beschreibung des Randes definiert. Die Wahl von **Anfangs-** und **Endknoten** definiert die ’**Richtung**’ des Segments.

6) **regi** (Beschreibung des Gebietes)

ir, mr, lnode(ir,i), i = 1,mr

<table>
<thead>
<tr>
<th>ir</th>
<th>Nummer des Gebietes</th>
</tr>
</thead>
<tbody>
<tr>
<td>mr</td>
<td>Anzahl der Segmente, die dieses Gebiet beschreiben</td>
</tr>
<tr>
<td>lnode(ir,i), i=1,mr</td>
<td>Segmente des Gebietes in umlaufender Reihenfolge</td>
</tr>
</tbody>
</table>


7) **pres** (Eingabe von Randlasten)

<table>
<thead>
<tr>
<th>ip</th>
<th>belastetes Randsegment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip</td>
<td>Lasten auf dem Anfangsknoten</td>
</tr>
<tr>
<td>ip</td>
<td>Lasten auf dem Endknoten</td>
</tr>
</tbody>
</table>


8) **fixe** (Eingabe von Randbedingungen)

<table>
<thead>
<tr>
<th>if</th>
<th>Randsegment mit Randbedingungen</th>
</tr>
</thead>
<tbody>
<tr>
<td>bc(1)</td>
<td>Randbedingung für Freiheitsgrad 1</td>
</tr>
<tr>
<td>bc(2)</td>
<td>Randbedingung für Freiheitsgrad 2</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

bc(i)=0: frei, bc(i)=1: gelagert, nur bc(1)=-2: Linie auf der Knoten liegen müssen.

9) **FEAP–Eingabedaten**

Anmerkungen:

- **nege** generiert Netze für ebene Scheiben- und Plattenprobleme mit $ndf \leq 3$ Ebene Schalen mit $ndf = 5,6$ können generiert werden. Hier sind allerdings die zu den Freiheitsgraden $> 3$ zugehörigen Randbedingungen und Lasten mit **FEAP**–Macros zu erzeugen.

- Mit mtyp = 6 können kreisförmige Ausrundungen sehr leicht erzeugt werden, da das Programm sich die Koordinaten der zunächst unbekannten Knoten 2 und 3 selbst ermittelt. Hierzu erhalten diese Knoten zunächst in **geom** die Koordinaten des Schnittpunktes S der beiden angrenzenden Segmente. In **segm** müssen dann der Anfangsknoten des vorhergehenden Segmentes, der Endknoten des nachfolgenden Segmentes und der Radius der Ausrundung angegeben werden. Dies ist im folgenden Beispiel dokumentiert. Die zugehörige Eingabe lautet:

```
geom
1  x_1  y_1
2  x_S  y_S
3  x_S  y_S
4  x_4  y_4
```

```
segm
1  1  2
2  2  3  6  1  4  R_2
3  3  4
```

Eingabe

Erzeugte Ausrundung
10) Beispiele

Beispiel 1

NEGE–Datensatz

nege Beispiel 1
4, 4, 2, 0, 3.50

neco ns: Konstante fuer NEGE 2
a=10
b=5

gem etrie: Knoten zur Beschreibung des Randes 4
1,0,0
2,a,0
3,a,b
4,0,b

segm ente: Elemente zur Beschreibung des Randes 5
1,1,2
2,2,3
3,3,4
4,4,1

regi on: Beschreibung eines Gebietes 6
1,4,1,2,3,4

mate
1,5
1,,,,,
,,,,,

end
opti
inte
stop

4–Knoten–Netz mit 354 Elementen und 395 Knoten
Beispiel 2

NEGE-Datensatz

```plaintext
nege Beispiel 2
   4, 4, 2, 0, 3.50
neco ns: Konstante fuer NEGE 2
   a=10
   b=5
   c=5
   d=3
   e=2
   f=2
geom etrie: Knoten zur Beschreibung des Randes 4
   1,0,0
   2,a,0
   3,a,b
   4,0,b
   5,c,e
   6,c+d,e
   7,c+d,e+f
   8,c,e+f
segm ente: Elemente zur Beschreibung des Randes 5
   1,1,2
   2,2,3
   3,3,4
   4,4,1
   5,6,5
   6,5,8
   7,8,7
   8,7,6
regi on: Beschreibung eines Gebietes 6
   1,8,1,2,3,4,5,6,7,8
mate
   1,5
   1,,,,,,
   ,,","","","","",
end
opti
inte
stop
```

4-Knoten-Netz mit 278 Elementen und 332 Knoten
Beispiel 3

NEGE-Datensatz

nege Beispiel 3
    4, 3, 2, 1, 3.50

neco ns: Konstante fuer NEGE 2
    r=10

geom etrie: Knoten zur Beschreibung des Randes 4
    1,0,0
    2,r,0
    3,0,r

segm ente: Elemente zur Beschreibung des Randes 5
    1,1,2
    2,2,3,2,0,0,r
    3,3,1

regi on: Beschreibung eines Gebietes 6
    1,3,1,2,3

mate
    1,5
    1,,,,,,
    ,,,,,
end
opti
inte
stop

3-Knoten-Netz mit 665 Elementen und 369 Knoten
Beispiel 4

NEGE-Datensatz

nege Beispiel 4
4, 4, 2, 0, 3.50

neco ns: Konstante fuer NEGE 2
a=20
b=10
c=5
d=10
e=5

geom etrie: Knoten zur Beschreibung des Randes 4
1,0,0
2,c,0
3,c,e
4,a,0
5,a-d,e
6,a,e
7,a,b
8,0,b

segm ente: Elemente zur Beschreibung des Randes 5
1,1,2
2,2,3
3,2,4
4,4,6
5,5,6
6,6,7
7,7,8
8,8,1

regi on: Beschreibung eines Gebietes 6
1,10,1,3,4,6,7,8,2,-2,5,-5

fixe d: Randbedingungen auf dem Rand 8
2,-2
2,1,0
5,-2
5,0,1

mate
1,5
1,,,,,,

end
opti
Beispiel 5

NEGE–Datensatz

nege Beispiel 5 (Scheibe mit Loch)
4, 4, 2, 1, 5

neco ns: Konstante fuer NEGE 2
a=5
r=1
p=10

background: Hintergrundnetz 3
1,1,2,3,
2,2,4,3,
1,0,0,0.05,1,1,0,
2,a,0,0.50,1,1,0,
3,0,a,0.50,1,1,0,
4,a,a,1.00,1,1,0,

geometry: Knoten zur Beschreibung des Randes 4
1,r,0
2,a,0
3,a,a
4,0,a
5,0,r

segment: Elemente zur Beschreibung des Randes 5
1,1,2
2,2,3
3,3,4
4,4,5
5,5,1,2,0,0,r

region: Beschreibung eines Gebietes 6

4–Knoten–Netz mit 320 Elementen und 363 Knoten
4–Knoten–Netz mit 154 Elementen und 181 Knoten

Beispiel 6

NEGE–Datensatz

nege Beispiel 6 1/4 HEA 300
4, 4, 1, 1, 2

t=1.4
r=2.7

geometrie: Knoten zur Beschreibung des Randes 4
1, 0,0,
2, s/2,0,
3, s/2,h/2-r-t,
4, s/2+r,h/2-t,

segmente: Elemente zur Beschreibung des Randes 5
1, 1,2,0, 0, 0,0
2, 2,3,0, 0, 0,0
3, 3,4,2,s/2+r,h/2-t-r,r,0
4,4,5,0, 0, 0,0
5,5,6,0, 0, 0,0
6,6,7,0, 0, 0,0

Institut für Baustatik, KIT Cont. Mesh Macro Plot Mate3D Elem Nege Gmesh Cylt Isogeo 450
7,7,1,0, 0, 0,0,0

region: Beschreibung eines Gebietes 6
1,7,1,2,3,4,5,6,7

fixed: Randbedingungen auf dem Rand 8
1,1
7,1

mate
1,12
5,,3

solv
2

end

inte
stop

4-Knoten-Netz mit 210 Elementen und 283 Knoten
Chapter 10

F E A P - GMESH Mesher

Vorbemerkungen

Eingabebeschreibung

1) gmesh Kommentartext (Kopfzeile)

node, ndm, ndf, iopt, ityp

node = 3, 4, 8, 9 : zu erzeugende Elementtypen
ndm = 2,3 : Anzahl der Dimensionen
ndf = 1-6 : Anzahl der Freiheitsgrade
[iopt] = 0,1 : ohne/mit Knotennummeroptimierung
[ityp] = 0,1 : normale Generierung/Kontrolle der Eingabe
[iprin] = 0,1 : ohne/mit Ausdruck von Zwischenergebnissen

[ 2) neco (Eingabe der Konstanten) ]

a = ......
b = ......
.......
Hier sind Konstanten zu definieren, die dann anschließend bei den weiteren Macros verwendet werden können. Die Regeln sind analog zum Macro `para`. Die Konstanten werden von FEAP übernommen! Eine Neudefinition unter `para` ist nicht erforderlich!

3) `gele` (Eingabe der Eingabeblöcke mit 4–8 Knoten)

\begin{verbatim}
is, matno(is), lnode(is,1),..., lnode(is,8)
is : Blocknummer
matno(is) : Materialnummer des Blockes
lnode(is,1), ...lnode(is,8) : Knotennummern der Blöcke
\end{verbatim}

Es sind 4–8 Knoten einzugeben (Eckknoten 1-4 zwingend, Seitenmittenknoten 5-8 optional) Die Reihenfolge der Knotennummern ist analog der FEAP-Eingabe, entgegen dem Uhrzeigersinn.

4) `gcor` (Koordinaten der Knotenpunkte)

\begin{verbatim}
ig, corg(ig,1), corg(ig,2), [corg(ig,3)], [ityp]
ig : Knotennummer
corg(ig,i), i = 1,2(3) : Koordinaten des Knotens
ityp : 0=cart., 1=pola(1-2), 2=pola(2-3), 3=pola(1-3), 4=sphe
\end{verbatim}

Für die eingeführten Zwischenknoten müssen die Koordinaten eingegeben werden. Ihre Position kann im Rahmen des isoparametrischen Konzeptes (siehe Macro `bloc`) gewählt werden.

5) `ndvi` (Unterteilung der Blöcke)

\begin{verbatim}
is, ndvix, [dx(i), i = 1,ndvix]
is, ndviy, [dy(i), i = 1,ndviy]
is : Blocknummer
ndvix, ndviy : Anzahl der Unterteilungen in x,y–Richtung
dx(i), dy(i) : Unterteilungen je Richtung
\end{verbatim}

Annmerkungen:

# An Zwischengrenzen der Blöcke ist darauf zu achten, dass die Unterteilungen zusammenpassen.
# Die Eingabe der Unterteilungen ist nur bei unterschiedlichen Werten notwendig.
# Es können beliebige Werte eingegeben werden. Die Verteilung ergibt sich aus dem Verhältnis der Eingabewerte.
# Je Zeile dürfen maximal 16 Werte eingeben werden. Darüberhinaus erfolgt die Eingabe in Folgezeilen.

6) `feap`–Eingabedaten

7) Beispiele

Beispiel 1 im Raum befindliche gekrümme Fläche
Isoparametrische Ansatzfunktion \( N_1 \) auf einem Quadrat 5x5 durch Vorgabe der Seitenmittenknoten.

GMESH–Datensatz

gmesh Isoparametrische Ansatzfunktion \( N_1 \)
4,3,2,0,0

gene
1,1,1,2,3,4,5,0,0,6

gcor Koordinaten, \((5+6=Seitenmittenknoten)\)
1,0,0,h
2,a,0,0
3,a,a,0
4,0,a,0
5,a/2,0,0
6,0,a/2,0

ndvi Teilung 10*10 Elemente
1,10
1,10

end

inte
stop

Beispiel 2 ebene Fläche mit Loch

GMESH–Datensatz

gmesh Scheibe mit Kreis-Oeffnung Dreieckelemente
3,2,2,0,0

eco
m=4
k=5
n=6
l=7
gele
1,1,1,2,4,3
2,1,3,4,8,6,0,0,7
3,1,3,6,11,14,0,9
4,1,8,4,15,13,0,0,10
5,1,11,13,15,14,12
6,1,4,5,16,15

gcor
1,0,0
2,5,0
3,0,5
4,5,5
5,8,5
6,1.4393,6.4393
7,2.5,6
8,3.5607,6.4393
9,1,7.5
10,4,7.5
11,1.4393,8.5607
12,2,5,9
13,3.5607,8.5607
14,0,10
15,5,10
16,8,10

da
1,1
1,k
2,1
2,m
3,m
3,n
4,n
4,m
5,n
5,m
6,k
6,n

dmv
1,l

end
inte
stop
Beispiel 3 Zweifeldplatte auf Wand mit Punktstützung

GMESH–Datensatz

gmesh Zweifeld-Platte auf punktgestützter Wand
4,3,6

neco Konstanten
a=8
b=16
h=2

gene Elemente
1,1,7,8,5,2
2,2,1,3,6,4

gcor Koordinaten
1, -a,0, 0
t2, 0,0, 0
t3, a,0, 0
t4, -a,b, 0
t5, 0,b, 0
t6, a,b, 0
t7, 0,0,-h
t8, 0,b,-h

gcor Koordinaten
1, -a,1,1,1
1, a,1,1,1
1,0,-h
0,0,0
1,1,1

gcor Koordinaten
0,b,-h
0,0,0
1,1,1

gcor Koordinaten
0,b,-h
0,0,0
1,1,1

gcor Koordinaten
0,b,-h
0,0,0
1,1,1

gcor Koordinaten
0,b,-h
0,0,0
1,1,1

ndvi Einteilung
1,12
1,5
2,12
2,12

end inte stop

Eingabennetz generiertes Netz
Beispiel 4 1/2 Doppel-T-Träger mit Löchern und seitlichem Anschluß

GMESH-Datensatz

gmesh I-Traeger mit Loechern
4,3,1,0,0

neco
i=4
k=6
m=8
n=12

gele
1, 1,30, 2,24,32
2, 2, 5, 6,28,27
3, 3, 1, 2,30,29
4, 4, 3, 9, 7, 0, 0, 8
5, 5, 3, 7,17,25, 0,13
6, 6, 9, 4,26,19, 0, 0,14
7, 7,17,19,26,25,18
8, 8, 4, 5,12,10, 0, 0,11
9, 9, 4,10,20,26, 0,15
10,10,12, 5,27,22, 0, 0, 0,16
11,11,20,22,27,26,21
12,12,23,24,32,31

gcor
1, 0,-8,-15.5 1,i
2, 94,-8,-15.5 1,m
3, 0, 0,-15.5 2,1
4, 47, 0,-15.5 2,m
5, 94, 0,-15.5 3,2*n
6, 128.5, 0,-15.5
7,17.964, 0,-3.536 3,i
8, 21.5, 0,-5 4,n
9,25.036, 0,-3.536 4,k
10,69.964, 0,-3.536 5,k
11, 73.5, 0,-5 5,m
12,77.036, 0,-3.536 6,k
13, 16.5, 0, 0 6,m*1.5
14, 26.5, 0, 0
15, 68.5, 0, 0
16, 78.5, 0, 0
17,17.964, 0, 3.536
18, 21.5, 0, 5
19,25.036, 0, 3.536
20,69.964, 0, 3.536
21, 73.5, 0, 5
22,77.036, 0, 3.536
23, 0,-8, 15.5

24, 94,-8, 15.5 12,2*n
25, 0, 0, 15.5
26, 47, 0, 15.5 12,i
27, 94, 0, 15.5
28, 128.5, 0, 15.5
29, 0, 8,-15.5
30, 94, 8,-15.5
31, 0, 8, 15.5
32, 94, 8, 15.5

end
inte
stop
Chapter 11

FEAP - ISOGEO

11.1 Overview

Input files for isogeometric FEAP - ISOGEO calculations have the same format as input files for standard FEAP calculations. For the definition of the geometry the three additional macros nmpq, knv1 and knv2 are introduced. In Isogeometric Analysis the shape functions from the NURBS geometry are used. The usage of the exact NURBS geometry requires a distinct form of input. Necessary changes include a slightly different use of macros feap and coor, the definition of new macros nmpq, knv1 and knv2. As the mesh is predefined by the NURBS geometry description, meshing macros like elem or bloc are neither needed nor allowed. All other feap mesh macros can be used as usual. tie can be used to connect patches if the control points coincide. Do not use opti, usage of PARDISO-solver is recommended. The input macros are explained in the mesh-part of this manual. Macro commands and plot commands work as in standard FEAP.

11.2 Terminology

NURBS surfaces are a tensor product area spanned by two knot vectors \( \Xi^1 = [\xi_1^1, \xi_2^1, \ldots, \xi_{n+p+1}^1] \) and \( \Xi^2 = [\xi_1^2, \xi_2^2, \ldots, \xi_{n+p+1}^2] \). The physical position of a point with the parameters \( \xi^1 \in \Xi^1 \) and \( \xi^2 \in \Xi^2 \) can be computed with the basis functions and the control points \( B \). The basis functions are computed from the knot vectors for every distinct set of parameters \( \xi^1 \) and \( \xi^2 \). So necessary information for a NURBS patch include the two knot vectors \( \Xi^1 \) and \( \Xi^2 \), the control points \( B \), the order of basis functions \( p \) in direction of \( \Xi^1 \), \( q \) in direction of \( \Xi^2 \) and the number of control points \( n \) in direction of \( \Xi^1 \) and \( m \) in direction of \( \Xi^2 \). NURBS geometries may consist of several patches. Every patch has its own control points and constants described above. Common control points of patches have to be given separately, as the sequence of control points may not be disturbed.

11.3 Known Errors

The plot macros disp and base do not work. Please report further bugs to wolfgang.dornisch@bauing.uni-kl.de.
Chapter 12

F E A P – CYLT Mesh Generator for Yield-Line Predictions

1. Preliminaries

This is a short instruction for the yield-line mesh prediction program CYLT – Computational Yield-Line Theory by J. Wüst, Institut für Baustatik, Universität Karlsruhe.

The program is adjusted to the finite element program FEAP being executable by Windows9x/NT/2000/ME/XP where further steps of yield-line calculation and optimization are performed.

2. Description

CYLT is a 2–dimensional mesh generator for yield-line prediction. Depending on the type option it shows the yield-line pattern or corresponding triangulation steps, see Section 3 or cylt command.

The generation is achieved by plane rotations where the edges of the considered plate define the rotation axes. Ruled by the yedg command the rotation angles will be defined to 45° in case of a supported edge or to 90° in case of a free edge. The intersection lines between the correspondingly deflected planes form potential yield-lines, see Fig. 1, and the intersection points of three plates yield potential branching points of yield-lines. In this way a possible failure geometry is found which can be optimized by a later optimization calculation performed by FEAP.

Fig. 1: Plane intersections forming yield-lines
The prediction algorithm detects all admissible configurations of yield-line patterns which will be numbered. In a further step the user can choose between one or more configurations depending on the given plate geometry. For internal reasons it may be possible that one yield-line pattern will be represented by several configuration numbers, see Section 4.

In order to perform a calculation with the triangular yield-line element in FEAP it is necessary to triangulate the yield-line pattern as mentioned in the beginning of this abstract. The configuration numbers will be the same for all type options.

3. General Input

The input file is divided into two parts. The first part rules the CYLT yield-line mesh generation, the second one is needed by FEAP to perform a calculation using the corresponding triangular yield-line element. Its general form reads as follows:

| CYLT <problem name> <type> | • header with optional problem name  
|                           |   • triangulation type option  
| ycon                      | • optional parameter definition  
| ynod <node_i>, <x>, <y>   | • nodal coordinates of the plate  
| yedg <edge_i>, <node_i>, <node_j>, <angle> | • edge definition  
| ...                       |                                                     
| mate                      | • material properties of YLT-element  
|                           |   • including $m_{pl}$ of area  
| edge                      | • definition of b.c. along line in FEAP  
|                           | (other FEAP commands are also possible)  
| ybou                      | • boundary conditions of YLT-element  
| ...                       |   • including $m_{pl}$ of edges  
| solv 11                   | • switch to simplex optimization solver |

3.1. The CYLT Part

The header CYLT introduces the mesh generation which is directed by the <type> option in the following line.

- <type> = 0: prediction of yield-line pattern(s).

It is recommended to check the predicted yield-line pattern at the beginning. Note: this type of mesh generation cannot be used for a calculation.
• \(<\text{type}> = i \text{ (for } i > 0)\): triangulation step.

For a calculation with the triangular yield-line element in \textit{FEAP} it is necessary to use \(<\text{type}> = 1\).

Polygonal plate areas (see Fig. 1) will be divided into triangles. Higher triangulation steps are optional to other programs; the successively subdivided mesh structure is shown in Fig. 2.

Fig. 2: Triangulation steps defined by \(<\text{type}> = 1, 2 \text{ and } 3\)

With the optional \texttt{ycon} command parameters can be defined similar to \texttt{para} in \textit{FEAP}.

The nodes of the plate border have to be input counterclockwise together with their coordinates using the \texttt{ynod} command.

In the next step the edges will be given with their numbers and nodes introduced by the \texttt{yedg} command.

The last entry of each line defines the angle of the plane deflection in order to detect intersection lines, see Fig. 1. The convention is

• \(<\text{angle}> = 0\): free edge.
  By creating the yield-line pattern the corresponding planes will be deflected with 90°.

• \(<\text{angle}> = 1\): supported edge.
  The deflection angle will be 45°.

Note: this command rules the yield-line mesh generation with \textit{CYLT} whereas \texttt{ybou} specify the boundary conditions of edges in \textit{FEAP} (e.g. plastic moments, clamping, etc.). In \textit{CYLT} only the division into supported and non-supported –i.e. free– edges is needed.

\section*{3.2. The \textit{FEAP} Part}

The \textit{FEAP} part defines an element type with the \texttt{mate} command. In order to perform a yield-line optimization process the corresponding triangular yield-line element has to be chosen. The upper and lower plastic moments have to be given for both coordinate directions; in this way orthotropic reinforcement can be modelled.

To specify the boundary conditions of an edge –especially in order to define individual upper and lower plastic moments \(m_{pl,u}\) and \(m_{pl,l}\) – the \texttt{ybou} command has been implemented into \textit{FEAP}. This information contributes to the internal work calculation. In the subsequent part of geometrical optimization regular \textit{FEAP} macros defining boundary conditions like \texttt{poin} or mainly \texttt{edge} are used. In this way the search directions during the optimization process are ruled: all degrees of freedom within the plate plane have to be left free.

Note: symmetrical problems can be mapped using free edges with plastic moments.

For further details see the \texttt{mesh} manual or the examples of Section 5.

\section*{4. Remarks}

At the end of the prediction process the user has to select a yield-line mesh being submitted to \textit{FEAP}. The number of which has nothing in common with the optimal solution. It is strongly recommended to perform all by starting a new calculation.
Especially symmetrical geometries may produce several numbers representing identical configurations. The reason is that they are internally encoded differently, see Fig. 3. So the calculation results will be the same.

![Fig. 3: A quadratic plate signify the link between two rectangular plates](image)

5. Examples

The following examples show the usage of the CYLT mesh generator in context to the FEAP yield-line optimization process. The required macro commands will be found in the macro manual.

Note: Using the ygra command it may be possible that an initial step length is required which can be given by the command dt.

Example 5.1: Rectangular Plate

This simple yield-line geometry of a rectangular plate can be reproduced by hand. All edges are simply supported.

![Fig. 4: Example 5.1: Geometry and mesh solution for <type>=0 and <type>=1](image)

Notes:

- Defining different plastic moments at the boundaries (ybou command) will produce a non-symmetrical solution during the optimization process in FEAP. The mesh generation performed by CYLT does not depend on material parameters, clamping moments, etc.
- Contemporarily the number of the yield-line element is 40, the simplex optimization solver has the number 11.
CYLT  Rectangular Plate
1

ycon
a=3.8
b=5.2

ynod
1, 0, 0
2, a, 0
3, a, b
4, 0, b

yedg
1,1,2,1
2,2,3,1
3,3,4,1
4,4,1,1

cons
a=3.8
b=5.2
m=6.558

dg:
0,0,,a,0
1,1,1
a,0,,a,b
1,1,1
0,b,,a,b
1,1,1
0,0,,0,b
1,1,1

mate
1, 40 <ylt-element-number>
1,m,m,m,m

ybou
0, 0,, a, 0,,1,0,0
a, 0,, a, b,,1,0,0
0, b,, a, b,,1,0,0
0, 0,, 0, b,,1,0,0

solv
11 <simplex optimization solver>

end
inte
stop
Example 5.2: Modelling of a Symmetrical Problem

With the definition of a free edge clamped by the same plastic moment as the plate area the precedent example can be modelled symmetrically. The following input data shows the differences compared to Example 5.1. Edge 4 between the nodes 4 and 1 will be the new symmetrical axis. The corresponding entry for the `edge` command (see first line) allows that node points –which will be searching points for the optimization process– move vertically and along the edge line; the perpendicular direction is fixed.

CYLT  Rectangular Plate (Symmetrical)

1

ycon
a=3.8
b=2.6

ynod
1, 0, 0
2, a, 0
3, a, b
4, 0, b

yedg
1,1,2,1
2,2,3,1
3,3,4,1
4,4,1,0

...  

edge
0,0,,a,0
1,0,0
a,0,,a,b
1,1,1
0,b,,a,b
1,1,1
0,0,,0,b
1,1,1

...  

ybou
0, 0, , a, 0,,1,m,m
a, 0, , a, b,,1,0,0
0, b, , a, b,,1,0,0
0, 0, , 0, b,,1,0,0

...
Example 5.3: Different Yield-line Configurations

An example of two different mesh solutions is shown. The following CYLT part of an input file produces the two yield-line configurations in Fig. 5.

cylt Example with two yield-line configurations

0

ycon
a=0.5
b=0.5
k=0.25

ynod
1, 0, 2*b,
2, k, b
3, k, 0
4, 2*a, 0
5, 2*a-k, b
6, 2*a-k, 2*b

yedg
1, 1, 2, 1
2, 2, 3, 1
3, 3, 4, 1
4, 4, 5, 1
5, 5, 6, 1
6, 6, 1, 1

Fig. 5: Example 5.3: One plate geometry with two yield-line configurations
Example 5.4: Yield-line Optimization Performed by FEAP

The yield-line element implemented in FEAP uses geometrical relations containing nodal displacements and edge rotations. Since three points define a plane it is a triangular element.

