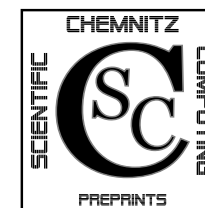


Arnd Meyer

**Programmbeschreibung  
SPC-PM3-AdH-XX  
Teil 2**

CSC/14-02



**Chemnitz Scientific Computing  
Preprints**

**Impressum:**

**Chemnitz Scientific Computing Preprints — ISSN 1864-0087**

(1995–2005: Preprintreihe des Chemnitzer SFB393)

**Herausgeber:**

Professuren für  
Numerische und Angewandte Mathematik  
an der Fakultät für Mathematik  
der Technischen Universität Chemnitz

**Postanschrift:**

TU Chemnitz, Fakultät für Mathematik  
09107 Chemnitz

**Sitz:**

Reichenhainer Str. 41, 09126 Chemnitz

<http://www.tu-chemnitz.de/mathematik/csc/>



**TECHNISCHE UNIVERSITÄT CHEMNITZ**  
**Chemnitz Scientific Computing**  
**Preprints**

Arnd Meyer

**Programmbeschreibung**  
**SPC-PM3-AdH-XX**  
**Teil 2**

CSC/14-02

**Zusammenfassung**

Beschreibung der Finite Elemente Software-Familie

**SPC – PM3 – AdH – XX**

für: **Scientific Parallel Computing -**  
**Programm-Modul 3D – adaptiv – Hexaederelemente.**

Für XX stehen die einzelnen Spezialvarianten, die hier detailliert geschildert werden.

Stand: Ende 2013

## Inhaltsverzeichnis

<b>1 Vorbemerkungen</b>	<b>1</b>
<b>2 Probleme mit transversal-isotropem Material</b>	<b>1</b>
<b>3 Gleichungen vom Sattelpunktstyp</b>	<b>3</b>
<b>4 Probleme der Thermo-Elastizität</b>	<b>5</b>
<b>5 Nichtlineare Probleme der großen Deformationen</b>	<b>7</b>

Some titles in