Two different calculation types can be employed for the yield-line calculation: the gradient optimization method or the direct search method. Beyond the CYLT mesh generator the following arbitrary example will show the differences – Fig. 6b displays the search directions respectively degrees of freedom.

Coordinates:

1 : [ 0.00 | 0.00 ]
2 : [ 0.00 | 18.00 ]
3 : [ 12.00 | 18.00 ]
4 : [ 12.00 | 0.00 ]
5 : [ 9.00 | 3.00 ]
6 : [ 9.00 | 15.00 ]
7 : [ 3.00 | 18.00 ]
8 : [ 12.00 | 13.50 ]
9 : [ 3.00 | 0.00 ]
10 : [ 3.00 | 14.00 ]

Direct Search Method

The direct search method is a simple approach of trial steps into the search directions. It is a very robust method but it is quite slow and takes increasing resources depending on the number of degrees of freedom. In every step the limit load has to be calculated in order to find the descend slope. The following macro commands have to be executed consecutively:

```
  ytab
  • calculate simplex tableau
  ytry
  • perform a searching step (trial)
```
Gradient Method

The gradient method moves along the slope of the objective function and is therefore very effective. It is necessary to define a step length which will be determined by FEAP. Otherwise it can be given manually by the dt command. Especially in the beginning of a calculation it may be necessary to define an initial step length. The procedure is as follows:

- ytab: calculate simplex tableau
- iyang: print yield-line rotation angles (optional)
- ygra: calculate gradient
- ymsh: determine step length and remesh

In this example FEAP calculates a zero step length within the first iteration step. So an initial value (e.g. here dt,1 has to be given. After the next turn of the presented procedure it should be removed by dt,0 – it will be calculated automatically again. The following iterations lead to the load minimum.

As the implemented optimization algorithm contains a sensitivity analysis, the method is able to detect a lower value than the direct search method. The reason is that the influence between the moves of the degrees of freedom is taken into account – the direct search method stacks at a local minimum.

Fig. 7 shows the different solutions, Fig. 8 gives an idea of the optimization function depending on several degrees of freedom.
Fig. 8: Optimization function
Chapter 13

F E A P - Beispielsammlung
13.1 Beispiele für 2D/3D - Fachwerkelement

13.1.1 Beispiel 1 - Dachbinder

13.1.1.1 System und Belastung

Querschnittswerte: Nadelholz, 10×10

\[ E = 1000 \text{ kN/cm}^2 \]

\[ A = 100,0 \text{ cm}^2 \]

gesucht: Schnittgrößen, Knotenverschiebungen

13.1.1.2 Eingabedatensatz (file: it01)

feap ebenes Fachwerk, 1. Beispiel: Dachbinder

```
4, 5, 1, 2, 2, 2, 0, 0
```

coor

```
1, 1, 0.000, 0.000
3, 0, 5.000, 0.000
4, 0, 2.500, 1.000
```

elem

```
1, 1, 1, 2, 1
4, 1, 1, 4
5, 1, 2, 4
```

boun

```
1, 0, 1, 1
3, 0, 0, 1
```

load

```
2, , 0.00, -40.00
4, , 30.00, 0.00
```

mate

```
1, 1
1000.d+04, 100.d-04, , NH 10 x 10
```

date

```
end
```

stop
13.1.1.3 Ergebnisse

Die Stabkräfte, Dehnungen und Längsspannungen ergeben sich unter der angegebenen Belastung zu:

**Truss Element**

<table>
<thead>
<tr>
<th>elem mate</th>
<th>1-coord</th>
<th>2-coord</th>
<th>3-coord</th>
<th>force</th>
<th>strain</th>
<th>stress</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.2500</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.65000E+02</td>
<td>0.65000E-03</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3.7500</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.65000E+02</td>
<td>0.65000E-03</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3.7500</td>
<td>0.5000</td>
<td>0.0000</td>
<td>-0.70007E+02</td>
<td>-0.70007E-03</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1.2500</td>
<td>0.5000</td>
<td>0.0000</td>
<td>-0.37696E+02</td>
<td>-0.37696E-03</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2.5000</td>
<td>0.5000</td>
<td>0.0000</td>
<td>0.40000E+02</td>
<td>0.40000E-03</td>
</tr>
</tbody>
</table>

Die Knotenverschiebungen sind:

**nodal displacements**

<table>
<thead>
<tr>
<th>node</th>
<th>1 coord</th>
<th>2 coord</th>
<th>1 displ</th>
<th>2 displ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>2</td>
<td>2.50000E+00</td>
<td>0.00000E+00</td>
<td>1.62500E+03</td>
<td>-8.36674E-03</td>
</tr>
<tr>
<td>3</td>
<td>5.00000E+00</td>
<td>0.00000E+00</td>
<td>3.25000E+03</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>4</td>
<td>2.50000E+00</td>
<td>1.00000E+00</td>
<td>2.09351E+03</td>
<td>-7.96674E-03</td>
</tr>
</tbody>
</table>

13.1.1.4 Vergleichslösung

Eine Handrechnung mit dem PVK ergibt als vertikale Verschiebung für den Knoten 2:

\[ \delta_v = \int \frac{NN}{EA} = 0.83667 \text{ cm} \]

13.1.1.5 verformte Struktur (20-fach überhöht)
13.1.2 Beispiel 2 - Kran

13.1.2.1 System und Belastung

Querschnittswerte: St 37, I 300

\begin{align*}
E &= 21000 \text{ kN/cm}^2 \\
I &= 9800 \text{ cm}^4 \\
A &= 69.0 \text{ cm}^2
\end{align*}

Belastung: wie angegeben in Skizze

gesucht: Schnittgrößen, Knotenverschiebungen

13.1.2.2 Eingabedatensatz (file: it02)

feap ebenes Fachwerk, 2. Beispiel: Kran

8, 13, 1, 2, 2, 0, 0

coor

\begin{align*}
1, 0, &-6.000, 0.000 \\
2, 0, &0.000, 0.000 \\
3, 0, &-2.000, 4.000 \\
4, 1, &0.000, 4.000 \\
6, 1, &10.000, 4.000 \\
8, 0, &0.000, 6.000
\end{align*}

elem

\begin{align*}
1, 1, 1, 2, 1 \\
8, 1, 5, 7, \\
9, 1, 4, 7, \\
10, 1, 4, 8, \\
11, 1, 3, 8, \\
12, 1, 2, 4, \\
13, 1, 1, 3,
\end{align*}

boun

\begin{align*}
1, 0, 0, 1 \\
2, 0, 1, 1
\end{align*}

load

\begin{align*}
3, , &0.00, -200.00 \\
5, , &0.00, -100.00 \\
6, , &0.00, -50.00 \\
8, , &75.00, 0.00
\end{align*}

mate

\begin{align*}
1, 1
\end{align*}
13.1.2.3 Ergebnisse

Die Stabkräfte, Dehnungen und Längsspannungen ergeben sich unter der angegebenen Belastung zu:

<table>
<thead>
<tr>
<th>elem mate</th>
<th>1-coord</th>
<th>2-coord</th>
<th>3-coord</th>
<th>force</th>
<th>strain</th>
<th>stress</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-3.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>-0.17500E+03</td>
<td>-0.12077E-03</td>
<td>-0.25362E+05</td>
</tr>
<tr>
<td>2</td>
<td>-1.0000</td>
<td>2.0000</td>
<td>0.0000</td>
<td>0.22361E+03</td>
<td>0.15432E-03</td>
<td>0.32407E+05</td>
</tr>
<tr>
<td>3</td>
<td>-1.0000</td>
<td>4.0000</td>
<td>0.0000</td>
<td>-0.25000E+03</td>
<td>-0.34507E-03</td>
<td>-0.72464E+05</td>
</tr>
<tr>
<td>4</td>
<td>2.5000</td>
<td>4.0000</td>
<td>0.0000</td>
<td>-0.17253E-03</td>
<td>-0.36232E+05</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7.5000</td>
<td>4.0000</td>
<td>0.0000</td>
<td>-0.25000E+03</td>
<td>-0.17253E-03</td>
<td>-0.36232E+05</td>
</tr>
<tr>
<td>6</td>
<td>7.5000</td>
<td>4.5000</td>
<td>0.0000</td>
<td>0.25495E+03</td>
<td>0.17595E-03</td>
<td>0.36949E+05</td>
</tr>
<tr>
<td>7</td>
<td>2.5000</td>
<td>5.5000</td>
<td>0.0000</td>
<td>0.50990E+03</td>
<td>0.35190E-03</td>
<td>0.73899E+05</td>
</tr>
<tr>
<td>8</td>
<td>5.0000</td>
<td>4.5000</td>
<td>0.0000</td>
<td>0.10000E+03</td>
<td>0.69013E-04</td>
<td>0.14493E+05</td>
</tr>
<tr>
<td>9</td>
<td>2.5000</td>
<td>4.5000</td>
<td>0.0000</td>
<td>-0.25495E+03</td>
<td>-0.17595E-03</td>
<td>-0.36949E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.0000</td>
<td>5.0000</td>
<td>0.0000</td>
<td>-0.67500E+03</td>
<td>-0.46584E-03</td>
<td>-0.97826E+05</td>
</tr>
<tr>
<td>11</td>
<td>-1.0000</td>
<td>5.0000</td>
<td>0.0000</td>
<td>0.81317E+03</td>
<td>0.56120E-03</td>
<td>0.11785E+06</td>
</tr>
<tr>
<td>12</td>
<td>0.0000</td>
<td>2.0000</td>
<td>0.0000</td>
<td>-0.72500E+03</td>
<td>-0.50035E-03</td>
<td>-0.10507E+06</td>
</tr>
<tr>
<td>13</td>
<td>-4.0000</td>
<td>2.0000</td>
<td>0.0000</td>
<td>-0.25495E+03</td>
<td>-0.17595E-03</td>
<td>-0.36949E+05</td>
</tr>
</tbody>
</table>

Die Knotenverschiebungen sind:

<table>
<thead>
<tr>
<th>node</th>
<th>1 coord</th>
<th>2 coord</th>
<th>1 displ</th>
<th>2 displ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-6.00000E+00</td>
<td>0.00000E+00</td>
<td>7.24638E-04</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>3</td>
<td>-2.00000E+00</td>
<td>4.00000E+00</td>
<td>8.79625E-04</td>
<td>1.21140E-03</td>
</tr>
<tr>
<td>4</td>
<td>0.00000E+00</td>
<td>4.00000E+00</td>
<td>1.89494E-04</td>
<td>2.00138E-03</td>
</tr>
<tr>
<td>5</td>
<td>5.00000E+00</td>
<td>4.00000E+00</td>
<td>-6.73170E-04</td>
<td>-2.70967E-02</td>
</tr>
<tr>
<td>6</td>
<td>1.00000E+01</td>
<td>4.00000E+00</td>
<td>-1.53583E-03</td>
<td>-6.06806E-02</td>
</tr>
<tr>
<td>7</td>
<td>5.00000E+00</td>
<td>5.00000E+00</td>
<td>4.27982E-03</td>
<td>-2.70277E-02</td>
</tr>
<tr>
<td>8</td>
<td>0.00000E+00</td>
<td>6.00000E+00</td>
<td>7.26887E-03</td>
<td>-2.93306E-03</td>
</tr>
</tbody>
</table>

13.1.2.4 Vergleichslösung

Die Stabkräfte können leicht über Knotengleichgewicht überprüft werden.

Die Verschiebungen können mit dem PVK errechnet werden.
13.1.2.5 verformte Struktur (20-fach überhöht)
13.1.3 Beispiel 3 - Brückenbogen

13.1.3.1 System und Belastung

Querschnittswerte:

- Obergurt: HEM 300
  \( I = 59200 \ cm^4 \)
  \( A = 303.0 \ cm^2 \)

- Untergurt: HEM 160
  \( I = 5100 \ cm^4 \)
  \( A = 97.1 \ cm^2 \)

- Pfosten: HEM 100
  \( I = 1140 \ cm^4 \)
  \( A = 53.2 \ cm^2 \)

- Diagonalen: HEM 140
  \( I = 3290 \ cm^4 \)
  \( A = 80.6 \ cm^2 \)

Belastung: \( P = 250 \ kN \)
wie in Skizze aufgebracht

gesucht: Schnittgrößen, Knotenverschiebungen

13.1.3.2 Eingabedatensatz (file: it03)

feap ebenes Fachwerk, 3. Beispiel: Brücke
12, 21, 4, 2, 2, 2, 0, 0

c l = Länge
c h = Höhe
c P = Last vertikal

cons
l= 5.d0
h= 5.d0
p= 250.d0

coor
1, 0, 0.000, 0.000
2, 0, 0.750*1, 0.000
3, 0, 1.750*1, 0.000
4, 0, 2.750*1, 0.000
5, 0, 3.750*1, 0.000
6, 0, 4.750*1, 0.000
7, 0, 5.500*1, 0.000
8, 0, 0.750*1, -1.000*h
9, 0, 1.750*1, -1.750*h
10, 0, 2.750*1, -2.000*h
11, 0, 3.750*1, -1.750*h
12, 0, 4.750*1, -1.000*h
elem
  1, 1, 1, 2, 1
  7, 2, 1, 8, 1
  8, 2, 8, 9, 1
  12, 2, 12, 7, 1
  13, 3, 2, 8, 1
  18, 4, 3, 8, 1
  19, 4, 3, 10, 1
  20, 4, 5, 10, 1
  21, 4, 5, 12, 1

boun
  1, 0, 1, 1
  7, 0, 0, 1

load
  2, , 0.00, -0.875*p
  3, 1, 0.00, -1.000*p
  5, , 0.00, -1.000*p
  6, , 0.00, -0.875*p

mate
  1, 1
    21000.d+04, 303.d-04, , HEM 300
  2, 1
    21000.d+04, 97.1d-04, , HEM 160
  3, 1
    21000.d+04, 53.2d-04, , HEM 100
  4, 1
    21000.d+04, 80.6d-04, , HEM 140

end

inte

stop

13.1.3.3 Ergebnisse

Die Stabkräfte können leicht über Knotengleichgewicht überprüft werden.
Die Verschiebungen können mit dem PVK errechnet werden.
Die Knotenverschiebungen sind:

<table>
<thead>
<tr>
<th>node</th>
<th>1 coord</th>
<th>2 coord</th>
<th>1 displ</th>
<th>2 displ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>2</td>
<td>3.75000E+00</td>
<td>0.00000E+00</td>
<td>-2.62443E-04</td>
<td>-6.24625E-03</td>
</tr>
<tr>
<td>3</td>
<td>8.75000E+00</td>
<td>0.00000E+00</td>
<td>-6.12366E-04</td>
<td>-8.08225E-03</td>
</tr>
<tr>
<td>4</td>
<td>1.37500E+01</td>
<td>0.00000E+00</td>
<td>-9.83776E-04</td>
<td>-1.05702E-02</td>
</tr>
<tr>
<td>5</td>
<td>1.87500E+01</td>
<td>0.00000E+00</td>
<td>-1.35519E-03</td>
<td>-8.08225E-03</td>
</tr>
<tr>
<td>6</td>
<td>2.37500E+01</td>
<td>0.00000E+00</td>
<td>-1.70511E-03</td>
<td>-6.24625E-03</td>
</tr>
<tr>
<td>7</td>
<td>2.75000E+01</td>
<td>0.00000E+00</td>
<td>-1.96755E-03</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>8</td>
<td>3.75000E+00</td>
<td>5.00000E+00</td>
<td>-3.23155E-03</td>
<td>-5.26724E-03</td>
</tr>
<tr>
<td>9</td>
<td>8.75000E+00</td>
<td>-8.75000E+00</td>
<td>-1.72114E-03</td>
<td>-6.24660E-03</td>
</tr>
<tr>
<td>10</td>
<td>1.37500E+01</td>
<td>-1.00000E+01</td>
<td>-9.83776E-04</td>
<td>-8.33246E-03</td>
</tr>
<tr>
<td>11</td>
<td>1.87500E+01</td>
<td>-8.75000E+00</td>
<td>-2.46412E-04</td>
<td>-6.24660E-03</td>
</tr>
<tr>
<td>12</td>
<td>2.37500E+01</td>
<td>-5.00000E+00</td>
<td>1.26400E-03</td>
<td>-5.26724E-03</td>
</tr>
</tbody>
</table>

13.1.3.4 Struktur und Verschiebungen
13.1.4 Beispiel 4 - Raumfachwerk

13.1.4.1 System und Belastung

Querschnittswerte: St 37, Rohr 101,6×2,9

\[ E = 21000 \text{ kN/cm}^2 \]

\[ A = 8,99 \text{ cm}^2 \]

Belastung: \( P \) wie angegeben in Skizze

gesucht: Schnittgrößen,
Knotenverschiebungen

13.1.4.2 Eingabetensatz (file: it04)
13.1.4.3 Ergebnisse

Die Stabkräfte, Dehnungen und Längsspannungen ergeben sich unter der angegebenen Belastung zu:

<table>
<thead>
<tr>
<th>elem mate</th>
<th>1-coord</th>
<th>2-coord</th>
<th>3-coord</th>
<th>force</th>
<th>strain</th>
<th>stress</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2.5000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.59524E+01</td>
<td>0.31529E-04</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>5.0000</td>
<td>2.5000</td>
<td>0.0000</td>
<td>0.59524E+01</td>
<td>0.31529E-04</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2.5000</td>
<td>5.0000</td>
<td>0.0000</td>
<td>0.59524E+01</td>
<td>0.31529E-04</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.0000</td>
<td>2.5000</td>
<td>0.0000</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2.0000</td>
<td>0.5000</td>
<td>1.5000</td>
<td>-0.60703E+01</td>
<td>-0.32154E-04</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>4.5000</td>
<td>0.5000</td>
<td>1.5000</td>
<td>-0.19742E+02</td>
<td>-0.10457E-03</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>4.5000</td>
<td>3.0000</td>
<td>1.5000</td>
<td>-0.60703E+01</td>
<td>-0.32154E-04</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>3.0000</td>
<td>4.5000</td>
<td>1.5000</td>
<td>-0.60703E+01</td>
<td>-0.32154E-04</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0.5000</td>
<td>4.5000</td>
<td>1.5000</td>
<td>-0.19742E+02</td>
<td>-0.10457E-03</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0.5000</td>
<td>2.0000</td>
<td>1.5000</td>
<td>-0.60703E+01</td>
<td>-0.32154E-04</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>2.5000</td>
<td>2.5000</td>
<td>3.0000</td>
<td>-0.17509E+02</td>
<td>-0.92745E-04</td>
</tr>
</tbody>
</table>

Die Knotenverschiebungen sind:

<table>
<thead>
<tr>
<th>node</th>
<th>1 coord</th>
<th>2 coord</th>
<th>3 coord</th>
<th>1 displ</th>
<th>2 displ</th>
<th>3 displ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>2</td>
<td>5.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.57646E+04</td>
<td>5.54547E-04</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>3</td>
<td>5.00000E+00</td>
<td>5.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>7.12192E-04</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.00000E+00</td>
<td>5.00000E+00</td>
<td>0.00000E+00</td>
<td>-1.57646E+04</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>5</td>
<td>4.00000E+00</td>
<td>1.00000E+00</td>
<td>3.00000E+00</td>
<td>-1.65240E+05</td>
<td>5.86278E-05</td>
<td>-4.52058E-04</td>
</tr>
<tr>
<td>6</td>
<td>1.00000E+00</td>
<td>4.00000E+00</td>
<td>3.00000E+00</td>
<td>4.80531E-05</td>
<td>9.43854E-05</td>
<td>-4.20529E-04</td>
</tr>
</tbody>
</table>

Die Stabkräfte können leicht über Knotengleichgewicht überprüft werden. Die Verschiebungen können mit dem PVK errecknet werden.
13.1.4.4 Struktur und Verschiebungen
13.2 Beispiele für 2D - Stabelement

13.2.1 Beispiel 1 - Zweigelenkrahmen

13.2.1.1 System und Belastung

Querschnittswerte: $E = 21000 \, kN/cm^2$
$\alpha_T = 1.2 \times 10^{-5} \, K^{-1}$
$I_{\text{Stiele}} = 606 \, cm^4$
$A_{\text{Stiele}} = 25.3 \, cm^2$
$h_{\text{Stiele}} = 11.4 \, cm$
$I_{\text{Riegel}} = 318 \, cm^4$
$A_{\text{Riegel}} = 13.2 \, cm^2$
$h_{\text{Riegel}} = 12.0 \, cm$

Belastung: Lastfall 1: Gleichstreckenlast:
$q = 0.05 \, kN/cm$

Lastfall 2: Temperaturlastfall:
Erwärmung Innenseite: $\Delta T = +10 K$
Abkühlung Außenseite: $\Delta T = -10 K$

Lastfall 3: Lagerverschiebung am linken Auflager um
$\Delta_h = 10 \, cm$ horizontal nach rechts

gesucht: Schnittgrößen, Knotenverschiebungen

13.2.1.2 Eingabedatensatz für Lastfall 1 (file: ib2d1a)

13.2.1.3 Eingabedatensatz für Lastfall 2 (file: ib2d1b)

Eingabedatensatz analog zu Lastfall 1 (file: ib2d1a) mit Ausnahme folgender Änderungen:

mate
1,3
21000,25.3,606.,11.4,0.,0.,0.,0.,0,0
1,0,0,0.000012,-10.,10.
2,3
21000,13.2,318.,12.0.,0.,0.,0.,0.,0,0
1,0,0,0.000012,-10.,10.

13.2.1.4 Eingabedatensatz für Lastfall 3 (file: ib2d1c)

Eingabedatensatz analog zu Lastfall 1 (file: ib2d1a) mit Ausnahme folgender Änderungen:
load
  1,,10.,0.,0

mate
  1,3
  21000,25.3,606.,11.4,0.,0.,0.,0.,0,0
  1,0,0,0,0,0
  2,3
  21000,13.2,318.,12.,0.,0.,0.,0.,0,0
  1,0,0,0,0,0

13.2.1.5 Ergebnisse

Das Eckmoment und die horizontale Knotenverschiebung in der Rahmenecke betragen:

\begin{align*}
\text{Lastfall 1: } M &= -860.8 \text{ kNm} \quad \Delta h = 2.58767E-03 \text{ cm} \\
\text{Lastfall 2: } M &= -180.1 \text{ kNm} \quad \Delta h = 5.41333E-04 \text{ cm} \\
\text{Lastfall 3: } M &= -367.9 \text{ kNm} \quad \Delta h = 5.00111E+00 \text{ cm}
\end{align*}

13.2.1.6 Vergleichslösung

Eine Handrechnung ergibt unter der Annahme von $A \rightarrow \infty$ für das Eckmoment im Rahmen

\begin{align*}
\text{Lastfall 1: } M &= -860.95 \text{ kNm} \\
\text{Lastfall 2: } M &= -180.11 \text{ kNm} \\
\text{Lastfall 3: } M &= -367.96 \text{ kNm}
\end{align*}
13.2.2 Beispiel 2 - Rahmentragwerk

13.2.2.1 System und Belastung

Querschnittswerte:  
\[ E = 21000 \text{ kN/cm}^2 \]
\[ I_{\text{Stiele}} = 5700 \text{ cm}^4 \]
\[ A_{\text{Stiele}} = 78.1 \text{ cm}^2 \]
\[ h_{\text{Stiele}} = 20.0 \text{ cm} \]
\[ I_{\text{Riegel}} = 3690 \text{ cm}^4 \]
\[ A_{\text{Riegel}} = 53.8 \text{ cm}^2 \]
\[ h_{\text{Riegel}} = 19.0 \text{ cm} \]

Belastung:  
\[ P = 20.0 \text{ kN} \]
\[ w = 9.5 \text{ kN/m} \]
\[ q = 7.0 \text{ kN/m} \]

gesucht: Schnittgrößen, Knotenverschiebungen

13.2.2.2 Eingabedatensatz (file: ib2d2)

13.2.2.3 Ergebnisse

Das Eckmoment und die horizontale Knotenverschiebung in der Rahmenecke betragen:  
\[ M = -15.40 \text{ kNm} \] und \[ \Delta h = 8.01 \text{ cm} \]

13.2.2.4 Vergleichslösung

Eine Handrechnung ergibt unter der Annahme von \( A \sim \infty \) für das Eckmoment im Rahmen \( M = -15.42 \text{ kNm} \).

13.2.2.5 Schnittkraftlinie

Angegeben ist der Verlauf der Biegemomente \( M_y \).
MOMENT M_y
Layer No. 1

F 3 Max 0.6315E+04
F 3 Min -.1150E+05
13.2.3 Beispiel 3 - elastisch gebetteter Balken

13.2.3.1 System und Belastung

Querschnittswerte: \[ E = 2100 \, \text{kN/cm}^2 \]
\[ I = 190500 \, \text{cm}^4 \]
\[ c = 0.01 \, \text{kN/cm}^3 \]
\[ \ell = 500 \, \text{cm} \]
\[ b = 100 \, \text{cm} \, \text{(Balkenbreite)} \]
\[ h = 28.38 \, \text{cm} \, \text{(Balkenhöhe)} \]

Belastung: \[ F = 1000 \, \text{kN} \]

gesucht: Biegemomente, Durchbiegung

13.2.3.2 Eingabedatensatz (file: ib2d3)

13.2.3.3 Ergebnisse

Das Biegemoment unter der Last beträgt 491.9 kNm, die Durchbiegung 2.86 cm.

13.2.3.4 Vergleichslösung

Die exakte Lösung liefert für das Moment 492 kNm, für die Durchbiegung 2.86 cm.
13.2.4 Beispiel 4 - Halbrahmen

13.2.4.1 System und Belastung

Querschnittswerte:  

\[ E = 2.1 \cdot 10^8 \text{ kN/m}^2 \]
\[ A \approx \infty \]
\[ I = 2000 \cdot 10^{-8} \text{ m}^4 \]
\[ \ell = 10.0 \text{ m} \]

Belastung:  

\[ F = 10 \text{ kN} \]
\[ q = 10 \text{ kN/m} \]

gesucht: Biegemomente nach Theorie II. Ordnung

13.2.4.2 Eingabetensatz (file: ib2d4)

Makroliste:

\[
\begin{align*}
&\text{DT,,1 <CR>} \\
&\text{PROP <CR>} \\
&\text{TIME} \\
&\text{TANG,,1} \rightarrow \text{lineare Lösung} \\
&\text{DT,,0} \\
&\text{TIME} \\
&\text{TANG,,1} \quad \{ \text{jeweils eine Iteration} \}
\end{align*}
\]

13.2.4.3 Ergebnisse

Das Eckmoment und die horizontale Knotenverschiebung in der Rahmenecke betragen nach Theorie II. Ordnung nach dem ersten Iterationsschritt:

\[ M = -44.0 \text{ kNm} \quad \Delta h = 1.046 \text{ cm} \]

13.2.4.4 Vergleichslösung

Eine Handrechnung ergibt unter der Annahme von \( A \sim \infty \) für das Eckmoment im Rahmen und die horizontale Knotenverschiebung nach dem ersten Iterationsschritt

\[ M = -44.06 \text{ kNm} \quad \Delta h = 1.046 \text{ cm} \]
13.3  Beispiele für 3D - Stabelement

13.3.1  Beispiel 1 - räumliche Rahmenecke

13.3.1.1  System und Belastung

Querschnittswerte:

- \( E = 21000 \) kN/cm²
- \( I_y = 4000 \) cm⁴
- \( I_z = 4000 \) cm⁴
- \( I_T = 8000 \) cm⁴
- \( G = 8100 \) kN/cm²
- \( A \approx \infty \)

Belastung: \( q = 1.00 \) kN/m

gesucht: Schnittgrößen

13.3.1.2  Eingabedatensatz (file: ib3d1)

```plaintext
feap
4, 3, 2, 3, 6, 2
mate
1,4
21000,8100,1000,8000,4000,4000,0.,0.,0.,0
coor
0,0.
1,, 0., 0., 0.
2,, 0.,600.,600.
3,, 0., 0.,600.
4,,600., 0.,600.
end
elem
1,1,1,3
2,1,3,2
3,2,3,4
boun
1,,1,1,0,0,0
2,,1,1,1,1,1
4,,1,1,1,1,1
```

13.3.1.3  Ergebnisse

Das Eckmoment im Riegel (in der Lastebene) beträgt -14,6 kNm.

13.3.1.4  Vergleichslösung

Die analytische Vergleichsrechnung ergibt als Eckmoment im Riegel (in der Lastebene) -14,6 kNm.
13.3.2 Beispiel 2 - räumlicher Rahmen

13.3.2.1 System und Belastung

Querschnittswerte:

\[ E = 21000 \text{ kN/cm}^2 \]
\[ I_y = 4000 \text{ cm}^4 \]
\[ I_z = 4000 \text{ cm}^4 \]
\[ I_T = 8000 \text{ cm}^4 \]
\[ G = 8100 \text{ kN/cm}^2 \]
\[ A \approx \infty \]

Belastung: \[ w = 10.0 \text{ kN/m} \]

gesucht: Schnittgrößen

13.3.2.2 Eingabedatensatz (file: ib3d2)

```plaintext
feap
12, 14, 3, 3, 6, 2
coor
1, , 0., 0., 0.
2, , 0., 0.,500.
3, ,500., 0.,500.
4, ,500., 0., 0.
5, , 0.,500., 0.
6,1, 0.,500.,500.
7, ,500.,500.,500.
8, ,500.,500., 0.
9,3,10,11,
10,1,11,12,
11,2, 2, 6,
12,2, 3, 6,
13,2, 3,11,
14,2, 4,11,

coor

1, , 0., 0., 0.
2, , 0., 0.,500.
3, ,500., 0.,500.
4, ,500., 0., 0.
5, , 0.,500., 0.
6,1, 0.,500.,500.
7, ,500.,500.,500.
8, ,500.,500., 0.
9,3,10,11,
10,1,11,12,
11,2, 2, 6,
12,2, 3, 6,
13,2, 3,11,
14,2, 4,11,

mate
1,4
21000,8100,10000,8000,4000,4000,0.,0.,0.,0.,0.
0,0.
21000,10000,0.
3,4
21000,8100,10000,8000,4000,4000,0.,0.,0.,0.,0.
0,0.

elem
1,1, 1, 2,1
0,0.
4,1, 5, 6,
5,3, 6, 7,1
end
9,3,10,11,
10,1,11,12,
11,2, 2, 6,
12,2, 3, 6,
inte
13,2, 3,11,
14,2, 4,11,

13.3.2.3 Ergebnisse

Die Eckmomente in den beiden Rahmen errechnen sich zu \( \pm 62.5 \text{ kNm} \).

13.3.2.4 Vergleichslösung

Die analytische Vergleichsrechnung unter der Annahme \( A \sim \infty \) ergibt für die Eckmomente in den beiden Rahmen jeweils einen Wert von \( \pm 62.5 \text{ kNm} \).
13.4 Beispiele für Scheibenelemente

13.4.1 Beispiel 1 - Kragarm mit Einzellast

13.4.1.1 System und Belastung

Querschnittswerte: \( E = 4 \cdot 10^8 \) kN/cm²

\( l = 100 \) cm

\( h = 1 \) cm

\( d = 0.1 \) cm

Belastung: \( P = 10 \) kN

gesucht: Endabsenkung

13.4.1.2 Eingabetextensatz für Beispiel 1 (file: is1)

```plaintext
feap kragarm mit einzellast
22, 10, 1, 2, 2, 4

cons
n=10
m=1
l=100
h=1
d=0.1
p=10

bloc
4,n,m,1,1,1,0
1,0,0
2,1,0
3,1,h
4,0,h

ebou
1,0,1,1

load
(n+1)*(m+1),0,0,-p
```

```
mate
1,5 oder 1,6
4.0e+8,0,0,1
4.0e+8,0,0,d,1

d,0,0,0,0

end
inte
stop
```
13.4.1.3 Ergebnisse

Die Verschiebungen der äußeren Eckknoten für die beiden Elemente betragen:

**Plane Stress Linear Elastic Element (elmt05)**

```
node 1 coord 2 coord 1 displ 2 displ
22 1.00000E+02 1.00000E+00 1.47059E-04-1.96088E-02
```

**Plane Stress Element (elmt06)**

```
node 1 coord 2 coord 1 displ 2 displ
22 1.00000E+02 1.00000E+00 7.50000E-03-9.97550E-01
```

Das Beispiel zeigt, daß Biegeprobleme nur eigenschränkt mit Scheibenelementen zu berechnen sind. Elmt05 ist nicht geeignet, Elmt06 liefert gute Ergebnisse. Bei den gegebenen Abmessungen sollte jedoch ein Stabelement verwendet werden!

Eine Verbesserung der Ergebnisse läßt sich für Elmt05 bei Netzverfeinerung erreichen (Netz 100 x 1):

**Plane Stress Linear Elastic Element (elmt05)**

```
node 1 coord 2 coord 1 displ 2 displ
202 1.00000E+02 1.00000E+00 5.00000E-03-6.66700E-01
```

13.4.1.4 Vergleichslösung

Die Handrechnung für einen Bernoulli-Stab ergibt:

\[
f = \frac{P l^3}{3EI} = 1 \text{ cm}
\]
13.4.2 Beispiel 2 - Wandscheibe mit Gleichstreckenlast auf zwei Stützen

13.4.2.1 System und Belastung

Systemkennwerte: \( E = 4 \cdot 10^8 \text{ kN/m}^2 \)
\( l = 10 \text{ m} \)
\( h = 10 \text{ m} \)
\( d = 0.1 \text{ m} \)
\( b = 1 \text{ m} \)

Belastung: \( q = 22 \text{ kN/m} \)

gesucht: Spannungsverlauf \( \sigma_x \) entlang der Symmetrieachse
13.4.2.2 Eingabedatensatz (file: is2)

feap scheibe mit streckenlast 4 knoten/element
121, 100, 1, 2, 2, 4

c n = Anzahl El in x
c m = Anzahl El in y
c l = Laenge
c h = Hoehe
c d = Dicke
c q = Streckenlast
c e = Anfangsknoten: Last
c f = Endknoten: Last

cons
n=10
m=10
l=10
h=10
d=0.1
q=22
f=(n+1)*(m+1)
e=f-n

bloc
4,n,m,1,1,1,0
1, 0,0,0
2,1/2,0,0
3,1/2,h,0
4, 0,h,0

edge *auflager
0,0,1,0,0,0,0,0,0,0
0,1

ebou *symmetrierand
1,1/2,1,0

eloa *streckenlast auf oberen rand
0,h,1/2,h,0,0,0
1,1,0,-q

mate
1,5 oder 1,6
4.0e+8,0,0,0,0,0,1
4.0e+8,0,0,0,0,d,1
d,0,0,0,0,0,0,0,0,0

end
inte
stop
13.4.2.3 Ergebnisse

Unter Ausnutzung der Symmetrie sehen die Ergebnisse wie folgt aus:

![Diagramm der Ergebnisse](image)

*Abb. 13.4.2.3: Ergebnisse unter Symmetrie.*

Institut für Baustatik, KIT
Die Spannung der Knoten auf der Symmetrielinie (Kn.11- unten/Kn.121-oben):

**nodal stresses (elmt05)**

<table>
<thead>
<tr>
<th>node</th>
<th>N-FOR N_11</th>
<th>N-FOR N_12</th>
<th>N-FOR N_22</th>
<th>N-FOR N_33</th>
<th>N-FOR N_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>2.48136E+02</td>
<td>3.79780E-01</td>
<td>6.36894E-01</td>
<td>0.00000E+00</td>
<td>2.48137E+02</td>
</tr>
<tr>
<td></td>
<td>6.36312E-01</td>
<td>8.79185E-02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>-3.58294E+01</td>
<td>-1.06836E+00</td>
<td>-2.17919E+02</td>
<td>0.00000E+00</td>
<td>-3.58231E+01</td>
</tr>
<tr>
<td></td>
<td>-2.17925E+02</td>
<td>-3.36153E-01</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**nodal stresses (elmt06)**

<table>
<thead>
<tr>
<th>node</th>
<th>N-FOR N_11</th>
<th>N-FOR N_12</th>
<th>N-FOR N_22</th>
<th>N-FOR N_33</th>
<th>N-FOR N_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>3.40312E+02</td>
<td>5.30462E-01</td>
<td>1.10814E+00</td>
<td>0.00000E+00</td>
<td>3.40313E+02</td>
</tr>
<tr>
<td></td>
<td>1.10731E+00</td>
<td>8.96014E-02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>-3.88116E+01</td>
<td>-1.06232E+00</td>
<td>-2.17835E+02</td>
<td>0.00000E+00</td>
<td>-3.88053E+01</td>
</tr>
<tr>
<td></td>
<td>-2.17842E+02</td>
<td>-3.39974E-01</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

13.4.2.4 Vergleichslösung

Das Näherungsverfahren von Schleich führt zu folgendem Ergebnis:

\[
\sigma_x^{Scheibe} = \sigma_x^{Balken} + \Delta \sigma_x
\]

\[
\sigma_y^{Scheibe} = 0 + \Delta \sigma_y
\]

Damit ergeben sich entlang der Symmetrielinie die Werte:

\[
\begin{align*}
\frac{h}{l} & \quad \sigma_x^{oben} \quad \sigma_x^{mitte} \quad \sigma_x^{unten} \\
\frac{10}{10} & \quad -55,0 \text{ kN/m}^2 \quad -74,8 \text{ kN/m}^2 \quad +365,4 \text{ kN/m}^2
\end{align*}
\]
13.4.3 Beispiel 3 - Wandscheibe mit Loch

13.4.3.1 System und Belastung

Querschnittswerte:  
\[ E = 21 \cdot 10^6 \text{ kN/m}^2 \]
\[ l_1 = 12 \text{ m} \]
\[ l_2 = 4 \text{ m} \]
\[ d = 20 \text{ cm} \]

Belastung:  
\[ q = 20 \text{ kN/m}^2 \]

gesucht: Verschiebung Wandmitte
13.4.3.2 NEGE-Eingabedatensatz (file: is3.neg)

nege
4,4,2,0,1.5

gem
1,0,0,0.0
2,1,0,0.0
3,11.5,0.0
4,12.0,0.0
5,12.0,4.0
6,0,0,4.0
7,6.0,0.2
8,8.0,0.2
9,8.0,1.8
10,7.5,2.0
11,6.5,2.0
12,6.0,1.8

segm
1,1,2
2,2,3
3,3,4
4,4,5
5,5,6
6,6,1
7,7,12
8,12,11
9,11,10
10,10,9
11,9,8
12,8,7

regi
1,12,1,2,3,4,5,6,7,8,9,10,11,12

pres
5,0,-20,0,-20

fixe
1,01
3,01
4,10
6,10

mate
1,6
21e+6,0,0,0.2,2

end

ini

stop
13.4.3.3 Ergebnisse – verformtes Netz

Verformung des mittleren Knotens:

<table>
<thead>
<tr>
<th>node</th>
<th>1 coord</th>
<th>2 coord</th>
<th>1 displ</th>
<th>2 displ</th>
</tr>
</thead>
<tbody>
<tr>
<td>728</td>
<td>6.06239E+00</td>
<td>0.00000E+00</td>
<td>-3.46907E-06</td>
<td>-2.78284E-05</td>
</tr>
</tbody>
</table>

13.4.3.4 Vergleichslösung

Eine Abschätzung der Durchbiegung durch Handrechnung mit Analogie zur Balkenlösung.

\[ \max f = \frac{ql^4}{384 \cdot EI} = 4.8214 \cdot 10^{-5} m \]
13.5 Beispiele für Plattenelement

13.5.1 Beispiel 1 - Platte mit verschiedenen Lagerbedingungen

13.5.1.1 Systeme und Belastung

Kennwerte: \( E = 3.0 \cdot 10^7 \ \text{kN/m}^2 \)

\[
\begin{align*}
\ell^\text{ges}_x &= 10.0 \ \text{m} \\
\ell^1_x &= 6.0 \ \text{m} \\
\ell^2_x &= 4.0 \ \text{m} \\
l_y &= 4.0 \ \text{m} \\
d &= 0.1 \ \text{m}
\end{align*}
\]

Belastung: \( q = 5.0 \ \text{kN/m}^2 \)

gesucht: Biegemomente \( m_x \) und \( m_y \)

- Einfeldträger – Lagerbedingung 1
- Allseitig gelenkig gelagerte Platte – Lagerbedingungen 1 und 3
- Zweifeldplatte – Lagerbedingungen 2 und 3
13.5.1.2 Eingabedatensatz

feap Platte mit Flächenlast
77,60,1,3,3,4

c n = Anzahl El in x
c m = Anzahl El in y
c l = Länge
c b = Breite
c d = Dicke
c q = Flächenlast

cons
n=10
m=6
l=10
b=4
d=0.10
q=5

bloc
4,n,m,1,1,1,0
1,0,0,0
2,1,0,0
3,1,b,0
4,0,b,0

ebou *Einfeldträger
1,0,1,0,0
1,1,1,0,0

mate *DKQ-Platte Element
1,7
3.0e+7,0,d,-q,0,0,0,0,0,0,0,0
0,0

end
inte
stop

Durch den Austausch von *ebou werden die übrigen Lagerbedingungen modelliert:

**ebou *allseitig gelagerte Platte**
1,0,1,0,0
1,1,1,0,0
2,0,1,0,0
2,b,1,0,0

**ebou *unterstützte Platte**
1,0,1,0,0
1,1,1,0,0
1,1*0.6,1,0,0
2,0,1,0,0
2,b,1,0,0
### 13.5.1.3 Ergebnisse

Einfeldträger / allseitig gelagerte Platte:

#### 11-Momente

<table>
<thead>
<tr>
<th>Stress</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.50E+00</td>
</tr>
<tr>
<td>2</td>
<td>1.53E+00</td>
</tr>
<tr>
<td>3</td>
<td>2.30E+00</td>
</tr>
<tr>
<td>4</td>
<td>3.07E+00</td>
</tr>
<tr>
<td>5</td>
<td>3.84E+00</td>
</tr>
<tr>
<td>6</td>
<td>4.61E+00</td>
</tr>
<tr>
<td>7</td>
<td>5.39E+00</td>
</tr>
<tr>
<td>8</td>
<td>6.17E+00</td>
</tr>
</tbody>
</table>

#### 22-Momente

<table>
<thead>
<tr>
<th>Stress</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.62E-01</td>
</tr>
<tr>
<td>2</td>
<td>1.43E+00</td>
</tr>
<tr>
<td>3</td>
<td>2.64E+00</td>
</tr>
<tr>
<td>4</td>
<td>3.78E+00</td>
</tr>
<tr>
<td>5</td>
<td>4.96E+00</td>
</tr>
<tr>
<td>6</td>
<td>6.19E+00</td>
</tr>
<tr>
<td>7</td>
<td>7.31E+00</td>
</tr>
<tr>
<td>8</td>
<td>8.40E+00</td>
</tr>
</tbody>
</table>

**FE-Lsg.**: \( m_{11} = 61,6667 \) kNm/m

**FE-Lsg.**: \( m_{22} = 8,49015 \) kNm/m

Unterstützte Platte:

#### 11-Momente

<table>
<thead>
<tr>
<th>Stress</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-4.44E+00</td>
</tr>
<tr>
<td>2</td>
<td>-3.47E+00</td>
</tr>
<tr>
<td>3</td>
<td>-2.50E+00</td>
</tr>
<tr>
<td>4</td>
<td>-1.52E+00</td>
</tr>
<tr>
<td>5</td>
<td>-5.57E+00</td>
</tr>
<tr>
<td>6</td>
<td>-1.44E+00</td>
</tr>
<tr>
<td>7</td>
<td>1.38E+00</td>
</tr>
<tr>
<td>8</td>
<td>2.35E+00</td>
</tr>
</tbody>
</table>

#### 22-Momente

<table>
<thead>
<tr>
<th>Stress</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.84E-01</td>
</tr>
<tr>
<td>2</td>
<td>1.44E+00</td>
</tr>
<tr>
<td>3</td>
<td>2.10E+00</td>
</tr>
<tr>
<td>4</td>
<td>2.76E+00</td>
</tr>
<tr>
<td>5</td>
<td>3.43E+00</td>
</tr>
<tr>
<td>6</td>
<td>4.07E+00</td>
</tr>
<tr>
<td>7</td>
<td>4.73E+00</td>
</tr>
<tr>
<td>8</td>
<td>5.39E+00</td>
</tr>
</tbody>
</table>

**FE-Lsg.**: \( m_{11}^1 = 2,28624 \) kNm/m

**FE-Lsg.**: \( m_{11}^2 = 2,39101 \) kNm/m

**FE-Lsg.**: \( m_{11}^3 = -4,44301 \) kNm/m

**FE-Lsg.**: \( m_{22}^1 = 4,75442 \) kNm/m

**FE-Lsg.**: \( m_{22}^2 = 2,31684 \) kNm/m
13.5.1.4 Vergleichslösungen

Handrechnung bzw. Anwendung der Tafeln von Pieper/Martens führen zu folgenden Ergebnissen:

- **Einfeldträger:**
  Feldmoment $m_x$:
  \[
  m_x = \frac{ql^2}{8}
  \]
  \[
  m_x = \frac{5 \cdot 10^2}{8} = 62,5 \text{ kNm/m}
  \]

- **Allseitig gelenkig gelagerte Platte:**
  Feldmomente $m_x$ und $m_y$:
  \[
  l_x/l_y > 2 \Rightarrow m_x \rightarrow 0
  \]
  \[
  m_y = \frac{5 \cdot 4^2}{8} = 10,0 \text{ kNm/m}
  \]

- **Zweifeldplatte**
  Momente $m_x$ und $m_y$ in Feld 1:
  \[
  l_x/l_y = 1,5 \Rightarrow m_x = \frac{5 \cdot 4^2}{33,5} = 2,39 \text{ kNm/m}
  \]
  \[
  m_y = \frac{5 \cdot 4^2}{15,0} = 5,33 \text{ kNm/m}
  \]

  Momente $m_x$ und $m_y$ in Feld 2:
  \[
  l_x/l_y = 1,0 \Rightarrow m_x = \frac{5 \cdot 4^2}{29,1} = 2,75 \text{ kNm/m}
  \]
  \[
  m_y = \frac{5 \cdot 4^2}{32,8} = 2,44 \text{ kNm/m}
  \]

Stützmoment $m_x$:
\[
  m_{s0} = \frac{m_{s01} + m_{s02}}{2}
  \]
\[
  = 5 \cdot 4^2 \cdot \frac{1}{2} \left( \frac{1}{-8,9} + \frac{1}{-11,9} \right) = -7,85 \text{ kNm/m}
  \]
13.5.1.5 Bemessung der unterstützten Platte

Feldbewehrung:

\[ FE - \text{Lsg.: } a_{11}^F = 1,209 \text{ cm}^2/m \]

Stützbewehrung:

\[ FE - \text{Lsg.: } a_{11}^S = 2,353 \text{ cm}^2/m \]
13.6 Beispiele für Rotationsschalenelement

13.6.1 Beispiel 1 - Zylinder mit Randlast

13.6.1.1 System und Belastung

Systemdaten: \[ E = 21000 \text{ kN/cm}^2 \]
\[ \mu = 0.2 \]
\[ \text{Radius } R = 100 \text{ cm} \]
\[ \text{Dicke } t = 0.5 \text{ cm} \]
\[ \text{Höhe } H = 200 \text{ cm} \]

Belastung: \[ Q = 1.0 \text{ kN/cm} \]

für die Eingabe als Knotenlast muß über den Umfang aufsummiert werden.

⇒ \[ F = 2\pi R Q = 628.3185 \text{ kN} \]

gesucht: Schnittgrößen, Knotenverschiebungen

13.6.1.2 Eingabedatensatz (file: isr-1)

```
feap ** Zylinder mit Querlast oben (32 Elemente) **
33,32,1,2,3,2

coor
1, 1, 100, 0,
9, 1, 100, 150
33, 0, 100, 200

elem
1,1,1,2,1,

boun
1,,0,1,0

load
33,,628.3185,0,0

mate
1,8
21000,0.2,0.5
0,0,0,0,0

end
inte
stop
```
13.6.1.3 Ergebnisse

Analytische Lösung:

Radiale Verschiebung am oberen Rand: \( w = 0.3590917 \, \text{cm} \)
Verdrehung am oberen Rand: \( \beta = 0.06464978 \)

FEM Lösung:

\text{n o d a l \ d i s p l a c e m e n t s}

\[
\begin{array}{cccccc}
\text{node} & 1 \text{ coord} & 2 \text{ coord} & 1 \text{ displ} & 2 \text{ displ} & 3 \text{ displ} \\
33 & 1.00000E+02 & 2.00000E+02 & 3.51289E-01 & -1.90476E-03 & 6.46498E-02 \\
\end{array}
\]
13.6.2 Beispiel 2 - Zylinder mit Wasserfüllung

13.6.2.1 System und Belastung

Systemdaten:

- \( E = 21000 \ \text{kN/cm}^2 \)
- \( \mu = 0.2 \)
- \( \text{Radius } R = 100 \ \text{cm} \)
- \( \text{Dicke } t = 0.5 \ \text{cm} \)
- \( \text{Höhe } H = 200 \ \text{cm} \)

Belastung:

Wasser mit \( \gamma = 0.00001 \ \text{kN/cm}^3 \)

Die Eingabe der Wasserlast erfolgt im Materialdatensatz

gesucht: Schnittgrößen, Knotenverschiebungen

13.6.2.2 Eingabedatensatz (file: isr-2)

```feap
17,16,1,2,3,2

coor
1, 1, 100, 0
17,0, 100, 200

elem
1,1,1,2,1

boun
1,,0,1,0

mate
1,8
21000,0.2,0.5
0,0,-1E-5,200,2,0
0,0,0

end

inte

stop
```
13.6.2.3 Ergebnisse

Analytische Lösung am unteren Rand (\( p = \gamma H = 0.002 \text{kN/cm}^2 \))

\[ w = \frac{R^2p}{Et} = 0.0019 \text{cm} \]

\[ \beta = \frac{R^2p}{EtH} = \frac{w}{H} = 9.5 \times 10^{-6} \]

\[ n_t = Rp = 0.2 \text{kN/cm} \]

FEM Lösung:

\text{n o d a l \ d i s p l a c e m e n t s}

\begin{tabular}{cccccc}
node & 1 & coord & 2 & coord & 1 & displ & 2 & displ & 3 & displ \\
1 & 1.00000E+02 & 0.00000E+00 & 1.90356E-03 & 0.00000E+00 & 9.34138E-06 \\
\end{tabular}

\text{stress resultants for axisymmetric shell}

\begin{tabular}{cccccccc}
elmt & ns-force & nt-force & ms-moment & mt-moment & qs-force \\
1 & 1.634E-16 & 1.937E-01 & -2.738E-06 & -5.477E-07 & 4.381E-07 \\
\end{tabular}
13.6.3 Beispiel 3 - Zylinder mit gleichmäßiger Temperaturbelastung

13.6.3.1 System und Belastung

<table>
<thead>
<tr>
<th>t_a</th>
<th>t_i</th>
<th>t_i</th>
<th>t_a</th>
</tr>
</thead>
</table>

Radius R = 1200 cm
Dicke t = 16 cm
Höhe H = 500 cm

Belastung: \( t_i = t_a = +20 \) K

gesucht: Schnittgrößen

13.6.3.2 Eingabetatensatz (file: isr-6)

feap
51,50,1,2,3,2

coor
1, 1, 1200, 0
21,1, 1200, 100
30,1, 1200, 400
51,0, 1200, 500

elem
1,1,1,2,1

boun
1,,1,1,1

mate
1,8
2100,1/6,16
0,0,0,0,0,0
12e-6,20,20

der

end

13.6.3.3 Ergebnisse

Lösungen am unteren Rand:

\[
\begin{align*}
N_{t, \text{anal.}} &= -8.06 \text{kN} & N_{t, \text{FEM}} &= -8.052 \text{kN} \\
M_{s, \text{anal.}} &= 38.00 \text{kN} & M_{s, \text{FEM}} &= 35.680 \text{kN}
\end{align*}
\]

Lösungen am oberen Rand:

\[
\begin{align*}
N_{t, \text{anal.}} &= 0.07 \text{kN} & N_{t, \text{FEM}} &= 0.366 \text{kN} \\
M_{s, \text{anal.}} \approx 0.00 \text{kN} & M_{s, \text{FEM}} &= 0.075 \text{kN}
\end{align*}
\]

Die analytischen Lösungen sind entnommen aus E. HAMPE; Statik rotationssymmetrischer Flächentragwerke Band 2; VEB Verlag für Bauwesen, Berlin 1964.
13.6.4 Beispiel 4 - Zylinder mit ungleichmäßiger Temperaturbelastung

13.6.4.1 System und Belastung

Systemdaten: \( E = 2100 \, \text{kN/cm}^2 \)
\( \mu = 1/6 \)
Radius \( R = 1200 \, \text{cm} \)
Dicke \( t = 16 \, \text{cm} \)
Höhe \( H = 500 \, \text{cm} \)

Belastung: \( t_i = +30 \, \text{K} \)
\( t_a = 0 \, \text{K} \)

gesucht: Schnittrößen

13.6.4.2 Eingabedatensatz (file: isr-7)

```
feap
51,50,1,2,3,2
coor
1, 1, 1200, 0
21,1, 1200, 100
30,1, 1200, 400
51,0, 1200, 500
elem
1,1,1,2,1
boun
1,,0,1,1
mate
1,8
2100,1/6,16
0,0,0,0,0
12e-6,0,30
end
inte
stop
```

13.6.4.3 Ergebnisse

Lösungen am unteren Rand:

\[ N_{t \text{anal.}} \approx -0.00 \, \text{kN} \quad N_{t \text{FEM}} = -0.844 \, \text{kN} \]
\[ M_{s \text{anal.}} = 19.34 \, \text{kN} \quad M_{s \text{FEM}} = 20.340 \, \text{kN} \]

Lösungen am oberen Rand:

\[ N_{t \text{anal.}} = 4.13 \, \text{kN} \quad N_{t \text{FEM}} = 4.307 \, \text{kN} \]
\[ M_{s \text{anal.}} \approx 0.00 \, \text{kN} \quad M_{s \text{FEM}} = 0.018 \, \text{kN} \]

Die analytischen Lösungen sind entnommen aus E. Hampe; Statik rotationssymmetrischer Flächentragwerke Band 2; VEB Verlag für Bauwesen, Berlin 1964.
13.6.5 Beispiel 5 - Kegel mit Randmoment

13.6.5.1 System und Belastung

Radius $R_u = 140$ cm
Radius $R_o = 100$ cm
Dicke $t = 0.5$ cm
Höhe $H = 69.282$ cm
Neigungswinkel $\theta = 60$ Grad

Belastung: $M = 1.0 \text{ kNcm/cm}$

für die Eingabe als Knotenlast muß über den Umfang aufsummiert werden.

$\Rightarrow F = 2\pi R M = 628.3185 kN/cm$

gesucht: Schnittgrößen, Knotenverschiebungen

13.6.5.2 Eingabedatensatz (file: isr-3)

feap ** Kegel mit Randmoment **
65,64,1,2,3,2

coor
1, 1, 140., 0.0
65, , 100., 69.282

elem
1,1,1,2,1

boun
1,,1,1,1

load
65,0,0,0,628.3185,,

mate
1,8
21000,0.2,0.5
0,0,0,0,0,0
0,0,0

diagram

Institut für Baustatik, KIT Cont. Mesh Macro Plot Mate3D Elem Nege Gmesh Cylt Isogeo 510
13.6.5.3 Ergebnisse

Analytische Lösung:

Radiale Verschiebung am oberen Rand: \( w = 0.0646497 \text{ cm} \)
Verdrehung am oberen Rand: \( \beta = 0.025597 \)

FEM Lösung:

\text{noddal displacements}

\begin{verbatim}
node 1 coord 2 coord 1 displ 2 displ 3 displ
65 1.0000E+02 6.92820E+01 6.45399E-02 3.73242E-02 2.55777E-02
\end{verbatim}
13.6.6 Beispiel 6 - Kugel mit Innendruck

13.6.6.1 System und Belastung

Systemdaten: 
- $E = 21000 \, kN/cm^2$
- $\mu = 0.2$
- Radius $R = 100 \, cm$
- Dicke $t = 0.25 \, cm$

Belastung: Innendruck $p = 0.1 \, kN/cm^2$

Die Eingabe des Druckes erfolgt im Materialdatensatz

gesucht: Schnittgrößen, Knotenverschiebungen

13.6.6.2 Eingabedatensatz (file: isr-4)

```plaintext
feap ** Kugelschale mit Innendruck Membranlagerung (50 Elemente) **
51,50,1,2,3,2

coor
1,1,100, 0
51,,100,90

pola
1,51,1

elem
1,1,1,2,1

boun
1,, 0,1,0
51,,1,0,1

mate
1,8
2.1e+04,0.2,0.25
0,-0.1,0,0,0,0
0,0,0

end
inte
stop
```

Institut für Baustatik, KIT Cont. Mesh Macro Plot Mate3D Elem Nege Gmesh Cybt Isoge 512
13.6.6.3 Ergebnisse

Analytische Lösung:

\[ N_s = N_t = \frac{pR}{2} = 5.0 \text{kN/cm} \]
\[ w = \frac{(1 - \mu)pr^2}{2Et} = 76.190476 \cdot 10^{-3} \text{cm} \]

FEM Lösung:

\text{n o d a l \ d i s p l a c e m e n t s}

\begin{verbatim}
node 1 coord 2 coord 1 displ 2 displ 3 displ
 1 1.00000E+02 0.00000E+00 7.61733E-02 0.00000E+00 -3.60341E-09
\end{verbatim}

\text{stress resultants for axisymmetric shell}

\begin{verbatim}
elem ns-force nt-force ms-moment mt-moment qs-force
 1 4.999E+00 4.999E+00 -4.164E-09 -8.164E-10 2.651E-09
\end{verbatim}
13.6.7 Beispiel 7 - Kreisplatte mit Flächenlast

13.6.7.1 System und Belastung

Systemdaten: \(E = 20000 \, MN/m^2\)  
\(\mu = 0.0\) bzw. 0.2  
\(a = 4 \, m\)  
\(b = 2 \, m\)  
\(h = 0.1 \, m\)

Belastung: \(p_1 = 12 \, kN/m^2\)  
\(p_2 = 8 \, kN/m^2\)

Die Eingabe der Belastung erfolgt im Materialdatensatz

gesucht: Schnittgrößen, Verschiebungen

13.6.7.2 Eingabedatensatz (file: isr-5)

feap ** Kreisplatte (20 Elemente) **  
151,150,2,2,3,2

coor  
1, 1, 0.0, 0.0  
100,1, 4.0, 0.0  
151,0, 6.0, 0.0

elem  
1, 1, 1, 2,1  
100,2,100,101,1

boun  
100,0,0,1,0  
151,0,1,0,1

mate  
1,8  
2.0e+07,0.0,0.1  
0,-12,1,4,1,0  
0,0,0  
2,8  
2.0e+07,0.0,0.1  
0,-12,0,0,0,0  
0,0,0

date  
inte  
stop

Institut für Baustatik, KIT Cont. Mesh Macro Plot Mate3D Elem Nege Gmesh Cylt Isogeo 514
13.6.8 Beispiel 8 - Kreisplatte mit Loch unter Temperaturbelastung

13.6.8.1 System und Belastung

Systemdaten:
\[ E = 2100 \text{ kN/cm}^2 \]
\[ \mu = \frac{1}{6} \]
\[ a = 100 \text{ cm} \]
\[ b = 900 \text{ cm} \]
\[ \text{Dicke } h = 16 \text{ cm} \]

Belastung:
\[ t_o = +20 \text{ K} \]
\[ t_u = -20 \text{ K} \]

gesucht: Schnittgrößen, Verschiebungen

13.6.8.2 Eingabetatsatz (file: isr-8)

```
feap
16,15,1,2,3,2

coor
1, 1, 100, 0
16,0,1000, 0

elem
1,1,1,2,1

boun
1,,0,1,0

mate
1,8
2100,1/6,16
0,0,0,0,0
12e-6,-20,20

end
inte
stop
```

Ausgangslage und Biegelinie
13.7 Beispiele für allgemeines Schalenelement

13.7.1 Beispiel 1 - Eingespannter T-Träger

13.7.1.1 System und Belastung

Systemdaten: \( E = 21000 \text{ kN/cm}^2 \)
\( \mu = 0.0 \)

Belastung: \( F = 10.0 \text{ kN am Trägerende} \)

gesucht: Durchbiegung, Spannungen

Maße in [cm]

13.7.1.2 Systemdaten für eine analytische Vergleichslösung

Fläche \( A = 49 \text{ cm}^2 \)
Schwerpunkt \( z_s \approx 1.0 \text{ cm} \)
Trägheitsmoment \( I_y \approx 296 \text{ cm}^4 \)
Widerstandsmomente \( W_o \approx 148 \text{ cm}^3 \)
\( W_u \approx 33 \text{ cm}^3 \)
**Eingabedatensatz (file: isg-1)**

feap ** T-Traeger mit allgemeinem Schalenelement **
143,110,2,3,6,4

const
l=100
b=20
h=10
x=10
z=5
y=6
i=(x+1)*(z+1)+1
j=x*z+1
f=(x+1)*(z+1)

block 1 (Steg)
4,x,z,1,1,1
1, 0.0, 0.0, h
2, 1, 0.0, h
3, 1, 0.0, 0.0
4, 0.0, 0.0, 0.0

block 2 (Platte)
4,x,y,i,j,2
1, 0.0, b/2, 0.0,
2, 1, b/2, 0.0,
3, 1,-b/2, 0.0,
4, 0.0,-b/2, 0.0,

ebou
1,0.0, 1,1,1, 1,1,1

load
f,,0,0,10

mate
1,9
21000,0.0,0.0,0.0,1.0
0,0,0,0,0,0,0,0,0,
0,0,0
2,9
21000,0.0,0.0,0.0,2.0
0,0,0,0,0,0,0,0,0
0,0,0

dae
end
tie
inte
stop

**Bemerkung zu den Konstanten:**

l laenge in cm
b breite platte in cm
h hoehe steg in cm
x anzahl elemente laengs
z anzahl elemente hoehe
y anzahl elemente quer (platte)
i,j startknoten und element fuer block 2
f knoten mit last

Verformte Lage
13.7.2 Beispiel 2 - Beidseitig eingespannter Zylinder mit Einzellasten

13.7.2.1 System und Belastung

- **Detail Auflager**

- **Systemdaten:**
  - \( E = 3 \cdot 10^6 \text{ kN/cm}^2 \)
  - \( \mu = 0.3 \)
  - Länge \( L = 600 \text{ cm} \)
  - Radius \( R = 300 \text{ cm} \)
  - Dicke \( t = 3 \text{ cm} \)

- **Belastung:**
  - \( F = 1.0 \text{ kN} \)

- gesucht: Verschiebung der Lastknoten

13.7.2.2 Eingabedatensatz (file: isg-2)

Die Berechnung erfolgt unter Ausnutzung der mehrfachen Symmetrie an einem Achtel des Zylinders.

```feap
feap 256,225,1,3,6,4
mate 1,9
  3.e6,0.3,0,t
cons
  i=15
  j=15
  r=300
  t=3
  l=300
bloc
  4,i,j,1,1,1
  1,r,90,0
  2,r,0,0
  3,r,0,-l
  4,r,90,-l
pola
  1,(i+1)*(j+1),1
ebou
  1,0,1,0,0,0,0,1
  2,0,0,1,0,0,0,1
  3,0,0,0,1,1,1,0
  3,-1,1,1,0,1,1,0
load
  1,0,0,-0.25
mate
  1,9
  3.e6,0.3,0,t
  0,0,0,0,0,0,0,0
  0,0,0
end
inte
stop
```

Institut für Baustatik, KIT
Cont. Mesh Macro Plot Mate3D Elem Nege Gmesh Cylt Isogeob 518
### 13.7.2.3 Ergebnisse und verformtes Netz

Ergebnisse der FEM-Rechnung:

**nodal displacements**

<table>
<thead>
<tr>
<th>node</th>
<th>1 displ</th>
<th>2 displ</th>
<th>3 displ</th>
<th>4 displ</th>
<th>5 displ</th>
<th>6 displ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000E+00</td>
<td>-1.82075E-05</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
</tbody>
</table>

Eine analytische Berechnung ergibt eine Verschiebung des Lastknotens von:

\[ u_{ref} = 1.8249 \cdot 10^{-5} \text{cm} \]
13.7.3 Beispiel 3 - Membrangelagertes Tonnendach

13.7.3.1 System und Belastung

Systemdaten:

\[ \begin{align*}
E &= 3 \cdot 10^7 \text{ kN/m}^2 \\
\mu &= 0.0 \\
\text{Länge } L &= 15.0 \text{ m} \\
\text{Radius } R &= 4.0 \text{ m} \\
\text{Dicke } t &= 0.05 \text{ m}
\end{align*} \]

Belastung:

\[ \text{Eigengewicht } g = 3.75 \text{ kN} \]

gesucht: Schnittkraftverläufe

13.7.3.2 Eingabedatensatz (file: isg-3)

```
feap,,1,3,6,4
cons
i=20
j=20
r=4
t=0.05
l=15
bloc
4,1,j,1,1,1
1,r,180, 0
2,r,180, l
3,r, 0, l
4,r, 0, 0
pola
1,(i+1)*(j+1),1

ebou
3, 0, 0,1,0, 0,0,0 
3, 1, 0,1,0, 0,0,0 
2, 0, 0,0,1, 0,0,0

angl
2,1
1, (i+1), 0 
(i+1)*j+1, 0,180 
(i+1), (i+1), 0 
(i+1)*(j+1), 0,180
```
13.7.3.3 Schnittkraftverläufe und Ergebnisvergleich


Ergebnisse der FEM-Rechnung:

\[
\begin{array}{cccccccc}
\text{elmt} & \text{x-coord} & N_{xx} & N_{xy} & N_{yy} & N_1 & N_2 & \text{angle}_N \\
\text{matl} & \text{y-coord} & M_{xx} & M_{xy} & M_{yy} & M_1 & M_2 & \text{angle}_M \\
\text{z-coord} & \text{eps}_{xx} & \text{eps}_{xy} & \text{eps}_{yy} & \kappa_{xx} & \kappa_{xy} & \kappa_{yy} \\
\hline
1 & 2.590 & 1.923E-03 & 4.568E-03 & -7.389E-02 & 2.197E-03 & -7.416E-02 & 3.44 \\
\end{array}
\]

Die analytische Berechnung ergibt für \( \varphi = 40.505^\circ \):

\[
\begin{align*}
N_s &= -g \frac{s}{R} (L - s) \sin \varphi = -30.06 \, kN/m \sim N_{xx} \\
N_\varphi &= -g R \sin \varphi = -9.74 \, kN/m \sim N_{yy} \\
T &= -g (L - 2s) \cos \varphi = -14.61 \, kN/m \sim N_{xy}
\end{align*}
\]
Chapter 14

Adding elements

14.1 General information

Elements can be added to FEAP without knowing the exact structure of the main program. Since the set of parameters is always the same in the element subroutines the following rules are of general use. The reader may also consult chapter 19 of the textbook, *The Finite Element Method* by O. C. Zienkiewicz & R. L. Taylor, McGraw Hill, 2000, to obtain additional information concerning coding elements for FEAP.

When an element subroutine is called from FEAP the main program computes all local arrays which are needed to perform different tasks on the element level. The switch to different tasks in the element subroutines depends on the value of the variable 'isw'. All variables, parameters and arrays are transferred via the list of parameters of the element subroutines and via a restricted number of common blocks which are defined below.

14.1.1 Name of Subprogram, parameters

The name of an element subprogram for feap must be specified according to the following *name* and *parameter* requirements

```
subroutine elmtnn(d,ul,xl,ix,tl,s,p,h1,h2,h3,ndf,ndm,nst,isw)
```

where *nn* defines the element number in the range between 01 and 100.

The argument list is defined as follows together with dimensioning information:
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d(ndd)</td>
<td>material parameters</td>
</tr>
<tr>
<td>ul(ndf,1)</td>
<td>element displacement parameters</td>
</tr>
<tr>
<td>xl(ndm,1)</td>
<td>element initial coordinates</td>
</tr>
<tr>
<td>ix(1)</td>
<td>global/local node numbers</td>
</tr>
<tr>
<td>tl(1)</td>
<td>element temperatures</td>
</tr>
<tr>
<td>s(nst,nst)</td>
<td>element tangent arrays</td>
</tr>
<tr>
<td>p(nst)</td>
<td>element load vector</td>
</tr>
<tr>
<td>h1(nhmax)</td>
<td>element history data vector h1</td>
</tr>
<tr>
<td>h2(nhmax)</td>
<td>element history data vector h2</td>
</tr>
<tr>
<td>h3(nh3max)</td>
<td>element history data vector h3</td>
</tr>
<tr>
<td>ndf</td>
<td>number d.o.f. at each node (1 to 6)</td>
</tr>
<tr>
<td>ndm</td>
<td>coordinate dimension (1,2,3)</td>
</tr>
<tr>
<td>nst</td>
<td>size element array/vector</td>
</tr>
<tr>
<td>nnd</td>
<td>number of material data for element; default nnd = 50*</td>
</tr>
<tr>
<td>isw</td>
<td>control switch parameter</td>
</tr>
</tbody>
</table>

* The value of nnd can be changed by the mesh macro `feap`.

The following of the above arrays and values are input quantities and will be transferred from the main program FEAP:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d(ndd)</td>
<td>material parameters</td>
</tr>
<tr>
<td>ul(ndf,nel)</td>
<td>element displacement parameters</td>
</tr>
<tr>
<td>xl(ndm,nel)</td>
<td>element initial coordinates</td>
</tr>
<tr>
<td>ix(1)</td>
<td>global/local node numbers</td>
</tr>
<tr>
<td>tl(nel)</td>
<td>element temperatures</td>
</tr>
<tr>
<td>ndf</td>
<td>number d.o.f. at each node (1 to 6)</td>
</tr>
<tr>
<td>ndm</td>
<td>coordinate dimension (1,2,3)</td>
</tr>
<tr>
<td>nst</td>
<td>size element array/vector =ndf*nel</td>
</tr>
<tr>
<td>nnd</td>
<td>number of material data for element; default nnd = 50</td>
</tr>
<tr>
<td>isw</td>
<td>control switch parameter</td>
</tr>
</tbody>
</table>

**Warning:** Do not change these values, since they are also used for the computation of other elements of this material group in the global f.e. problem.
The following arrays are results of the computation on element level their values depend on the task to be performed, see definition of ‘isw’.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d(ndd)</td>
<td>material parameters (only isw = 1)</td>
</tr>
<tr>
<td>s(nst,nst)</td>
<td>element tangent array</td>
</tr>
<tr>
<td>p(nst)</td>
<td>element vector</td>
</tr>
<tr>
<td>h1(nhmax)</td>
<td>element history data vector</td>
</tr>
<tr>
<td>h2(nhmax)</td>
<td>element history data vector</td>
</tr>
<tr>
<td>h3(nh3max)</td>
<td>element history data vector</td>
</tr>
</tbody>
</table>

**Warning**: Only compute and store values within the range of these arrays. An erroneous specification of e.g. \( s(nst,nst+2) = \text{fact} \) may lead to unpredictable errors in FEAP.
### 14.1.2 Control Switch Parameter Values

The values of the control switch parameter, \( isw \), which define a task to be performed on element level are defined as:

<table>
<thead>
<tr>
<th>isw – Value</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Define material parameters in ( d(i) )</td>
</tr>
<tr>
<td>2</td>
<td>Check mesh for errors</td>
</tr>
<tr>
<td>3</td>
<td>Compute tangent array, ( s ), and residual, ( p )</td>
</tr>
<tr>
<td>4</td>
<td>Output element variables (e.g., stress)</td>
</tr>
<tr>
<td>5</td>
<td>Compute element mass; ( s = ) consist., ( p = ) lump.; geometrical matrix ( s )</td>
</tr>
<tr>
<td>6</td>
<td>Compute element residual vector in ( p )</td>
</tr>
<tr>
<td>7</td>
<td>Compute surface loading; ( s = ) surface tangent, ( p = ) surface load vector</td>
</tr>
<tr>
<td>8</td>
<td>Project element variables to nodes (e.g., stress)</td>
</tr>
<tr>
<td>9</td>
<td>Perform error analysis</td>
</tr>
<tr>
<td>10</td>
<td>Update variables within augmented lagrangian algorithm</td>
</tr>
<tr>
<td>11</td>
<td>Perform numerical differentiation within extended system</td>
</tr>
<tr>
<td>12</td>
<td>Compute damping matrix; ( s = ) consist., ( p = ) lump.</td>
</tr>
<tr>
<td>13</td>
<td>Plot element stress resultants for beam, axi-shell and – membrane elements</td>
</tr>
<tr>
<td>14</td>
<td>Plot element stresses at center of element without averaging</td>
</tr>
<tr>
<td>15</td>
<td>update ( H_2 \rightarrow H_1 ) on micro-level, similar to stre, until call of mateli3d</td>
</tr>
<tr>
<td>16</td>
<td>Compute J–integral</td>
</tr>
<tr>
<td>17</td>
<td>Compute stresses at nodes and layer boundaries</td>
</tr>
<tr>
<td>18</td>
<td>Compute director for shells and surfaces</td>
</tr>
<tr>
<td>19</td>
<td>Implex time step control</td>
</tr>
<tr>
<td>20</td>
<td>Plot stresses for sections of beam elements</td>
</tr>
<tr>
<td>21</td>
<td>Compute non–symmetric tangent for eigensolver <strong>ueig</strong></td>
</tr>
<tr>
<td>22</td>
<td>Element load vectors QLOA</td>
</tr>
<tr>
<td>23</td>
<td>REME,REST</td>
</tr>
<tr>
<td>24</td>
<td>Element data for PARAVIEW</td>
</tr>
</tbody>
</table>

You do not have to set the vectors ‘\( p \)’ and ‘\( s \)’ to zero within the element routine since FEAP zeros these arrays before transferring them to the element subroutine.
14.2 Common Blocks

FEAP uses several common areas to pass information to and from the main program. Each common block must have exactly the specified length and precisions in order for the program to work. The useful common blocks are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Parameters</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>/bdata/</td>
<td>o, head(20)</td>
<td>character*4</td>
</tr>
<tr>
<td>/cdata/</td>
<td>numnp, numel, nummat, nen, neq, ipr</td>
<td>integer</td>
</tr>
<tr>
<td>/eddata/</td>
<td>dm, a, ma, mct, iel, nel</td>
<td>real, integer</td>
</tr>
<tr>
<td>/evdata/</td>
<td>intyp, ibuck</td>
<td>integer</td>
</tr>
<tr>
<td>/hdata/</td>
<td>nh1, nh2, nh3, nhmi, nhmf</td>
<td>integer</td>
</tr>
<tr>
<td></td>
<td>gh1(<em>), gh2(</em>), gh3(*)</td>
<td>real</td>
</tr>
<tr>
<td>/iofile/</td>
<td>ior, iow</td>
<td>integer</td>
</tr>
<tr>
<td>/pdata6/</td>
<td>inord(100), ipord(40, 100)</td>
<td>integer</td>
</tr>
<tr>
<td>/pdata7/</td>
<td>ipb, ipma(40), ipla</td>
<td>integer</td>
</tr>
<tr>
<td>/pdata10/</td>
<td>cfp, xmaxf, xminf, scal, nfp, klay, ifor, flfp</td>
<td>real, integer, logical</td>
</tr>
<tr>
<td>/strnam/</td>
<td>np, istv, strsus(26)</td>
<td>integer, character*15</td>
</tr>
<tr>
<td>/prisdat/</td>
<td>nptyp, nprip(8)</td>
<td>integer (default: 1, [1, 2, 3, 4, 5, 6, 7, 8])</td>
</tr>
<tr>
<td>/fornam/</td>
<td>forsus(11)</td>
<td>character*15</td>
</tr>
<tr>
<td>/pdam/</td>
<td>iprd, ipld</td>
<td>integer</td>
</tr>
<tr>
<td>/prlod/</td>
<td>prop, a(6, 10), iexp(10), ik(10), npld</td>
<td>real, real, integer, integer, integer</td>
</tr>
</tbody>
</table>

The programmer of an element might want to use own common blocks within the element to transfer data to the diverse subroutines defining the element. In this case it is preferable to create names like e.g. /elnn1/ etc. where ‘nn’ is the element number. This avoids possible confusion with common blocks defined in FEAP.
The parameters of the common blocks listed on the previous page are defined as:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>o</td>
<td>page output control</td>
</tr>
<tr>
<td>head</td>
<td>page header/title</td>
</tr>
<tr>
<td>numnp</td>
<td>number of nodes in problem</td>
</tr>
<tr>
<td>numel</td>
<td>number of elements in problem</td>
</tr>
<tr>
<td>nummat</td>
<td>number of material sets</td>
</tr>
<tr>
<td>nen</td>
<td>maximum number of nodes on element</td>
</tr>
<tr>
<td>neq</td>
<td>number of active equations</td>
</tr>
<tr>
<td>ipr</td>
<td>precision of real variables (FEAP only)</td>
</tr>
<tr>
<td>dm</td>
<td>no longer used</td>
</tr>
<tr>
<td>n</td>
<td>number of current element processed</td>
</tr>
<tr>
<td>ma</td>
<td>material set number of current element</td>
</tr>
<tr>
<td></td>
<td>boundary type of surface loads (isw = 7)</td>
</tr>
<tr>
<td>mct</td>
<td>line counter control for output (isw = 4)</td>
</tr>
<tr>
<td></td>
<td>surface load type (isw = 7)</td>
</tr>
<tr>
<td>iel</td>
<td>current element type (nn)</td>
</tr>
<tr>
<td>nel</td>
<td>number of nodes on current element</td>
</tr>
<tr>
<td></td>
<td>number of nodes on current surface (isw = 7)</td>
</tr>
<tr>
<td>nh1</td>
<td>length of history arrays h1, h2</td>
</tr>
<tr>
<td>nh3</td>
<td>length of history arrays h3</td>
</tr>
<tr>
<td>np</td>
<td>pointer for nodal stress parameters</td>
</tr>
<tr>
<td>ior</td>
<td>input file unit number (if negative use *)</td>
</tr>
<tr>
<td>iow</td>
<td>output file unit number</td>
</tr>
<tr>
<td>istv</td>
<td>number of nodal stresses (default 8)</td>
</tr>
<tr>
<td></td>
<td>a negative value prevents calc. of principal stresses</td>
</tr>
<tr>
<td></td>
<td>( (\sigma_I = 5, \sigma_{II} = 6, \phi_{I} = 7) )</td>
</tr>
<tr>
<td>strsus(26)</td>
<td>array for description of plotted stresses eg. ( S_{11}, S_{22} )</td>
</tr>
<tr>
<td>forsus(11)</td>
<td>array for description of plotted forces eg. ( N_{11}, N_{22} )</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| imtyp     | 1: calculate mass matrix $\mathbf{M}$
            | 2: calculate geom. matrix $\mathbf{K}_G$
            | 3: calculate geom. matrix $\mathbf{K}_U + \mathbf{K}_G$
            | 4: calculate lin. matrix $\mathbf{K}_L$
| inord     | number of nodes of element |
| ipord     | series of node numbers to be plotted |
| ipb       | not used in element |
| ipma      | not used in element |
| ipla      | 0: standard displacement element ($u_1, u_2, u_3, \varphi_1, \varphi_2, \varphi_3$)
            | 1: plate element ($w, \varphi_1, \varphi_2$)
            | 2: beam, axi-shell/membrane element ($u_1, u_2, \varphi$) |
| cfp       | scaling factor for stress resultant plot |
| nfp       | number of stress resultant to be plotted
            | positive: calculate main stresses in 4-6 from 1-3 automatically
            | negative: stress values 1-11 can be defined free |
| xmaxf     | max. stress resultant value |
| xminf     | min. stress resultant value |
| flfp      | logical for switch between plotting stress resultants and min/max-search |
| klay      | layer number |
| ifor      | local plot direction 13 (default) or 12 |
| iprd      | pointer for damage print output (0=no,1=yes) |
| ipld      | pointer for damage plot output (0=no,1=yes) |
| prop      | actual load factor |
| a(6,npld) | $t_{\min}, t_{\max}, A_1 - A_4$, see macro $\text{prop}$ |
| iexp(npld)| exponent |
| ik(npld)  | loading type (1-3) |
| npld      | max. number of prop. cards=3 |
14.3 Treatment of history terms

FEAP provides options for each element to manage variables which must be saved during the solution. These are history variables and are separated into three groups:

a) Variables associated with the last converged solution time \( t_n \), stored in \( h_1 \)

b) variables associated with the current solution time \( t_{n+1} \), stored in \( h_2 \), and

c) variables which are not associated to any particular time, stored in \( h_3 \).

Examples for a) and b) are the integration of non-linear constitutive equations over a time step \( (\Delta t = t_{n+1} - t_n) \), examples for c) are e.g. terms used within mixed elements (e.g. based on Hu-Washizu-principles).

Before calling the element routine for each element, FEAP transfers the required history variables from global to local arrays \( h_1, h_2, h_3 \). Users may then access the history data for each element and if necessary update values and return them FEAP. Only for specific actions the local history data will be transferred back to the appropriate global locations.

14.3.1 Assigning amount of storage for each element

The specification for the amount of history information to be associated with each material set is controlled in the isw = 1 task of an element routine. For each material type specified within the element routine a value for the length of the ‘nh1’ and the ‘nh3’ data must be provided (the amount of ‘nh2’ data will be the same as for ‘nh1’). This is accomplished by setting the variables ‘nh1’ and ‘nh3’ in common hdata to the required values. That is, the statements required are:

```
USE hdata
...
if(isw .eq. 1) then
...
  nh1 = 24
  nh3 = 10
...
```

reserves 24 words of ‘nh1’ and ‘nh2’ data and 10 words of ‘nh3’ data for each element with the current material number. Typically it holds \( nh1 = ngaus \cdot nh \), with \( ngaus \) = number of integration points and \( nh \) = number of history terms at integration point. Care should be taken to minimize the number of history variables since, for very large problems, the memory requirements can become large, thus reducing the size of problem that FEAP can solve. Different materials/elements could have different amount of history data. For that FEAP calculates maximum values \( nhmax = max(nh1_i) \) and \( nh3max = max(nh3_i) \) and provide local arrays \( h1(nhmax), h2(nhmax), h3(nh3max) \). On element level these arrays are defined in general by

```
dimension h1(*), h2(*), h3(*)
```
14.3.2 Accessing history data for each element

As noted above the data for each element is contained in arrays whose first word is located at local arrays h1(1), h2(1). ‘nh1’-‘nh3’ are here no longer used. Note that arrays only exist if non-zero values are assigned to ‘nh1’ and/or ‘nh3’ during the isw = 1 task. Any other allocated data follows immediately after each first word. It is a user’s responsibility to manage what is retained in each variable type; however, the order of placing the \( t_n \) and \( t_{n+1} \) data into the h1- and h2-arrays should be identical. When elements are developed for FEAP it is the programmer’s responsibility to define and update the necessary information in the history arrays. Normally, information is read from the h1-array and written to the h2-array. The time-macro command is used to advance time and also is the command which redefines the information in the h1-array from that currently stored as the h2-array. Generally, it is not advisable to write information into the h1-array space as this may destroy information needed for subsequent iterations or solution step.

There are no provisions to store integer history variables separately from double precision quantities. It is necessary to cast the integer data as double precision and move to the history location. For example, using the statement \( h3(6) = \text{dble}(ivarbl) \) saves the value for the integer variable ivarbl in the sixth word of the h3 element history array. At a subsequent iteration for this element the value of the integer would be recovered as ivarbl = int(h3(6)). While this wastes storage for integer variables, experience indicates there is little need to save many integer quantities and, thus, it was not deemed necessary to provide for integer history variables separately. Although users may define new values for any of the h1, h2, or h3 types, the new quantities will be returned to the global arrays only for isw tasks where residuals are being formed for a solution step (i.e., solution command form, tang.,1, or utan.,1 and for history reinitialization during a time update (i.e., solution command, time). These access the task options isw equal to e.g. 3 or 6, respectively. Data will be not returned to the global arrays for isw tasks like e.g. stre. Otherwise each stre-command would lead to an update of h2. Thus, it is necessary to set the variables hflgu and h3flgu to true if an update is required, if no update is wanted the variables should be set to false. These variables are typically set in subroutine PMACR. Nothing has to be done on element level.

The mentioned parameters are located in

Module
USE hdatam

nhmax, nh3max, hflgu, h3f1gu
14.3.3 Typical 2D-element framework for history data

subroutine elmtnn(d,ul,xl,ix,t1,s,p,h1,h2,h3,ndf,ndm,nst,isw)
USE hdata
...  
if(isw.eq.1) then
  c... input material (examples)
  ngs = 2 ! no. of Gauss-points
  nlay= 4 ! no. of layers
  nhv = 3 ! no. of history terms at layer and Gauss point
  nh1 = ngs*ngs*nlay*nhv
  return
else if (isw.eq.3) then
  c.... stiffness matrix and residual
  ngs = 2
  nlay = 4
  nhv = 3
  nn = 0 ! counter history-arrays
  do 30 igs = 1,ngs*ngs ! loop Gauss points
     do 31 ilay = 1,nlay ! loop layer
        c.... read history data
        s1 = h1(nn+1)
        s2 = h1(nn+2)
        s3 = h1(nn+3)
        c.... modify history data (example)
        s1 = s1+1
        s2 = s1+2
        s3 = s1+3
        c.... store history data
        h2(nn+1) = s1
        h2(nn+2) = s2
        h2(nn+3) = s3
     do 33 ino =1,nel ! loop over node i
        p = p + p(s1,s2,s3) ! residual
     do 33 jno = ino,nel ! loop over node j
        s = s + s(s1,s2,s3) ! tangent
  33 continue
  32 continue
  nn = nn + nhv ! update counter history-arrays
  do 30 igs = 1,ngs*ngs ! loop Gauss points
     do 31 ilay = 1,nlay ! loop layer
        c.... read history data
        s1 = h1(nn+1)
        s2 = h1(nn+2)
        s3 = h1(nn+3)
        c.... modify history data (example)
        s1 = s1+1
        s2 = s1+2
        s3 = s1+3
        c.... store history data
        h2(nn+1) = s1
        h2(nn+2) = s2
        h2(nn+3) = s3
     do 33 ino =1,nel ! loop over node i
        p = p + p(s1,s2,s3) ! residual
     do 33 jno = ino,nel ! loop over node j
        s = s + s(s1,s2,s3) ! tangent
  33 continue
  32 continue
  nn = nn + nhv ! update counter history-arrays
  do 30 igs = 1,ngs*ngs ! loop Gauss points
     do 31 ilay = 1,nlay ! loop layer
        c.... read history data
        s1 = h1(nn+1)
        s2 = h1(nn+2)
        s3 = h1(nn+3)
        c.... modify history data (example)
        s1 = s1+1
        s2 = s1+2
        s3 = s1+3
        c.... store history data
        h2(nn+1) = s1
        h2(nn+2) = s2
        h2(nn+3) = s3
     do 33 ino =1,nel ! loop over node i
        p = p + p(s1,s2,s3) ! residual
     do 33 jno = ino,nel ! loop over node j
        s = s + s(s1,s2,s3) ! tangent
  33 continue
  32 continue
  nn = nn + nhv ! update counter history-arrays
  else if(isw.eq....) then
    ....
end if
end
14.4 Typical element framework

subroutine elmtnn(d,ul,xl,ix,t1,s,p,h1,h2,h3,ndf,ndm,nst,isw)
implicit double precision (a-h,o-z)
USE bdata
USE cdata
USE eldata
USE hdata
USE iofile
USE pdata6
USE strnam
if(isw.eq.1) then
   c Input/output of property data after command: 'mate'
   c d(ndd) stores information for each material set
   c Return: nh1 = number of nh1/nh2 words/element
   c Return: nh3 = number of nh3 words/element
else if(isw.eq.2) then
   c Check element for errors. Negative jacobian, etc.
else if(isw.eq.3) then
   c Return: Element coefficient matrix and residual
   c s(nst,nst) element coefficient matrix
   c p(nst) element residual
   c h1/h2 history data base: previous/current time step
   c h3 history data base: time independent
else if(isw.eq.4) then
   c Output element quantities (e.g. stresses)
   c n is the current element number, mct is a line counter
else if(isw.eq.5) then
   c Return: Element mass matrix
   c s(nst,nst) consistent matrix
   c p(nst) diagonal lumped matrix
else if(isw.eq.6) then
   c Compute residual only
   c p(nst) element residual
else if(isw.eq.7) then
   c Return: Surface loading for element
   c s(nst,nst) coefficient matrix
   c p(nst) nodal forces
else if(isw.eq.8) then
   c Compute stress projections to nodes (diagonal)
   call plotnn (ix,strea,strea(1+numnp)....)
   c strea(1+numnp) projection weight: dt
   c strea projection values: st
else if(isw.eq....) then
   ....
else if(isw.eq.22) then
   c Compute element loads
end if
end
14.5 Use of Parameters and Expression Inputs in New Elements

This is part of programming instructions for adding new elements or subprograms into FEAP. The routine DINPUT must always be used if expressions are to be included as part of the data.

The subroutine DINPUT may be used to input data in any new program module. This routine inputs the data using parameters and expressions as described in the introductory part of the FEAP manual. DINPUT returns them to the subprogram in a double precision array \( \text{td}(\text{nn}) \), see parameters of DINPUT:

\[
\text{subroutine dinput (td, nn)}.
\]

The following statements may be included as part of the routine performing the input.

\[
\begin{align*}
\text{USE errchk} \\
\text{USE iofile} \\
\text{real*8 td(5)} \\
\text{l} & \quad \text{if}(\text{ior.lt.0}) \text{write(*,3000)} \\
 & \quad \text{call dinput(td, 5)} \\
 & \quad \text{if}(\text{errck}) \text{go to 1}
\end{align*}
\]

The parameters defined in the common blocks are:

- \text{ior} - input file unit number (if negative, input from terminal).
- \text{iow} - output file unit number.
- \text{errck} - logical variable, if true and error occurred in dinput.
- \text{td} - a double precision variable to store values of input.

If any \( \text{td}(i) \) is to be used as an integer or real\( *\)4 quantity, it must be assigned to the correct variable. That is, the following operations must be performed.

\[
\begin{align*}
\text{real t} \\
\text{integer j}
\end{align*}
\]

\[
\begin{align*}
\text{call dinput (td, 5)} \\
\text{t} & \quad = \text{td(2)} \\
\text{j} & \quad = \text{td(1)}
\end{align*}
\]

will perform the reassignments. DINPUT may be used to input up to 16 individual expressions on one input record (each input record is, however, limited to 80 characters).
14.6 Error analysis

Stress and energy error indicators are calculated on element level under isw = 9. The indicators base on norms which are calculated from the differences between stresses at gauss points and stresses which have been projected to the nodes (based on stre,node see isw = 8). The indicator is known in the literature as $Z^2$-indicator (Zienkiewicz-Zhu).

Implemented forms can be found for the macro erro.

14.7 Logical Flags

<table>
<thead>
<tr>
<th>Flag</th>
<th>Initial value</th>
<th>Description</th>
<th>Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>fl(1)</td>
<td>False</td>
<td>True=consistent mass</td>
<td>CMAS</td>
</tr>
<tr>
<td>fl(2)</td>
<td>F</td>
<td>T=lumped mass</td>
<td>LMAS</td>
</tr>
<tr>
<td>fl(3)</td>
<td>T</td>
<td>F=memory for AL</td>
<td>UTAN</td>
</tr>
<tr>
<td>fl(4)</td>
<td>T</td>
<td>F=memory for AD,AU</td>
<td>TANG</td>
</tr>
<tr>
<td>fl(5)</td>
<td>T</td>
<td>F=memory for lump.M</td>
<td>LMAS</td>
</tr>
<tr>
<td>fl(6)</td>
<td>T</td>
<td>F=memory for cons.M</td>
<td>CMAS</td>
</tr>
<tr>
<td>fl(7)</td>
<td>T</td>
<td>T=no solution computed</td>
<td>REST</td>
</tr>
<tr>
<td>fl(8)</td>
<td>F</td>
<td>T=residual computed</td>
<td>FORM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TANG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UTANG</td>
</tr>
<tr>
<td>fl(9)</td>
<td>F</td>
<td>T=transient calculation</td>
<td>TRANS</td>
</tr>
<tr>
<td>fl(10)</td>
<td>F</td>
<td>T=new time step</td>
<td>TIME</td>
</tr>
<tr>
<td>fl(11)</td>
<td>F</td>
<td>T=stress projection computed</td>
<td>STRE,NODE</td>
</tr>
<tr>
<td>fl(12)</td>
<td>F</td>
<td>T=memory for BFGS</td>
<td>BFGS</td>
</tr>
<tr>
<td>arcf</td>
<td>F</td>
<td>T=arc length calculation</td>
<td>ARCL</td>
</tr>
<tr>
<td>refl</td>
<td>F</td>
<td>T=reactions computed</td>
<td>REAC</td>
</tr>
<tr>
<td>ctfI</td>
<td>F</td>
<td>T=contact solution</td>
<td>CONT</td>
</tr>
<tr>
<td>hadd</td>
<td>F</td>
<td>T=macro commands added to history</td>
<td></td>
</tr>
<tr>
<td>hflgu</td>
<td>T</td>
<td>F=history terms are not updated</td>
<td>STRE (ex.)</td>
</tr>
<tr>
<td>h3flgu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pfl</td>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pfr</td>
<td>T</td>
<td>F=supress diagnostic prints</td>
<td>PRIN</td>
</tr>
<tr>
<td>NOPR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pltfl</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zfg</td>
<td>T</td>
<td>F=complex solution</td>
<td></td>
</tr>
</tbody>
</table>
14.8 3D-Material library

It is possible to use every 3D-material implemented in the material library via an user interface, which is described in detail in the following.

The interface is adressed by

call matelib3d

(h1,h2,nh,d(i),ndd,EPS,SIG,CMAT,nsig,ntyp,plout,xgp,tgp,dvp,ddf,skfy,skfz,ngp,lgp,lay1gp,lay2gp,imat,isw)

Input of the material data is performed for isw=1. Within this procedure the number of history terms used at each Gauss-point is defined with the parameter nh. The total length of the history-arrays have to be defined in the element: e.g. \( nh1 = ngaus \cdot nh \). As output within this phase ndd defines the number of used input parameters, which can be used to check the length of d-array.

Further calls of matelib3d are for other values of isw. These have to be done within an element loop over the Gauss-Points. Based on the calculated strains EPS at this point with respect to a local cartesian base system the subroutine matelib3d returns the associated material matrix CMAT together with the stresses SIG.

Parameters are described in detail in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>h1(nh)</td>
<td>input</td>
<td>element history data vector h1 at Gauss-Point</td>
</tr>
<tr>
<td>h2(nh)</td>
<td>output</td>
<td>element history data vector h2 at Gauss-Point</td>
</tr>
<tr>
<td>nh</td>
<td>output</td>
<td>number of history terms at Gauss-Point</td>
</tr>
<tr>
<td>d(i)</td>
<td>in/output</td>
<td>array of material parameters, where i describes the position of material parameters in d-array</td>
</tr>
<tr>
<td>md</td>
<td>output</td>
<td>number of material data</td>
</tr>
<tr>
<td>EPS(*)</td>
<td>input</td>
<td>strain vector E</td>
</tr>
<tr>
<td>SIG(*)</td>
<td>output</td>
<td>stress vector S</td>
</tr>
<tr>
<td>CMAT(*)</td>
<td>output</td>
<td>tangent material matrix C</td>
</tr>
<tr>
<td>nsig</td>
<td>input</td>
<td>number of strain/stress components, typically=6</td>
</tr>
<tr>
<td>ntyp</td>
<td>input</td>
<td>type of element for A-matrix: 1=3D, 2=shell, 3=beam</td>
</tr>
<tr>
<td>plout(10)</td>
<td>output</td>
<td>internal parameter for plot</td>
</tr>
<tr>
<td>xgp(3)</td>
<td>input</td>
<td>coordinates of Gauss-Point</td>
</tr>
<tr>
<td>tgp</td>
<td>input</td>
<td>temperature load at Gauss-Point</td>
</tr>
<tr>
<td>dvp</td>
<td>input</td>
<td>det J \times \text{weighting factor at Gauss-Point}</td>
</tr>
<tr>
<td>detf</td>
<td>input</td>
<td>det F at Gauss-Point</td>
</tr>
<tr>
<td>skfy</td>
<td>input</td>
<td>\sqrt{\kappa} \quad \text{[for beams \sqrt{\kappa_y}]: sqrt. of shear correction factor}</td>
</tr>
<tr>
<td>skfz</td>
<td>input</td>
<td>not used \quad \text{[for beams \sqrt{\kappa_z}]: sqrt. of shear correction factor}</td>
</tr>
<tr>
<td>ngp</td>
<td>input</td>
<td>element number</td>
</tr>
<tr>
<td>lgp</td>
<td>input</td>
<td>Gauss-Point number</td>
</tr>
<tr>
<td>lay1gp</td>
<td>input</td>
<td>layer number</td>
</tr>
<tr>
<td>lay2gp</td>
<td>input</td>
<td>Gauss-Point number at layer</td>
</tr>
<tr>
<td>imat</td>
<td>input</td>
<td>number of material model to be used</td>
</tr>
<tr>
<td>isw</td>
<td>input</td>
<td>solution option from element</td>
</tr>
</tbody>
</table>
14.9 Adding elements for FE²

- **Update of History Parameter on micro-level**
  After convergence of the global load step an update of the history variables on micro-level is necessary. This is done using the macro `updh`. For this purpose code has to be added in the macro-element. For `isw=15` a loop over the integration points similar to `isw=4`(`stre,print`) or `isw=8`(`stre,plot`) is necessary. This loop could end after `call matelib3d(h1....)

- **Material number of element on macro-level**
  Up to 10 different micro-problems are possible. Thus it is necessary to know the material type number of the actual element. This needs

  a) USE `fe2mat`
  b) `isw=3,4,6,8,...`
     `matfe2=ma`
     before `call matelib3d(h1....)

- **Storage of restart files in FE²**
  The restart files of the micro-problem could be stored separately (irtyp=0) or in on global file (irtyp=1). In the latter case the local restart-file have to be extracted from the global file.

  a) USE `fe2tran`
  b) `isw=3,4,6,8,...` before Gauss-Point loop: extract all restart files of element
     `if(matn.eq.8.and.irtyp.eq.1) call matt3d08(1,n,numel,lint)`
  c) `isw=3,6,...` after integration-point loop: store all restart files of element
     `if(matn.eq.8.and.irtyp.eq.1) then`
     `if(hfigu.and.hf3figu) call matt3d08(2,n,numel,lint)`
     `end if`
14.10 Plot of user defined data

Two user defined values 'valuse1' and 'valuse2' can be plotted and stored with the tplo-macro. Plots are possible with respect to displacement or time. Values are set to zero within initializing FEAP.

To provide the data some additional code is necessary on element level.

**Example 1:** Energy of system

```plaintext
USE plodfu

e.g. for isw = 4

C.... calculate values, e.g.
    energy\_pot=....
    energy\_kin=....

    valuse1 = valuse1 + energy\_pot
    valuse2 = valuse2 + energy\_kin
```

**Example 2:** stress in el.x gp. y

```plaintext
USE plodfu
loop gp
    stress(x,y)=...
    if(n.eq.x and gp.eq.y) valuse1=stress(x,y)
end loop gp
```

The 'Run'-PCD-File has to be modified slightly to store and show the data.

```plaintext
e.g for one time step
time
loop,,20
tag,,1
next
stres,all (see above isw=4, for ex. 1)
stres,x (see above isw=4, for ex. 2)
tplo,us1d or
tplo,us2d or
tplo,us1t or
tplo,us2t
```

The user defined values 'valuse1' and 'valuse2' can be found also in the LDF-File after the macro `tplo,save` has been used.
14.11 Developing 2D-/3D-elements with higher order shape functions

This is implemented in element 21. Associated Shape functions for the given node numbering can be found in module P6shape.

On element level USE P6shape needs to be set. Then the interface routine shapeL=shape_lagrange is available. The node positions are at the boundary $\xi, \eta, \zeta = \pm 1$ and the gaussian points. To increase the order of approximation only the corresponding 1–D shape functions have to be added in shape1dL.

Usage:

<table>
<thead>
<tr>
<th>Case</th>
<th>Call statement</th>
<th>Informations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1–D</td>
<td>call shapeL(order,xsi,shp,dshp)</td>
<td>order: shape function order, xsi: parametric coordinate, shp: shape function, dshp: derivative of shp with respect to the parametric coordinate [optional]</td>
</tr>
<tr>
<td>2–D</td>
<td>call shapeL(order,xsi,eta,shp,dshp)</td>
<td>order: shape function order, xsi, eta: parametric coordinates, shp: shape function, dshp: derivative of shp with respect to the parametric coordinates [optional]</td>
</tr>
<tr>
<td>3–D</td>
<td>call shapeL(order,xsi,eta,zeta,shp,dshp)</td>
<td>order: shape function order, xsi, eta, zeta: parametric coordinates, shp: shape function, dshp: derivative of shp with respect to the parametric coordinates [optional]</td>
</tr>
</tbody>
</table>

Institut für Baustatik, KIT Cont. Mesh Macro Plot Mate3D Elem Nege Gmesh Cylt Isoge 538
Chapter 15

Theory Manual

15.1 Programming structure of FEAP

15.1.1 General structure

The general structure with the associated subroutines is depicted in the following figure.
15.1.2 Calculating arrays on element level

Within algorithms it is necessary to calculate arrays on element level. Examples are the tangent stiffness matrix (tang), print or plot of stress values (stre, stre) or the mass matrix (cmas). Thus, these values have to be calculated from macro or plot-level. The general procedure is described in principle in the following figure. Available data are governed by the parameter isw. Details can be found in the part 'Adding elements' (Control Switch Parameter Values).

15.1.3 Displacement arrays

It holds for node i and degree of freedom idf
\[ u \stackrel{\Delta}{=} u_l (idf,i) \]
\[ \Delta u \stackrel{\Delta}{=} u_l (idf, nen+i) \]
\[ \Delta \Delta u \stackrel{\Delta}{=} u_l (idf, 2 nen+i) \]
15.1.4 Making FEAP faster

Two strategies have been followed. These are (i) the use of Compiler options ’Optimize’ and (ii) the parallelization of the code. Within the parallelization the element loop, defined in SR PFORM, has been parallelized using OpenMP-techniques. Thus all tasks on element level, like element stiffness matrix, residual vector and stress calculation among others are calculated with respect to the number of processors. The main time consuming part of a linear/non-linear analysis is the solution process. Thus the use of a parallel-solver is necessary. For this purpose an interface to the direct parallel solver package PARDISO has been implemented, see the Input-macro solv. A schematic scheme of parallelization within the nonlinear time-/load-stepping algorithm is shown in the following figure.

15.2 Nonlinear Iteration Algorithms

Solutions of a nonlinear analysis could be found only iteratively. The following macros have to be used within a time/load step

- A) Newton Raphson algorithm
  
  \[
  \text{loop}, N \quad \text{tang} \quad \text{loop}, N \\
  \text{form} = \text{tang}, 1 \\
  \text{solv} \quad \text{next}
  \]

- B) modified Newton Raphson algorithm
  
  \[
  \text{tang} \quad \text{loop}, N \\
  \text{form} \quad \text{solv} \quad \text{next}
  \]
• C) BFGS algorithm
  \[ \text{tang} \]
  \[ \text{bfgs}, N \]

15.3 Stability analysis

In a stability analysis critical loads and associated buckling modes (eigenvectors) are calculated. Different versions are possible.

• A) Linear stability analysis at \( t = 0 \) (\( v = 0, P = 0 \))

\[
\begin{align*}
\# & \quad [K_L + \Lambda_0 K_{NL}] \Psi = 0 \\
\end{align*}
\]

- \( K_L \): linear stiffness matrix
- \( K_{NL} \): nonlinear part of stiffness matrix \( K_{NL} = K_T - K_L \)
- \( \Lambda_0 \): amplification factor (lowest value, to choose from solution of EV-problem)
- \( \Psi \): associated eigenvector
- \( P \): actual load
- \( P_0 \): incremental load
- \( P_{crit} = \Lambda_0 P_0 \) (estimation, correct in case of lin. prebuckling behavior)

\# Realisation in FEAP
  \[ \text{tang}, 1 \]
  \[ \text{geom} \]
  \[ \text{subs}, n \quad n: \text{number of eigenpairs} \]
  Note that \text{geom} destroys \( K_T \)!

• B) Linear stability analysis at \( t = 0 \) (\( v = 0, P = 0 \))

\[
\begin{align*}
\# & \quad [K_T - \omega 1] \varphi = 0 \\
\end{align*}
\]

- \( K_T \): tangent stiffness matrix
- \( 1 \): unity matrix
- \( \omega \): eigenvalue (lowest value, to choose from solution of EV-problem)
- \( \varphi \): associated eigenvector
- \( \Lambda_0 \): critical load with \( \Lambda_0 = \frac{\varphi^T K_L \varphi}{\varphi^T K_L \varphi - \omega \varphi^T \varphi} \)
- \( P_{crit} = \Lambda_0 P_0 \) (estimation, correct in case of lin. prebuckling behavior)

\# Realisation in FEAP
  \[ \text{tang}, 1 \]
  \[ \text{tang}, 1 \]
  \[ \text{idem} \]
  \[ \text{subs}, n \]
  \[ \text{lamb}, -1 \]
• A) Nonlinear stability analysis \( t = \bar{t} \) (\( v \neq 0, \ P = \lambda P_0 \neq 0 \))

\[
\begin{align*}
\# & \ [K_L + \Lambda K_{NL}] \varphi = 0 \\
K_L & \text{ linear stiffness matrix} \\
K_{NL} & \text{ nonlinear part of stiffness matrix } K_{NL} = K_T - K_L \\
\lambda & \text{ actual load factor, see macro } \text{prop} \rightarrow P = \lambda P_0 \\
\Lambda & \text{ amplification factor (lowest value, to choose from solution of EV-problem)} \\
\varphi & \text{ associated eigenvector} \\

P_{crit} = \Lambda(\lambda P_0) \text{ (estimation, correct for } \Lambda \rightarrow 1) \\
\end{align*}
\]

\# Realisation in FEAP

tang,,1
geom
subs,,n
Note that geom destroys \( K_T \)!

• B) Nonlinear stability analysis \( t = \bar{t} \) (\( v \neq 0, \ P = \lambda P_0 \neq 0 \))

\[
\begin{align*}
\# & \ [K_T - \omega 1] \varphi = 0 \\
K_T & \text{ tangent stiffness matrix} \\
1 & \text{ unity matrix} \\
\omega & \text{ eigenvalue (lowest value, to choose from solution of EV-problem)} \\
\varphi & \text{ associated eigenvector} \\
\end{align*}
\]

\# Estimation of the critical load \( \lambda = f(\omega) \)

The eigenvalue problem \([K_T - \omega 1] \varphi = 0\) leads to a critical load for \( \omega \rightarrow 0 \). This presents no 'engineering' information on the 'distance' to the critical load. Based on a comparison of both eigenvalue problems it holds in the limiting case:

\[
\Lambda = \frac{\varphi^T K_L \varphi}{\varphi^T K_L \varphi - \omega_i \varphi^T \varphi} \\
P_{crit} = \lambda P_0 \text{ (estimation, correct for } \Lambda \rightarrow 1) \\
\]

\# Realisation in FEAP

tang,,1
iden
subs,,n
lamb,all
15.4 Time integration procedures

15.4.1 Newmark–method

- Approach acceleration
  \[ \ddot{v}_{n+\gamma} = (1 - \gamma)\ddot{v}_n + \gamma\dot{v}_{n+1} \quad \text{for } \ddot{v} \]
  \[ \ddot{v}_{n+2\beta} = (1 - 2\beta)\ddot{v}_n + 2\beta\dot{v}_{n+1} \quad \text{for } v \]

- Integration of accelerations
  \[ \ddot{v}_{n+1} = (1 - \gamma)\ddot{v}_n\Delta t + \gamma\ddot{v}_{n+1}\Delta t + \ddot{v}_n \]
  \[ \ddot{v}_{n+1} = (1 - 2\beta)\ddot{v}_n\Delta t + 2\beta\ddot{v}_{n+1}\Delta t + \ddot{v}_n \]
  \[ \ddot{v}_{n+1} = \frac{1}{2} - \beta)\ddot{v}_n\Delta t^2 + \beta\ddot{v}_{n+1}\Delta t^2 + \ddot{v}_n\Delta t + \ddot{v}_n \]

- Update
  \[ \ddot{v}_{n+1} = \frac{1}{\beta\Delta t^2} \mu_{n+1} - \nu_n \frac{\Delta v_n}{\Delta n} + \frac{1}{\beta\Delta t} \dot{v}_n + \left( 1 - \frac{1}{2\beta} \right) \ddot{v}_n \]
  \[ \ddot{v}_{n+1} = \frac{\gamma}{\beta\Delta t} \mu_{n+1} - \nu_n \frac{\Delta v_n}{\Delta n} + \left( 1 - \frac{1}{\beta} \right) \dot{v}_n + \left( 1 - \frac{1}{2\beta} \right) \ddot{v}_n \Delta t \]

- Differential equations at \( n + 1 \)
  \[ M\ddot{v}_{n+1} + C\dot{v}_{n+1} + F_{\text{int}, n+1} = P_{n+1} \]

- Residual
  \[ G = M \left[ \frac{1}{\beta\Delta t^2} \mu_{n+1} - \nu_n \frac{\Delta v_n}{\Delta n} \right] + C \left[ \frac{\gamma}{\beta\Delta t} \mu_{n+1} - \nu_n \frac{\Delta v_n}{\Delta n} \right] + F_{\text{int}, n+1} - P_{n+1} \]

- Linearization for geometrical/material nonlinear problem
  \[ G^{i+1}(v_{n+1}^{i+1} + \Delta v_{n+1}^{i+1}) = \partial G_i^{i+1} \left( \frac{\partial v_n^{i+1}}{\partial v_{n+1}^{i+1}} \right) \Delta v_{n+1}^{i+1} + G_i^{i}(v_{n+1}^{i+1}) \approx 0 \]
  \[ \left[ \frac{1}{\beta\Delta t^2} M + \frac{\gamma}{\beta\Delta t} C + K_T^{i} \right] \Delta v_{n+1}^{i} = -G_i^{i} \]

- Newton algorithm
  1) \[ G^i = ... \]
  2) \[ \|G^i\| < \varepsilon \implies \text{stop} \]
  3) \[ K_{eff}^{i} \Delta v_{n+1}^{i} = -G_i^{i} \]
  4) \[ v_{n+1}^{i+1} = v_{n+1}^{i} + \Delta v_{n+1}^{i} \]
  5) \[ \dot{v}_{n+1}^{i+1} = \dot{v}_{n+1}^{i} + \frac{\gamma}{\beta\Delta t} \Delta v_{n+1}^{i+1} \]
  6) \[ \ddot{v}_{n+1}^{i+1} = \ddot{v}_{n+1}^{i} + \frac{1}{\beta\Delta t^2} \Delta v_{n+1}^{i+1} \]

- Associated FEAP- macros (1 time step)
  \[ \text{time} \]
  \[ \text{loop}_{n} \]
  \[ \text{tang}_{,1} \]
  \[ \text{next} \]
15.4.2 Overview Implicit Schemes

- Newmark

\[
\begin{align*}
G &= M \ddot{v}_{n+1} + C \gamma \dot{v}_{n+1} + F^\text{int}_{n+1} - P_{n+1} \\
\left[ \frac{1}{\beta t^2} M + \frac{\gamma}{\beta t} C + K_T \right] \Delta v_{n+1} &= -G
\end{align*}
\]

Remarks: unconditionally stable: \( \beta \geq \frac{1}{4}, \gamma \geq \frac{1}{2} \)
- dissipation: \( \gamma = \frac{1}{2} \) no numerical dissipation, \( \gamma > \frac{1}{2} \) numerical dissipation
- stability: \( \beta \geq \frac{1}{4} \) unconditionally stable, \( \beta = 0 \) explicit

- HHT (Hilber-Hughes-Taylor)

\[
\begin{align*}
G &= M \ddot{v}_{n+1} + C[(1 - \alpha_f) \dot{v}_{n+1} + \alpha_f \dot{v}_n] + (1 - \alpha_f) F^\text{int}_{n+1} - (1 - \alpha_f) P_{n+1} \\
\left[ \frac{1}{1 - \alpha_f} \frac{1}{\beta t^2} M + \frac{\gamma}{\beta t} C + K_T \right] \Delta \dot{v}_{n+1} &= -G
\end{align*}
\]

Remarks: \( G = \frac{\dot{G}}{1 - \alpha_f}, \beta = \frac{1}{4}(1 + \alpha_f)^2, \gamma = \frac{1}{2} + \alpha_f, \ 0 \leq \alpha \leq 1/3 \)

- Generalized alpha

\[
\begin{align*}
G &= M[(1 - \alpha_m) \ddot{v}_{n+1} + \alpha_m \ddot{v}_n] + C[(1 - \alpha_f) \dot{v}_{n+1} + \alpha_f \dot{v}_n] + (1 - \alpha_f) F^\text{int}_{n+1} - (1 - \alpha_f) P_{n+1} \\
\left[ \frac{1 - \alpha_m}{1 - \alpha_f} \frac{1}{\beta t^2} M + \frac{\gamma}{\beta t} C + K_T \right] \Delta \dot{v}_{n+1} &= -G
\end{align*}
\]

Remarks: \( G = \frac{\dot{G}}{1 - \alpha_f}, \beta = \frac{1}{4}(1 - \alpha_m + \alpha_f)^2, \gamma = \frac{1}{2} - \alpha_m + \alpha_f \)
- numerical damping for high and low frequencies
- against oscillations
- stabilize nonlinear calculations

15.4.3 Explicit Schemes

- Explicit algorithm 1 (Taylor)

\[
\begin{align*}
\dot{v}_{n+1} &= \dot{v}_n + \frac{\Delta t}{2} (\ddot{v}_n + \ddot{v}_{n+1}) = \dot{v}_n + \frac{\Delta t}{2} \ddot{v}_n + \frac{\Delta t}{2} \ddot{v}_{n+1} \\
\ddot{v}_{n+1} &= \ddot{v}_n + \Delta t \dddot{v}_n + \frac{\Delta t^2}{2} \dddot{v}_n \\
\end{align*}
\]

# Approach velocity, displacement

\[
\begin{align*}
G &= M \ddot{v}_{n+1} + C \dddot{v}_{n+1} + F^\text{int}_{n+1} - P_{n+1} \\
G &= M \ddot{v}_{n+1} + C(\dddot{v}_n^0 + \frac{\Delta t}{2} \dddot{v}_{n+1}) + F^\text{int}_{n+1} - P_{n+1}
\end{align*}
\]

# Residual

\[
G = M \ddot{v}_{n+1} + C \dddot{v}_{n+1} + F^\text{int}_{n+1} - P_{n+1}
\]

# Solution (no iteration!)

\[
(M + \frac{\Delta t}{2} C) \dddot{v}_{n+1} = -F^\text{int}_{n+1} + P_{n+1} - C \dddot{v}_n^0
\]

For \( M = [M^{ii}], C = [C^{ii}] \) (lumped matrices)

\[
\dddot{v}_{n+1}^{(i)} = \frac{1}{(M^{ii} + \frac{\Delta t}{2} C^{ii})} \left[ -F^\text{int}_{n+1}^{(i)} + P_{n+1}^{(i)} - C^{(ii)} \dddot{v}_n^{(i)} \right]
\]
• Explicit algorithm 2 (Verlet) Standard in LS-Dyna, Abaqus,....

# Approach velocity, displacement
\[ \dot{v}_{n+1/2} = \dot{v}_{n-1/2} + \Delta t \ddot{v}_n \text{ mit } \Delta t = \frac{t_{n+1} - t_n}{2} \text{ und } \Delta t = t_{n+1} - t_n \]

# Residual
\[ G = M \ddot{v}_{n+1} + C \ddot{v}_{n+1} + F_{\text{int},n+1} - P_{n+1} \approx M \ddot{v}_{n+1} + C \ddot{v}_{n+1/2} + F_{\text{int},n+1} - P_{n+1} \]

# Solution (no iteration!)
\[ M \ddot{v}_{n+1} = -F_{\text{int},n+1} + P_{n+1} - C \dot{v}_{n+1/2} \]

15.4.4 Energy-conserving algorithm

• Energy conserving: \( C = 0 \)

• Approach velocity, acceleration for \( \alpha = 1/2 \)
\[ \ddot{v}_{n+\alpha} = \frac{\dot{v}_{n+1} + \dot{v}_n}{2} = \frac{\dot{v}_{n+1} - \dot{v}_n}{\Delta t} \]
\[ \ddot{v}_{n+\alpha} = \frac{\ddot{v}_{n+1} + \ddot{v}_n}{2} = \frac{\ddot{v}_{n+1} - \ddot{v}_n}{\Delta t} \]

• Approach velocity, acceleration at end of time step
\[ \ddot{v}_{n+1} = \frac{2}{\Delta t} \left( \frac{v_{n+1} - v_n}{\Delta v_{n+1}} \right) \ddot{v}_{n+1} \]
\[ \ddot{v}_{n+1} = \frac{4}{\Delta t^2} \left( \frac{v_{n+1} - v_n}{\Delta v_{n+1}} \right) \ddot{v}_{n+1} \]
\[ \ddot{v}_{n+1} = \frac{2}{\Delta t^2} \left( \frac{v_{n+1} - v_n}{\Delta v_{n+1}} \right) \ddot{v}_{n+1} \]

• Residual at \( n + \alpha \)
\[ G_{n+\alpha} = M \ddot{v}_{n+\alpha} + F_{\text{int},n+\alpha} - P_{n+\alpha} \]
\[ G_{n+\alpha} = \frac{2}{\Delta t^2} M \left( v_{n+1} - v_n \right) - \frac{2}{\Delta t} M \ddot{v}_n + F_{\text{int},n+\alpha} - P_{n+\alpha} \]

• Iteration at \( n + \alpha \)
\[ C^{i+1}_{n+\alpha} = G^i_{n+\alpha} + \left( \frac{2}{\Delta t^2} M + K^i_T \right) \Delta v^{i+1}_{n+1} \]
• **Update**

\[
\begin{align*}
\mathbf{v}_{n+1}^{i+1} &= \mathbf{v}_{n+1}^{i} + \Delta \mathbf{v}_{n+1}^{i} \\
\mathbf{\dot{v}}_{n+1}^{i+1} &= \mathbf{\dot{v}}_{n+1}^{i} + \frac{2}{\Delta t} \Delta \mathbf{v}_{n+1}^{i} \\
\mathbf{\ddot{v}}_{n+1}^{i+1} &= \mathbf{\ddot{v}}_{n+1}^{i} + \frac{4}{\Delta t^2} \Delta \mathbf{v}_{n+1}^{i} \\
\mathbf{\dddot{v}}_{n+1}^{i+1} &= \mathbf{\dddot{v}}_{n+1}^{i} + \frac{2}{\Delta t^2} \Delta \mathbf{v}_{n+1}^{i}
\end{align*}
\]

• **External forces at** \(n + \alpha\)

\[
\mathbf{P}_{n+\alpha} = \frac{1}{2}(\mathbf{P}_{n+1} + \mathbf{P}_t)
\]

• **Internal forces at** \(n + \alpha\)

\[
\mathbf{F}_{n+\alpha}^{int} = \sum_{e=1}^{numel} \int_{\Omega} \mathbf{B}_{n+\alpha}^T \sigma_{n+\alpha} \, d\Omega, \quad \sigma_{n+\alpha} = \mathbf{C} \epsilon_{n+\alpha}, \quad \epsilon_{n+\alpha} = \frac{1}{2}(\epsilon_{n+1} + \epsilon_t)
\]

• **Tangent stiffness matrix at** \(n + \alpha\)

\[
\mathbf{K}_T = \sum_{e=1}^{numel} \int_{\Omega} \mathbf{B}_{n+\alpha}^T \mathbf{C}_1 \frac{1}{2} \mathbf{B}_{n+1} \, d\Omega + \mathbf{K}_0^T(\sigma_{n+\alpha}) \text{ not symmetric!}, \text{ thus } \mathbf{utan} \text{ must be used.}
\]

• Programming on element-level for isw = 3 (\(\mathbf{K}_T\)) and 6 (\(\mathbf{F}^{int}\)) necessary!

use module ddata with theta(4),nrk,nrc,nrm,nrt,nop. nop = 5 \(\sim\) energy-conserving algorithm

15.4.5 **Implicit composite scheme - Bathe**

• First substep: see Newmark-method with \(2 \cdot \beta = \gamma = \frac{1}{2}\)

• Second substep based on backward difference scheme

\[
\begin{align*}
\mathbf{v}_{n+1} &= a_1 \cdot \mathbf{v}_n + a_2 \cdot \mathbf{v}_{n+\alpha} + a_3 \cdot \mathbf{v}_{n+1} \\
\mathbf{\dot{v}}_{n+1} &= a_1 \cdot \mathbf{\dot{v}}_n + a_2 \cdot \mathbf{\dot{v}}_{n+\alpha} + a_3 \cdot \mathbf{\dot{v}}_{n+1} \\
\mathbf{\ddot{v}}_{n+1} &= \frac{1 - \alpha}{\alpha \cdot \Delta t} \mathbf{\dot{v}}_{n+1} - \frac{1}{(1 - \alpha) \cdot \alpha \cdot \Delta t} \mathbf{\ddot{v}}_n + \frac{2 - \alpha}{(1 - \alpha) \cdot \Delta t} \mathbf{\dddot{v}}_n
\end{align*}
\]

\[
\begin{align*}
\mathbf{v}_{n+1} &= \mathbf{v}_{n+\alpha} \cdot \left(\mathbf{v}_{n+1} - \mathbf{v}_{n+\alpha}\right) \\
\mathbf{\dot{v}}_{n+1} &= \mathbf{\dot{v}}_n + (a_2 + a_3) \cdot \mathbf{\dot{v}}_{n+\alpha} + a_3^2 \cdot \mathbf{\ddot{v}}_{n+\alpha} \\
\mathbf{\ddot{v}}_{n+1} &= \mathbf{\ddot{v}}_n + (a_2 + a_3) \cdot \mathbf{\dddot{v}}_{n+\alpha} + a_3 \cdot \mathbf{\dddot{v}}_{n+1}
\end{align*}
\]
15.4.6 Automatic time stepping procedure

In dynamic analysis when implicit integration is used, the automatic time stepping is based on the concept of half-step residuals (Hibbitt and Karlsson, 1979). The basic idea is that the time stepping operator defines the velocities and accelerations at the end of the step \( n + 1 \) in terms of displacement at the end of the step and conditions at the beginning of the step. Equilibrium is then established at \( n + 1 \) which ensures an equilibrium solution at the end of each time step and, thus, at the beginning and end of any individual time step. However, these equilibrium solutions do not guarantee equilibrium throughout the step. The time step control is based on measuring the equilibrium error (the force residuals) at some point during the time step, by using the integration operator, together with the solution obtained at \( n + 1 \), to interpolate within the time step. The evaluation is performed at the half step \( n + 1/2 \). If the maximum entry in this residual vector—the maximum 'half-step residual'—is greater than a user-specified tolerance, the time step is considered to be too big and is reduced by an appropriate factor. If the maximum half-step residual is sufficiently below the user-specified tolerance, the time step can be increased by an appropriate factor for the next increment. Otherwise, the time step is deemed adequate.

The algorithm is purely empirical, but experience shows that it works quite well, most especially in initially excited problems with high dissipation.

The half-step residual is based on the assumption that the accelerations vary linearly over the time interval (this is the basis of Newmark’s formulae)

\[
\ddot{v}_{n+\tau} = (1 - \tau)\ddot{v}_n + \tau \ddot{v}_{n+1} \quad \text{for } 0 \leq \tau \leq 1
\]

Having already solved for the state at \( n + 1 \), this equation, together with Newmark’s formulae now written for the time interval from \( n \) to \( n + 1 \), requires that

\[
\Delta v_{n+\tau} = \tau^3 \Delta v_n + \tau(1 - \tau^2)\Delta t \dot{v}_n + \tau^2(1 - \tau)\frac{\Delta t^2}{2} \ddot{v}_n
\]

\[
\dot{v}_{n+\tau} = \frac{\gamma}{\beta \Delta t} \Delta v_{n+\tau} + (1 - \frac{\gamma}{\beta})\dot{v}_n + (1 - \frac{\gamma}{2\beta})\tau \Delta t \ddot{v}_n
\]

\[
\ddot{v}_{n+\tau} = \frac{1}{\beta \Delta t^2} \Delta v_{n+\tau} - \frac{1}{\beta \Delta t} \ddot{v}_n + (1 - \frac{1}{2\beta})\dot{v}_n
\]

with \( \Delta v_n = \Delta v_{n+1} - \Delta v_n \). At the mid interval it holds \( \tau = 1/2 \), which leads to

\[
\Delta v_{n+1/2} = \frac{1}{8} v_n + \frac{3}{8} \Delta t \dot{v}_n + \frac{1}{16} \Delta t^2 \ddot{v}_n
\]

\[
\dot{v}_{n+1/2} = \frac{2\gamma}{\beta \Delta t} \Delta v_{n+1/2} + (1 - \frac{\gamma}{\beta})\dot{v}_n + \frac{1}{2}(1 - \frac{\gamma}{2\beta})\Delta t \ddot{v}_n
\]

\[
\ddot{v}_{n+1/2} = \frac{4}{\beta \Delta t^2} \Delta v_{n+1/2} - \frac{2}{\beta \Delta t} \ddot{v}_n + (1 - \frac{1}{2\beta})\dot{v}_n
\]

The algorithm could be summarized as follows

Suggest that, if \( P \) is a typical magnitude of real forces in an undamped elastic system (for which the high frequency response must be modeled reasonably accurately), then

- if \( G_{n+1/2} \approx 0.1P \) consistently, the time stepping solution has high accuracy;
- if \( G_{n+1/2} \approx P \) consistently, the time stepping solution has moderately good accuracy;
- if \( G_{n+1/2} \approx 10P \) consistently, the time stepping solution is rather coarse.

Problems with large amounts of natural dissipation of energy, such as elastic-plastic systems, are usually less sensitive to time step choice than purely elastic problems, because the energy that appears in higher
frequency modes is quickly dissipated. In such cases maximum half-step residuals in the range of 1–10 times typical forces indicate quite acceptable accuracy for most studies, and even values of 10–100 times typical forces can give useful results for primary effects, such as overall deformation. Thus, the method can offer relatively cost-effective solutions for highly dissipative systems for which we require only moderately accurate prediction of the overall response.

The use of the algorithm is described in detail in the following.

\[
\text{Do I = 1, NTIME} \\
\text{Set } \Delta t \\
\text{Set } t = t + \Delta t \\
\text{Calculate at time } t + \Delta t \\
\quad \dot{v}_{t+\Delta t}, \ddot{v}_{t+\Delta t}, \dddot{v}_{t+\Delta t} \\
\text{using time integration scheme until } ||G_{t+\Delta t}|| = 0 \\
\text{Calculate at time } t + \Delta t/2 \\
\quad \dot{v}_{t+\Delta t/2}, \ddot{v}_{t+\Delta t/2}, \dddot{v}_{t+\Delta t/2} \\
\text{Calculate half step residual } \\
\quad G_{t+\Delta t/2}, \text{ and } S = \max |G_{t+\Delta t/2}| \\
\text{if } S > T \text{ yes} \rightarrow i = 0 \rightarrow \Delta t_{\text{new}} = D_A \frac{T}{S} \Delta t \text{ no} \\
\text{if } S < WT \text{ no} \rightarrow i = 0 \\
\text{if } S < W \text{ yes} \rightarrow i = i + 1 \\
\text{if } i \geq 3 \text{ no} \\
\Delta t_{\text{new}} = \min (D_G \frac{T}{S} \Delta t, D_M \Delta t) \\
\text{if } i \geq 3 \text{ yes} \rightarrow \Delta t_{\text{new}} = \Delta t \\
\text{END DO}
\]

Input: \( D_A (=0.85), D_G (=0.80), D_M (=1.25), W (=0.75), S, T \)
15.5 PCG-Methods

Preconditioned Conjugate Gradient methods are used to solve the problem:

\[ \mathbf{K} \cdot \mathbf{u} = \mathbf{f} \]

The simplest algorithm is based on the idea of minimizing the function

\[ f(x) = \frac{1}{2} \mathbf{u} \cdot \mathbf{K} \cdot \mathbf{u} - \mathbf{f} \cdot \mathbf{u} \]

The function is minimized when its gradient

\[ \nabla f = \mathbf{K} \cdot \mathbf{u} = \mathbf{f} \]

is zero. The minimization is carried out by generating a succession of search directions \( \mathbf{p}_k \) and improved minimizers \( \mathbf{u}_k \). At each stage a quantity \( \alpha_k \) is found that minimizes

\[ f(\mathbf{u}_k + \alpha \mathbf{p}_k) \]

The efficiency of the algorithm is based on an appropriate predconditioning.
A generalization is the bi-conjugate gradient method which is valid for non positive definite matrices $K$ including preconditioning. One possible algorithm is:

1. Initial guess solution vector $u_0$

2. Initial residual vectors
   
   \[
   r_0 = f - K \cdot u_0 \\
   \bar{r}_0 = r_0
   \]

   Initial preconditioning
   
   \[
   M \cdot z_0 = r_0 \\
   M^T \cdot \bar{z}_0 = \bar{r}_0
   \]

   Initial direction vectors
   
   \[
   p_0 = z_0 \\
   \bar{p}_0 = \bar{z}_0 \\
   k = 0
   \]

3. if $p_k = 0$ or $r_k = 0$ stop. $u_k$ is the solution of $K \cdot u = f$.

4. if not, then
   
   new solution vector
   
   \[
   u_{k+1} = u_k + \alpha_k p_k
   \]
   
   with
   
   \[
   \alpha_k = \frac{\bar{r}_k \cdot z_k}{p_k \cdot K \cdot p_k}
   \]

   new residual vectors
   
   \[
   r_{k+1} = r_k - \alpha_k z_k \\
   \bar{r}_{k+1} = \bar{r}_k - \alpha_k \bar{z}_k
   \]

   new preconditioned vectors
   
   \[
   K \cdot p_k = z_{k+1} \quad \text{(calculated in $\alpha_k$)} \\
   K^T \cdot \bar{p}_k = \bar{z}_{k+1}
   \]

   new direction vectors
   
   \[
   p_{k+1} = z_{k+1} + \beta_k p_k \\
   \bar{p}_{k+1} = \bar{z}_{k+1} + \beta_k \bar{p}_k
   \]
   
   with
   
   \[
   \beta_k = \frac{r_{k+1} \cdot z_{k+1}}{r_k \cdot z_k}
   \]

5. Set $k \rightarrow k + 1$ and go to 3.

Tolerances:

\[
\text{itol} = 1 \quad \frac{||K \cdot x - f||}{||f||} < tol
\]

\[
\text{itol} = 2 \quad \frac{||M^{-1}(K \cdot x - f)||}{||M^{-1}f||} < tol
\]

\[
\text{itol} = 3 \quad \approx ||u|| < tol
\]

\[
\text{itol} = 4 \quad \approx u_{i \ max} < tol
\]
15.6 PGMRES-Methods

An excellent iterative equation solver for large linear equation systems with a general unsymmetric and not necessarily positive definite coefficient matrix is the Preconditioned General Minimum RESidual algorithm, which has been proposed in the current form by Saad (1981)\[19\] and Saad (1993)\[20\]. This algorithm is slightly slower than the CG algorithm for well conditioned symmetric positive definite matrices, and it also requires a lot more memory which depends on the choice of the dimension $m$ of the KRYLOV subspace, but it is far more stable and also works for badly conditioned coefficient matrices much better than the CG algorithm. The preconditioned PGMRES algorithm with modified GRAMSCHMIDT orthogonalisation and the KRYLOV subspace (dimension $m$) reads for solving $K \cdot u = f$

1. Initial guess $u_0$ for the solution

2. $r_0 = f - K \cdot u_0$
   
   $c = M^{-1} \cdot r_0$
   
   $\beta = ||c||_2$
   
   $v_1 = \frac{c}{\beta}$

   For $j = 1, \ldots, m$
   
   $w_j = M^{-1} \cdot K \cdot v_j$

   For $i = 1, \ldots, j$
   
   $h_{ij} = w_j \cdot v_i$

   $w_j = w_j - h_{ij} v_i$

   end

   $h_{j+1,j} = ||w_j||_2$

   If $h_{j+1,j} = 0$ then set $m = j$ and go to 3

   $v_{j+1} = \frac{w_j}{h_{j+1,j}}$

   end

3. Define HESSENBERG matrix

   $H_m = \{h_{ij}\}_{1 \leq i \leq m+1, 1 \leq j \leq m}$

4. Compute $y_m$ which minimizes $||\beta e_1 - H_m \cdot y||_2$ by applying GIVENS rotations to transform the HESSENBERG matrix into an upper triangular matrix. Here, $e_1$ is the $m + 1$ dimensional unit normal vector pointing in the 1-direction.

5. $u_m = u_0 + V_m \cdot y_m$ with $V_m = (v_1, \ldots, v_m)$

6. If $||f - K \cdot u_m||_2 > \text{tol}$ then set $u_0 = u_m$ and go to 2

There are many different variations of the PGMRES algorithm which focus on different aspects. More detailed information about solving large linear equation systems and especially about the PGMRES algorithm can be found in Saad (2003)\[21\]. http://www-users.cs.umn.edu/~saad/books.html
15.7 Error Analysis

The theoretical background may be found in Zienkiewicz/Taylor: The Finite Element Method I, 4.th ed., chapter 14.

The basic idea of error analysis is to have the same amount of error in all elements of the FE-mesh.

Thus, the element error is compared to a certain percentage of \( \frac{1}{numel} \times \text{system energy} \).

To do that we calculate stresses at Gauss-Point \( S_h \), averaged stresses at nodes \( S^* \) and furthermore stress differences \( \Delta S = (S^* - S_h) \).

At present three indicators are implemented using erro on macro or , erro on plot level.

1. Energy-norm
2. \( L_2 \)-norm
3. Equivalent stress \( \sigma_v \)

Norms 1 and 2 are used in the following way:

'System energy values' \[ ||u_1||_\Omega = \left( \sum_{e=1}^{numel} \int_{\Omega_e} (S_h^T D^{-1} S_h) \, d\Omega_e \right)^{1/2} \]

\[ ||u_2||_\Omega = \left( \sum_{e=1}^{numel} \int_{\Omega_e} (S_h^T S_h) \, d\Omega_e \right)^{1/2} \]

'Averaged element energy' \[ ||u||_{\Omega_e} = \frac{||u_n||_{\Omega_e}}{\sqrt{numel}} \]

'relative element energy' \[ ||e_1||_{\Omega_e} = \left( \int_{\Omega_e} [\Delta S^T D^{-1} \Delta S] \, d\Omega_e \right)^{1/2} \]

\[ ||e_2||_{\Omega_e} = \left( \int_{\Omega_e} [\Delta S^T \Delta S] \, d\Omega_e \right)^{1/2} \]

'accepted error in element' \[ \bar{e} = \frac{n1}{100} \cdot ||u||_{\Omega_e} \]

which means an accepted error of 5%.

\[ \xi_{\Omega_e} = \frac{||e_n||_{\Omega_e}}{\bar{e}} \]

If all values \( \xi_{\Omega_e} \) are less than 1 no refinement is necessary. A value of \( \xi_{\Omega_e} = 3 \) indicates that the permissible error is violated in this element by a factor 3.
Norms 3 is used to compare $\sigma_v$ on element level with $Y_0$ as indicator for plasticity:

'System energy values' $\|u_3\|_\Omega = \left( \sum_{e=1}^{numel} Y_0 \cdot Y_0 \right)^{1/2} = Y_0 \sqrt{numel}$

'Averaged element energy' $\|u\|_{\Omega_e} = Y_0 \sqrt{numel \sqrt{numel}} = Y_0$

'relative element energy' $\|e_3\|_{\Omega_e} = (\sigma_{ve})^{1/2} = \sigma_{ve}$

'accepted error in element' $\tilde{e} = \frac{n1}{100} \cdot ||u||_{\Omega_e}$ e.g. n1=5 which means an accepted error of 5% of $Y_0$

Based on these errors different further options are possible

• choose a mesh refinement with `reme,adap`
• use different elements in a FE-mesh, e.g. `Elmt05` and `Elmt06` with `mate,new`
• use different material models in one element in a FE-mesh, e.g. elasticity and plasticity with `mate,new`

### 15.8 Eigenvalue Computations

#### 15.8.1 Subspace Iteration
15.8.2 Lanczos Iteration

Lanczos method with complete reorthogonalization

Starting vector $x$

normalize $x_1 = \frac{1}{\gamma} x$ with $\gamma = x^T M x$

start value $\beta_0 = 0$

for $i = 1, ..., n$

$K\tilde{x}_i = M\tilde{x}_i$

$\alpha_i = \tilde{x}_i^T M x_i$

$\tilde{x}_i' = \tilde{x}_i - \alpha_i x_i - \beta_{i-1} x_{i-1}$

$\tilde{x}_i = \tilde{x}_i' - \sum_{k=1}^{i} (\tilde{x}_i' M x_k) x_k$

$\beta_i = (\tilde{x}_i^T M x_i)^{\frac{1}{2}}$

$x_{i+1} = \frac{\tilde{x}_i}{\beta_i}$

$T_{(i)} \tilde{\varphi}_{(i)} = \tilde{\lambda}_{(i)} \tilde{\varphi}_{(i)}$

if $\lambda_{(i)} - \lambda_{(i-1)} > \varepsilon$ go to $i$
\[ \varphi = X\tilde{\varphi} \quad \lambda = \frac{1}{\lambda} \]

Some theory:

it holds (theoretically) \[ x_i^T M x_j = \delta_{ij} \]

define tridiagonal matrix \[ X_n^T (MK^{-1}M) X_n = T_n \] with \[ T_n = \begin{bmatrix} \alpha_1 & \beta_1 & & & & \beta_1 & \alpha_2 & \beta_2 \\ \beta_1 & \alpha_2 & \beta_2 & & & & \ddots & \ddots & \ddots \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \beta_1 & \alpha_{n-1} & \beta_{n-1} & & & \end{bmatrix} \]

and \[ X_n = [x_1, x_2, \ldots, x_n] \]

eigenvalue problem \[ K\varphi = \lambda M\varphi \sim \frac{1}{\lambda} M\varphi = MK^{-1}M\varphi \]

transformation \[ \varphi = X_n\tilde{\varphi} \]

leads to \[ T_n\tilde{\varphi} = \frac{1}{\lambda} \tilde{\varphi} = \tilde{\lambda}\tilde{\varphi} \quad \lambda = \frac{1}{\tilde{\lambda}} \]

application in practice \[ m \ll n \quad T_m\tilde{\varphi} = \tilde{\lambda}_m\tilde{\varphi} \sim \text{lowest values of } \lambda_n \approx \frac{1}{\lambda_m} \]
15.9 Augmented Lagrange Formulation

In a calculation - e.g. including contact - an iteration is necessary. The iteration behavior can be improved with the Augmented Lagrange Formulation, which reduces the dependency on the ’correct’ choice of the penalty parameters. This will be achieved by an update of the contact forces with the macro **augm**.

The following algorithms are possible.

**Classical version**

Here an update is performed at the end of iteration. This is correct from mathematical point of view and leads within the iterations to quadratic convergence behaviour. To have this behaviour an additional loop is necessary!

```plaintext
loop.,n  time step
  time
  loop.,m  augm
    loop.,k  iteration
      tang.,1
      next
      augm
      next
next
```

**Alternative version**

Here an update is performed within the iteration. This may be in general faster and is practical from an engineering point of view. The additional loop is not necessary but the convergence behaviour is only linear.

```plaintext
loop.,n  time step
  time
  loop.,m  iteration
    tang.,1
    augm
    next
next
```
### 15.10 Theory of element formulations

#### 15.10.1 3D-Material library

The 3D-material library can be used e.g. with ELMT21, which is an 8/20/27–node geometrical linear/nonlinear 3D-Solid Element.

- **standard stresses and strains and material model:**

\[
\begin{bmatrix}
S_{11} \\
S_{22} \\
S_{33} \\
S_{12} \\
S_{13} \\
S_{23}
\end{bmatrix}
= \mathbf{C}
\begin{bmatrix}
E_{11} \\
E_{22} \\
E_{33} \\
2E_{12} \\
2E_{13} \\
2E_{23}
\end{bmatrix}
\]

The elasticity matrix \( \mathbf{C} \) is described in detail for the implemented models:

#### 15.10.1.1 Material model 1:

linear elastic isotropic, (Input data)

\[
\mathbf{C}^{-1} = \begin{bmatrix}
\frac{1}{E} & -\frac{\nu}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\
-\frac{\nu}{E} & \frac{1}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\
-\frac{\nu}{E} & -\frac{\nu}{E} & \frac{1}{E} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{G} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{G} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{G}
\end{bmatrix}
\]

\[
\mathbf{C} = \begin{bmatrix}
(1-\nu)E^* & \nu E^* & \nu E^* & 0 & 0 & 0 \\
\nu E^* & (1-\nu)E^* & \nu E^* & 0 & 0 & 0 \\
\nu E^* & \nu E^* & (1-\nu)E^* & 0 & 0 & 0 \\
0 & 0 & 0 & G & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & G \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

with \( E^* = \frac{E}{(1+\nu)(1-2\nu)} \), \( G = \frac{E}{2(1+\nu)} \)

Plane stress condition: \( S_{33} = 0 \)

\[
\dot{\mathbf{C}} = \begin{bmatrix}
\dot{E} & \nu \dot{E} & 0 & 0 & 0 \\
\nu \dot{E} & \dot{E} & 0 & 0 & 0 \\
0 & 0 & G & 0 & 0 \\
0 & 0 & 0 & G & 0 \\
0 & 0 & 0 & 0 & G
\end{bmatrix}
\]

\( \dot{E} = \frac{E}{(1-\nu^2)} \), \( G = \frac{E}{2(1+\nu)} \)
15.10.1.2 Material model 2:

linear elastic orthotropic, (Input data)

\[
C^{-1} = \begin{bmatrix}
\frac{1}{E_1} & -\frac{\nu_{12}}{E_1} & -\frac{\nu_{13}}{E_1} & 0 & 0 & 0 \\
-\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & -\frac{\nu_{23}}{E_2} & 0 & 0 & 0 \\
-\frac{\nu_{13}}{E_1} & -\frac{\nu_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{G_{13}} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{G_{23}}
\end{bmatrix}
\]

\[
C = \begin{bmatrix}
(E_2 - \nu_{23}^2 E_3)\frac{E_2}{\Delta} & (\nu_{12} E_2 + \nu_{13} \nu_{23} E_3) E_2 \frac{\Delta}{E_1} & (\nu_{12} \nu_{23} + \nu_{13}) E_1 E_2 E_3 \frac{\Delta}{E_1} & 0 & 0 & 0 \\
(\nu_{12} + \nu_{13} \nu_{23} E_3) E_1 E_2 \frac{\Delta}{E_1} & (E_1 - \nu_{13}^2 E_3)\frac{E_2}{\Delta} & (\nu_{23} E_1 + \nu_{12} \nu_{13} E_2) E_2 E_3 \frac{\Delta}{E_1} & 0 & 0 & 0 \\
(\nu_{12} \nu_{23} + \nu_{13}) E_1 E_2 E_3 \frac{\Delta}{E_1} & (\nu_{23} E_1 + \nu_{12} \nu_{23} E_2) E_2 E_3 \frac{\Delta}{E_1} & (E_1 - \nu_{23}^2 E_2) E_2 E_3 \frac{\Delta}{E_1} & 0 & 0 & 0 \\
0 & 0 & 0 & G_{12} & 0 & 0 \\
0 & 0 & 0 & 0 & G_{13} & 0 \\
0 & 0 & 0 & 0 & 0 & G_{23}
\end{bmatrix}
\]

with \(\Delta = (E_1 E_2 - \nu_{13}^2 E_2 E_3 - \nu_{23}^2 E_1 E_3 - \nu_{12}^2 E_3^2 - 2 \nu_{12} \nu_{13} \nu_{23} E_2 E_3)\)

An associated model with damage is described in material model 15.

15.10.1.3 Material model 3:

linear elastic transversal isotropic, (Input data)

\[
C^{-1} = \begin{bmatrix}
\frac{1}{E_1} & -\frac{\nu_{12}}{E_1} & -\frac{\nu_{13}}{E_1} & 0 & 0 & 0 \\
-\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & -\frac{\nu_{23}}{E_2} & 0 & 0 & 0 \\
-\frac{\nu_{13}}{E_1} & -\frac{\nu_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{G_{13}} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{G_{23}}
\end{bmatrix}
\]

with \(E_2 = E_3, G_{12} = G_{13}\) and \(\nu_{12} = \nu_{13}, \nu_{23} = E_2/(2G_{23}) - 1\)

\[
C = \begin{bmatrix}
E_1(1 - \nu_{23}^2)/\Delta & E_2 \nu_{12}(1 + \nu_{23})/\Delta & E_2 \nu_{12}(1 + \nu_{23})/\Delta & 0 & 0 & 0 \\
E_2 \nu_{12}(1 + \nu_{23})/\Delta & E_2(1 - \nu_{23}^2 E_3^2)/\Delta & E_2(\nu_{23} + \nu_{13}^2 E_3^2)/\Delta & 0 & 0 & 0 \\
E_2 \nu_{12}(1 + \nu_{23})/\Delta & E_2(\nu_{23} + \nu_{13}^2 E_3^2)/\Delta & E_2(1 - \nu_{23}^2 E_3^2)/\Delta & 0 & 0 & 0 \\
0 & 0 & 0 & G_{12} & 0 & 0 \\
0 & 0 & 0 & 0 & G_{12} & 0 \\
0 & 0 & 0 & 0 & 0 & G_{23}
\end{bmatrix}
\]

with \(\Delta = (1 + \nu_{23})(1 - \nu_{23} - 2 \nu_{23}^2 E_2^2/E_1)\)
• **Material model 3a:** linear elastic transversal isotropic and $\sigma_{33} = 0$

Based on material model 3 the elasticity matrix is

$$
C = \begin{bmatrix}
C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\
C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\
C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\
0 & 0 & 0 & G_{12} & 0 & 0 \\
0 & 0 & 0 & 0 & G_{12} & 0 \\
0 & 0 & 0 & 0 & 0 & G_{23}
\end{bmatrix}
$$

Now the assumption $\sigma_{33} = 0$ is added and $\varepsilon_{33}$ is eliminated via $\varepsilon_{33} = -\frac{1}{C_{33}} (C_{13} \varepsilon_{11} - C_{23} C_{22})$. Finally it holds for the elasticity matrix

$$
C = \begin{bmatrix}
\frac{E_1}{\Delta} & \nu_{12} \frac{E_2}{\Delta} & 0 & 0 & 0 & 0 \\
\nu_{12} \frac{E_2}{\Delta} & \frac{E_2}{\Delta} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & G_{12} & 0 & 0 \\
0 & 0 & 0 & 0 & G_{12} & 0 \\
0 & 0 & 0 & 0 & 0 & G_{23}
\end{bmatrix}
$$

with $\Delta = (1 - \nu_{12}^2 \frac{E_2}{E_1})$

### 15.10.1.4 Material model 4:

elastic plastic isotropic small strains $E = E_e + E_p$, *(Input data)*

**References:**

Details on the theoretical and numerical background can be found in v. Mises (1928) [12], Schütt (2005) [22] and Perzyna (1966) [16].

### 15.10.1.5 Material model 5:

elastic plastic isotropic finite strains $F = F_e \cdot F_p$, *(Input data)*

**References:**

Details on the theoretical and numerical background can be found in Simo (1992) [27] and Klinkel (2000) [9].
15.10.1.6 Material model 6:
elastic isotropic finite strains Ogden, (Input data)

Strain energy function:

\[ W_{OG} = \sum_{r=1}^{m} \left( \frac{\mu_r}{\alpha_r} (\lambda_1^{\alpha_r} + \lambda_2^{\alpha_r} + \lambda_3^{\alpha_r} - 3) - \frac{\mu_r}{2} \ln(I_3) \right) + \frac{\Lambda}{4} (I_3 - 1 - \ln(I_3)) \]

\[ I_3 = \lambda_1^2 \lambda_2^2 \lambda_3^2 \]

Eigenvalue problem:

\[ (C - \lambda^2 A)N^A = 0 \quad C = 2E + 1, \quad G = 1 \]

Stress vector:

\[ S_A = \frac{1}{\lambda^2} \sum_{r=1}^{m} (\mu_r(\lambda_A^{\alpha_r} - 1)) + \frac{\Lambda}{2} (I_3 - 1) \]

\[ \hat{S} = (S_1, S_2, S_3)^T \quad \hat{S} = (S^{11}, S^{22}, S^{33}, S^{12}, S^{13}, S^{23})^T \]

\[ t^{AB} = \begin{pmatrix} (N^A \cdot G^1) (N^B \cdot G^1) \\ (N^A \cdot G^2) (N^B \cdot G^2) \\ (N^A \cdot G^3) (N^B \cdot G^3) \\ (N^A \cdot G^1) (N^B \cdot G^3) \\ (N^A \cdot G^2) (N^B \cdot G^2) \end{pmatrix} \]

\[ \mathbf{T}_1 = \begin{pmatrix} t^{11T} \\ t^{22T} \\ t^{33T} \end{pmatrix} \quad \mathbf{T}_2 = \begin{pmatrix} t^{12T} + t^{21T} \\ t^{13T} + t^{31T} \\ t^{23T} + t^{32T} \end{pmatrix} \]

Material matrix:

\[ \mathbf{C}_1 = \begin{pmatrix} C_{11} & \Lambda \frac{I_3}{\lambda_1^2 \lambda_3^2} & \Lambda \frac{I_3}{\lambda_1^2 \lambda_2^2} \\ \Lambda \frac{I_3}{\lambda_2^2 \lambda_1^2} & C_{22} & \Lambda \frac{I_3}{\lambda_2^2 \lambda_3^2} \\ \Lambda \frac{I_3}{\lambda_3^2 \lambda_1^2} & \Lambda \frac{I_3}{\lambda_3^2 \lambda_2^2} & C_{33} \end{pmatrix} \]

\[ C_{11} = \frac{1}{\lambda_1^2} \sum_{r=1}^{m} (\mu_r(\alpha_r - 2) \lambda_1^{\alpha_r} + 2) + \frac{\Lambda}{\lambda_1^2} \]

\[ C_{22} = \frac{1}{\lambda_2^2} \sum_{r=1}^{m} (\mu_r(\alpha_r - 2) \lambda_2^{\alpha_r} + 2) + \frac{\Lambda}{\lambda_2^2} \]

\[ C_{33} = \frac{1}{\lambda_3^2} \sum_{r=1}^{m} (\mu_r(\alpha_r - 2) \lambda_3^{\alpha_r} + 2) + \frac{\Lambda}{\lambda_3^2} \]

\[ \mathbf{C}_2 = \begin{pmatrix} S_1 - S_2 & 0 & 0 \\ 0 & S_1 - S_3 & 0 \\ 0 & 0 & S_2 - S_3 \end{pmatrix} \]

\[ \mathbf{C} = \mathbf{T}_1^T \mathbf{C}_1 \mathbf{T}_1 + \mathbf{T}_2^T \mathbf{C}_2 \mathbf{T}_2 \]

References:

Details on the theoretical and numerical background can be found in Ogden (1972) [14] and Klinkel (2000) [9].

15.10.1.7 Material model 7:

Method of Cells (MOC), Aboudi, (Input data)

References:

Details on the theoretical and numerical background can be found in Gardner (1994) [3] and Aboudi (1991) [1].
15.10.1.8 Material model 8:

FE\textsuperscript{2}, (Input data)

FEAP runs typically with an input-file called e.g. *ifile*

During e.g. *tang,,1* a loop through all elements and within that a loop through all Gauss-Points occurs. On Gauss-point-level the so called micro-problem is executed. For that purpose FEAP is started again (FE\textsuperscript{2}).

Using the INTEL-Compiler the element loop is treated in parallel. The calculation on elements is running on one THREAD separately. Within that the calculation on Gauss Points is then sequential. Here and on the local RVE every form of parallelization is not allowed. The Micro-version of FEAP is compiled sequentially, see the Install-Manual. Furthermore, PARDISO must be used with one processor (*solv,4,,1*).

Using the SALFORD-Compiler FEAP runs sequentially in all loops.

Micro problems are called for *ifile,i* (i=1,...,nproc) identical files. Here i/nproc is the actual/maximum number of threads (parallel) or the actual/maximum number of Gauss-Points (sequential). After one macro step the associated ofiles contain results:

- parallel: ofile,i contains results for last element associated with thread i at last Gauss-point
- sequential: ofile,i contains results for last element at Gauss-point i

Each micro problem has the length values \(l_x, l_y, l_z\). All boundaries are fixed using *ebou*. Further input are the mesh, e.g. with *bloc* and the description of the material. Loads are not defined. Here the macro problem transfers strains \(E\) which lead to prescribed displacements \(V_b = AE\) as 'loads'. These displacements are calculated on the micro problem via the macro *epsq*. For the transfer files *ffile,i* are used. The displacements are defined for the actual loading state. Thus, the *prop,,1* macro has to describe \(F(t) = const\). The actual time and time increment are provided during the strain transfer from the macro problem. After solving the local micro problem \(S\) and \(C\) are calculated with macro *sigq*. These data are send back using files *bfile,i*. This completes the calculation on micro level.
The input data of Ifile \( i \) are

\[
\text{nopr} \quad \text{no output!}
\]

\[
\text{.....}
\]

\[
\text{bloc}
\]

\[
\text{.....}
\]

\[
\text{ebou}
\]

\[
\text{.....}
\]

\[
\text{mate}
\]

\[
\text{solv}
\]

\[
4., 1
\]

\[
\text{end}
\]

\[
\text{batic,tang} \quad \text{calculate one step on RVE from MACRO-Problem}
\]

\[
\text{nopr} \quad \text{no output on screen!}
\]

\[
\text{prop,,1} \quad \text{set load function}
\]

\[
\text{rest,,0}
\]

\[
\text{epsq}
\]

\[
\text{read} \ E, \ \text{calculate} \ V = A \ E, \ \text{set actual time}
\]

\[
\text{loop,,n}
\]

\[
\text{tang,,1}
\]

\[
\text{next}
\]

\[
\text{sigq}
\]

\[
\text{calculate} \ S, \ C \ \text{and send to macro problem}
\]

\[
\text{end,,0}
\]

\[
1, 1, 0, t_{\text{max}}, 1 \quad F(t) = \text{const.}
\]

\[
\text{stop,tang}
\]

\[
\text{batic,updh} \quad \text{update H-Array on RVE within MICRO-Problem}
\]

\[
\text{nopr} \quad \text{no output on screen!}
\]

\[
\text{prop,,1} \quad \text{set load function}
\]

\[
\text{rest,,0}
\]

\[
\text{updh,,2}
\]

\[
\text{update} \ H_2 \rightarrow H_1 \ \text{in case of convergence}
\]

\[
\text{end,,0}
\]

\[
1, 1, 0, t_{\text{max}}, 1 \quad F(t) = \text{const.}
\]

\[
\text{stop,updh}
\]

\[
\text{batic,micr} \quad \text{show results on RVE for one GP of element n}
\]

\[
\text{nopr} \quad \text{no output on screen!}
\]

\[
\text{prop,,1} \quad \text{set load function}
\]

\[
\text{rest,,0}
\]

\[
\text{plot,pers} \quad \text{perspective plot}
\]

\[
\text{plot,hide,1} \quad \text{hidden line plot}
\]

\[
\text{plot,mesh} \quad \text{plot mesh}
\]

\[
\text{plot,axis} \quad \text{plot coordinate system}
\]

\[
\text{end,,0}
\]

\[
1, 1, 0, t_{\text{max}}, 1 \quad F(t) = \text{const.}
\]

\[
10, 10, 5 \quad \text{view point}
\]

\[
0, 0, 1 \quad \text{vertical axis}
\]

\[
\text{inte} \quad \text{switch to interactive mode}
\]

\[
\text{stop,micr}
\]
A procedure step.pcd for one time step on the macro scale should be

```
n1,0,0
```

```
time     set new time
loop,,n   begin iteration for solution
tang,,1   solve problem
next      end iteration at convergence
updh,,1   activate update on local level: done by batch,updh on micro-level
```

One has to prepare or to modify only one input file Ifile_01. Necessary copies of this file, e.g. Ifile_02, Ifile_03, ... are produced using the below defined batch file fbatch. A minimum of nproc=number of threads(parallel) or nproc=number of Gauss points(sequential) is necessary. More copies are allowed and do not influence the calculations. Furthermore all no longer used files in the micro-directory are deleted running this file.

**Example for file fbatch**

```
cd d:\w\feap\exe\rve
copy irve_01 irve_02
copy irve_01 irve_03
copy irve_01 irve_04
copy irve_01 irve_05
copy irve_01 irve_06
copy irve_01 irve_07
copy irve_01 irve_08
.....
```

```
until nproc=no. of threads(parallel) or nproc=no. of Gauss points(sequential)
del frve*
del brve*
del hrve*
del rrve*
del orve*
del *.lnk
```

**Use of Macro fast**

Within a nonlinear approach the same input file is read for each RVE and each iteration step. Thus, it is advantageous to store the input data in a **binary** file via the macro fast once for the file irve_01, which results in a file irve_01.sys. Thus, the new file irve_01 is given by

```
f
```

```
   fast
   irve_01 = name of sys-file (including path)
nopr
end
batch
.....
.....
Macro-Problem

Read Macro-mesh from ifile

LOOP A-Time

Set 'TIME' \( T = T + \Delta T \)

'LOAD'

LOOP,i,N B-Tang

TANG,,1

LOOP C-Elmt

LOOP D-GP

Send \( E \) and \( \Delta T \) to micro-problem \( \rightarrow \) to (TGP)

Start FEAP and Read micro-mesh ifile_iproc

(iproc=actual processor)

(Feap.exe ...

-ifile_iproc -Oofile_iproc -Rrfile_GP.EL -Srfile_GP.EL )

Read History data of GP (\( H_{1L}, H_{2L} \))

(REST,rfile_GP.EL)

PROP,,1 with parameters 1,1,0,tmax,1 (\( F(t) = \text{const}! \))

Read \( E \) and actual macro-'TIME' from (TGP)

EPSQ, Set 'loads' \( V = AE \)

Loop,,M A-Tang

TANG,,1

Next A-Tang

SIGQ

END

(includes save of hist.data of GP to rfile_GP.EL) (\( H_{1L}, H_{2L} \))

(TGP) \( \leftarrow \) Send \( C, S \) to macro-problem

Read \( C, S \) from (TGP)

NEXT D-GP

Calculate \( K \) and \( G = R - \lambda P \)

NEXT C-Elmt

NEXT B-Tang

'UPDATE' History Data of MICRO-Problem (\( H_{2L} \rightarrow H_{1L} \))

NEXT A-Time

END

Institut f"ur Baustatik, KIT Cont. Mesh Macro Plot Mate3D Elem Nege Gmesh Cylt Isogo 565
RVE-Boundary Conditions

- Dimensions of RVE
  \[ \begin{align*}
  &xm \leq x \leq xp \\
  &ym \leq y \leq yp \\
  &zm \leq z \leq zp
  \end{align*} \]
  typically with \( xm=ym=zm=-0.5 \) and \( xp=yp=zp=+0.5 \)

- Displacement boundary conditions
  To do/Theory: Fix nodes on all faces. Then boundary conditions are set via \( v_{bc} = A\varepsilon \).
  FEAP-Input:
  
  \[
  \begin{align*}
  &eboun \\
  &1,xm,1,1,1 \\
  &1,xp,1,1,1 \\
  &2,ym,1,1,1 \\
  &2,yp,1,1,1 \\
  &3,zm,1,1,1 \\
  &3,zp,1,1,1
  \end{align*}
  \]

- Stress boundary conditions
  Boundary conditions are set via \( t_{bc} = \sigma n_{bc} \).
  This type of boundary condition is currently not implemented.

- Periodic boundary conditions
  To do/Theory:
  
  # 1 corner nodes: nodes have to be fixed with standard boundary conditions.
  These are set via \( v_{bc} = A\varepsilon \)
  # 2 all other nodes on edges and outer faces have to be linked.
  The linking condition is \( v_{slave} = v_{master} + A(x_{slave} - x_{master})\varepsilon \)
  Typically one can use \( x_{slave} = xp \) and \( x_{master} = xm \).
  FEAP-Input:
  Step 1: Fix all corner nodes with

  \[
  \begin{align*}
  &poin \\
  &xm,ym,zm \quad 0, \\
  &xp,ym,zm \quad 0, \\
  &xp,yp,zm \quad 0, \\
  &xm,yp,zm \quad 0, \\
  &xm,ym,zp
  \end{align*}
  \]
0,
1,1,1
xp,ym,zp
0,
1,1,1
xp,yp,zp
0,
1,1,1
xm,yp,zp
0,
1,1,1

Step 2a: Set first link conditions on edges. (Note that corner nodes have displacement b.c.s.)
Step 2b: Set then link conditions on all faces

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,1, xm, xp, 0, 0, 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4,2, ym, yp, 0, 0, 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4,3, zm, zp, 0, 0, 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Set both link commands together!

- Periodic boundary conditions for RVEs of shell structures

To do/Theory:

- # 1 Fix center node vertical (to suppress rigid body rotations—other nodes are possible!).
- # 2 Fix all corner nodes horizontal. Boundary conditions are set for fixed nodes via $v_{bc} = A \varepsilon$.
- # 3 Set link conditions on faces vertical with $v_{\text{slave}} = v_{\text{master}} + A(x_{\text{slave}} - x_{\text{master}})\varepsilon$
- # 4 Top and bottom faces are free from boundary conditions.

FEAP-Input:
Step 1: Fix center node vertical (to suppress rigid body rotations—other nodes are possible!)

```plaintext
poin
0,0,0,f
0
0,0,1
0
0
0
```
Step 2: Fix all corner nodes horizontal

```
#eboun
1,xm,1,1,0
1,xp,1,1,0
2,ym,1,1,0
2,yp,1,1,0
```

Step 3: Set link conditions on faces vertical

```
#link
6,1,xm,xp,1,1,0
6,2,ym,yp,1,1,0
```

---

**Results on MICRO-problem (RVE)**

It is possible to check results on the MICRO-problem (RVE) during the calculation. For that purpose the Macro-Problem is running in interactive mode.

Results on the associated MICRO-problem RVE can be checked, if a second Version of FEAP is started. Here, the *input file and the restart file* have to be chosen with respect to the Gauss-point number $i$ and the element number $n$.

Example: `irve_i` and `rrve_i.n`

**Summary**

1. Start MICRO-problem FEAP with input file `Ifile_i` and restart file `RFile_i.n` and the macros within `batc,micr... stop,micr` (see above)
2. View results on element $n$, Gauss-point $i$ using macros
3. Close MICRO-problem with **QUIT**
4. Goto 1 for other element $n$ or other Gauss-point $i$

**References:**

Details on the theoretical and numerical background can be found in Gruttmann, Wagner (2013) [5].
15.10.1.9 Material model 9:
small strain isotropic damage, (Input data)

References:
Details on the theoretical and numerical background can be found in Simo, Ju (1987) [25].

15.10.1.10 Material model 10:
concrete model, Schütt, (Input data)

References:
Details on the theoretical and numerical background can be found in Schütt (2005) [22].

15.10.1.11 Material model 11:
small strain visco-elastic, (Input data)
The material model can be used on the deviatoric part of the strains or on all strain components.
It holds
\[ \sigma = s + mp \]
with Cauchy stress \( \sigma \), stress deviator \( s \), and mean (pressure) stress \( p = \frac{1}{3}m^T \sigma \)
Furthermore it holds
\[ \varepsilon = e + \frac{1}{3}m\theta \]
with strain \( \varepsilon \), strain deviator \( e \), and volume change \( \theta = m^T \varepsilon \), \( m = [1,1,1,0,0,0] \)
The pressure-volume parts are governed by a linear elastic model \( p = K\theta \) with \( K = \frac{E}{3(1-2\nu)} \) the bulk elastic modulus.
The deviatoric parts are assumed to satisfy a linear viscoelastic model.
This could be
\[ s = 2G(\mu_0 e + \sum_{i=1}^{N} \mu_i q_i) \quad \dot{q}_i + \frac{1}{\tau_i}q_i = \dot{\varepsilon} \]
\( \mu_i \) are the viscoelastic shear parameters and \( \tau_i \) are the viscoelastic relaxation times
This form of the representation is equivalent to a a set of Maxwell models in parallel.

References:
Details on the theoretical and numerical background can be found in Simo, Hughes (1988) [26].
15.10.1.12 Material model 12:

linear piezoelectric material model for extended 'stresses' and 'strains', (Input data)

\[
\begin{bmatrix}
  S \\
  -\bar{D}
\end{bmatrix} = 
\begin{bmatrix}
  C & -\phi^T \\
  -\phi & -\epsilon
\end{bmatrix}
\begin{bmatrix}
  E \\
  \bar{E}
\end{bmatrix}
\]

\[
\begin{bmatrix}
  S_{11} & S_{22} & S_{33} & S_{12} & S_{13} & S_{23} \\
  (1 - \nu)E^* & \nu E^* & \nu E^* & 0 & 0 & 0 \\
  \nu E^* & (1 - \nu)E^* & \nu E^* & 0 & 0 & 0 \\
  \nu E^* & \nu E^* & (1 - \nu)E^* & 0 & 0 & 0 \\
  0 & 0 & 0 & G & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
  -\epsilon_{15} & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
  E_{11} \\
  E_{22} \\
  E_{33} \\
  2E_{12} \\
  2E_{13} \\
  2E_{23}
\end{bmatrix}
\]

with \( E^* = \frac{E}{(1 + \nu)(1 - 2\nu)} \), \( G = \frac{E}{2(1 + \nu)} \) and \( C \) as in material model 1.

15.10.1.13 Material model 13:

dielectric elastomers, (Input data)

15.10.1.14 Material model 14:

finite elastic strains, Blatz-Ko, (Input data)

References:
Details on the theoretical and numerical background can be found in Klinkel (2000) [9] and Holzapfel [8].
15.10.1.15 Material model 15:

Transversal isotropic with damage, (Input data)

The general form of an orthotropic material is described for material model 2

With the assumptions $E_2 = E_3$ and $G_{12} = G_{13}$ follows

$$
C = \begin{bmatrix}
(1 - \nu_{23}^2)E_1 E_3 / \Delta & (\nu_{12} + \nu_{13}\nu_{23})E_1 E_3^2 / \Delta & (\nu_{12}\nu_{23} + \nu_{13})E_1 E_3^3 / \Delta & 0 & 0 & 0 \\
(\nu_{12} + \nu_{13}\nu_{23})E_1 E_3 / \Delta & (E_1 - \nu_{23}^2 E_2)E_3 / \Delta & (\nu_{23} + \nu_{12}\nu_{13})E_1 E_3^2 / \Delta & 0 & 0 & 0 \\
(\nu_{12}\nu_{23} + \nu_{13})E_1 E_3 / \Delta & (\nu_{23} + \nu_{12}\nu_{13})E_1 E_3^2 / \Delta & (1 - \nu_{12}^2)E_1 E_3^3 / \Delta & 0 & 0 & 0 \\
0 & 0 & 0 & G_{12} & 0 & 0 \\
0 & 0 & 0 & 0 & G_{12} & 0 \\
0 & 0 & 0 & 0 & 0 & G_{23}
\end{bmatrix}
$$

with $\Delta = (E_1 E_2 - \nu_{12}^2 E_1 E_3 - \nu_{13}^2 E_1 E_3 - \nu_{23}^2 E_1 E_3 - 2\nu_{12}\nu_{13}\nu_{23} E_2^2)$

The further assumptions $\nu_{13} = \nu_{23} = 0$ (!?) lead to:

$$
C = \begin{bmatrix}
E_1 / \Delta & \nu_{12} E_2 / \Delta & 0 & 0 & 0 & 0 \\
\nu_{12} E_2 / \Delta & E_3 / \Delta & 0 & 0 & 0 & 0 \\
0 & 0 & E_3 & 0 & 0 & 0 \\
0 & 0 & 0 & G_{12} & 0 & 0 \\
0 & 0 & 0 & 0 & G_{12} & 0 \\
0 & 0 & 0 & 0 & 0 & G_{23}
\end{bmatrix}
$$

with $\Delta = (1 - \nu_{12}^2 E_2 E_3 / E_1)$

This is similar to material model 3a, except $C_{33} \neq 0$!

Failure model Hashin

\begin{align*}
F^t : & \left( \frac{S_{11}}{R^t} \right)^2 = 1, \quad S_{11} > 0 \\
F^c : & \left( \frac{S_{11}}{R^c} \right)^2 = 1, \quad S_{11} < 0 \\
M^t : & \left( \frac{S_{22}}{R^t} \right)^2 + \left( \frac{S_{12}}{R^t} \right)^2 + \left( \frac{S_{13}}{R^t} \right)^2 + \left( \frac{S_{23}}{R^t} \right)^2 = 1 \\
M^c : & \left( \frac{S_{22}}{2R^c} \right)^2 + \left[ \frac{R^c}{2R^c} \right]^2 - 1 \left( \frac{S_{22}}{R^c} \right)^2 + \left( \frac{S_{12}}{R^c} \right)^2 + \left( \frac{S_{13}}{R^c} \right)^2 + \left( \frac{S_{23}}{R^c} \right)^2 = 1 \\
FMS : & \left( \frac{-S_{11}}{R^c} \right)^2 + \left( \frac{S_{12}}{R^c} \right)^2 + \left( \frac{S_{13}}{R^c} \right)^2 = 1
\end{align*}
Failure model Puck

**FF:**
\[
\varepsilon_{FF} = \left( \frac{\langle S_{11} \rangle}{R_{11}} \right)^2 + \left( \frac{\langle -S_{11} \rangle}{R_{11}} \right)^2 = 1
\]

**IFF A:**
\[
\varepsilon_{IFFA} = \frac{1}{R_{11}} \left( \frac{S_{12}^2}{R_{12}} + \frac{(p_{12}^c S_{22})^2}{R_{12}} \right) = 1, \quad S_{22} > 0
\]

**IFF B:**
\[
\varepsilon_{IFFB} = \frac{1}{R_{11}} \left( \frac{S_{12}^2}{R_{12}} + \frac{(p_{12}^c S_{22})^2}{R_{12}} \right) = 1
\]

**IFF C:**
\[
\varepsilon_{IFFC} = \left( \frac{S_{12}}{2 (1 + p_{11}^c R_{11})} \right)^2 + \left( \frac{S_{22}}{R_{22}} \right)^2 = 1
\]

Failure model Cuntze

**FF1:**
\[
\frac{I_1}{R_{11}} = 1
\]

**FF2:**
\[
\frac{-I_2}{R_{11}} = 1
\]

**IFF1:**
\[
\frac{I_2 + \sqrt{I_4}}{R_{11}} = 1
\]

**IFF2:**
\[
\sqrt{I_3} + b_{11}(I_2 I_3 - I_5) = 1
\]

**IFF3:**
\[
\frac{(b_{11}^2 - 1) I_2 + b_{11}^2 \sqrt{I_4}}{R_{11}} = 1
\]

\[I_1 = S_{11}, \quad I_2 = S_{22}, \quad I_3 = S_{22}^2 + S_{13}^2, \quad I_4 = S_{22}^2 + 4S_{23}^2, \quad I_5 = S_{22}(S_{13}^2 - S_{12}^2) - 4S_{12}S_{13}S_{23}\]

References:
Details on the theoretical and numerical background can be found in Krawiec (2013) [10], for Tsai-Wu criteria in Liu, Tsai [11], for extended Hashin criteria in Goyal et.al. (2004) [4], for Puck criteria in Puck (1996) [17] and Puck, Schürmann (1998) [18], and for Cuntze criteria in Cuntze, Freund 2004 [2].

15.10.1.16 Material model 16:

functionally graded linear elastic, (Input data)

\[E(x, y, z) = E(z) = E = E_0 \cdot (c_0 + c_1 \cdot z + c_2 \cdot z^2)\]

\[
C = \begin{bmatrix}
(1 - \nu)E^* & \nu E^* & \nu E^* & 0 & 0 & 0 \\
\nu E^* & (1 - \nu)E^* & \nu E^* & 0 & 0 & 0 \\
\nu E^* & \nu E^* & (1 - \nu)E^* & 0 & 0 & 0 \\
0 & 0 & 0 & G & 0 & 0 \\
0 & 0 & 0 & 0 & G & 0 \\
0 & 0 & 0 & 0 & 0 & G 
\end{bmatrix}
\]

with \(E^* = \frac{E}{(1 + \nu)(1 - 2\nu)^*}, \quad G = \frac{E}{2(1 + \nu)}\)
Chapter 16

Applications and FAQs

16.1 FAQs

16.1.1 Crash within start procedure

If there occurs a crash of the program within the start procedure, it may happens that the file \texttt{feapname}, which contains the names of input- output- and restart file will be empty. This results in the following error message

![Error message](image)

which then stops the program.

The problem is solved by deleting the file \texttt{feapname}!

16.1.2 Mixing elements

A mixing of elements in a finite element mesh is in general possible within FEAP. The following aspects have to be noted.

Here we will discuss - as an example - the mixing of a solid element (3D) ($\text{nel}_{3D}=8$, $\text{ndf}_{3D}=3$) with a shell element (sh) ($\text{nel}_{sh}=4$, $\text{ndf}_{sh}=6$).

- Within the first input card \texttt{feap} one has to define $\text{ndf}$, which is the maximum number of degrees-of-freedom on any node. In the example it holds $\text{ndf}=\text{ndf}_{\text{max}}=\max(\text{ndf}_{3D}, \text{ndf}_{sh})=6$.
  
  A similar definition is for $\text{nen}$, which is maximum number of nodes on any element. In the example it holds $\text{nen}=\text{nel}_{\text{max}}=\max(\text{nel}_{3D}, \text{nel}_{sh})=8$.

- The boundary conditions have to be set carefully. This means that degrees-of-freedom, where no stiffness is applied by an element have to be fixed. In the example it holds that for any node $i$ which is not connected with a shell element a boundary card like e.g.
boun
i,0,0,0,1,1,1
has to be set.

- Within the element input via the macro `elem` only the necessary nodes are input. This means in the example

  \begin{verbatim}
  elem
  1(3D),mat1,node1,node2,node3,node4,node5,node6,node7,node8
  2(sh),mat2,node1,node2,node3,node4
  \end{verbatim}

- The used elements have to be prepared for the situation of larger numbers of nodes or degrees-of-freedom. For that purpose the storage of data into the element stiffness matrix $K$ and the element load vector or residual vector $P$ have to be correct. $K$ is stored into $s(nst,nst)$ and $P$ is stored into $p(nst)$. Here, $nst$ is defined as $nen \times ndf$. Be aware of the difference $nen=nel_{max}=$ maximum number of nodes on elements and $nel=$ number of nodes on actual element. In the example it holds $nen=8$ and $nel=8$ for volume elements and $nel=4$ for shell elements. Displacements are stored in $ul(ndf,nen)$. Thus matrices $s$, $p$, $ul$ are defined always for the maximal possible situation. Storage of data is then done using the $nel_{local}$ nodes and $ndf_{local}$ degrees of freedom. Jumps are necessary if $nel_{local} < nel_{max}=nen$ or $ndf_{local} < ndf_{max}=ndf$.

- The introduced degrees of freedom must fit together. In the example it holds $u_{3D} = [u_x, u_y, u_z]^{T}$ and $u_{sh} = [u_x, u_y, u_z, \varphi_x, \varphi_y, \varphi_z]^{T}$, which can be used without problems.

If problems occur it is possible to change the position of degrees of freedoms within the `mate`-macro defining an idfg-array with the global dofnumbers for the used element.

\begin{verbatim}
mate
ma,iel,<idfg1,idfg2,.....idfgn>
\end{verbatim}

- The plot macro `disp` should act without any problem.

- The plot macro `stre` should be used carefully. Typically stress$_i$ is plotted for all elements which makes no sense. In the example it holds `stress$_{1(3D)} = \sigma_{xx}$ and `stress$_{1(sh)} = N_x$. For this purpose one can plot only values for each material $i$ via `matn,-1` and then `matn,i`. Note that the element code have to be changed with respect to `matn`.

The title of the stresses in the legend are set to the first element type of `mate` in the Input-file. Thus a simple change is possible.

- The macro `sigq` calculates $C = A^{T}KA - G^{T}K_{a}a^{-1}G$ and $S$ (on micro level). Using the INTEL-Compiler $C$ will only be calculated, if OPEN-MP is off!

### 16.2 Some details using FEAP

#### 16.2.1 Use of macro prop

The loading of a structure could be described using the macro `prop` in combination with the `time` command. Two possibilities exist, which are described in the following within an example.
Use for example a time step of 1 via $dt$, 1. The presented behaviour could be described with load type 1:

$$prop(t) = a(1) + a(2) \cdot (t - t_{min}) + a(3) \cdot (\sin[a(4) \cdot (t - t_{min})]^k).$$

Functions are valid in the range $t_1 \leq t \leq t_2$. In case of any problem at interval limits (e.g. no load or double load) a small 'imperfection' could be helpful, thus change e.g. $t$ to $t+e$ with e.g. $e=1.e-5$.

- Option 1: definition in advance
  
  ```
  prop,3
  1,0 0,10, 0, 1.0
  1,0,e+10,20,10, 0.5
  1,0,e+20,30,15,-1.0
  then
  step,30 (step may be a user defined definition making one step)
  ```

- Option 2: re-definition at the current step of loading
  
  ```
  prop,1
  1,0 0,10, 0, 1.0
  step,10
  prop,1
  1,0,e+10,20,10, 0.5
  step,10
  prop,1
  1,0,e+20,30,15,-1.0
  step,10
  ```
### 16.2.2 Use of macro newf

Problem: Structure with two prescribed displacements and different behavior over time:

![Structure Diagram]

**Input-file:**

```plaintext
... disp 6,0,0,0,0,0,0,0.0 11,0,0,5,0,0,0,0.0 ... Macro-sequence:

dt,,1
prop,,1

**step** → $u_1 = 0.5$ (**step** may be a user defined definition making one step)

**newf** → set $F_0$

**mesh** → switch to mesh-modus and reset prescribed displacements
disp
6,0,0,0,0.15,0.0 11,0,0,0.00,0.0 end → switch to macro-modus

**prop,,1**
1,0,1,4,0,1

**step** → $u_1 = 0.5$  $u_2 = 0.15$

**step** → $u_1 = 0.5$  $u_2 = 0.30$

**step** → $u_1 = 0.5$  $u_2 = 0.45$
Bibliography


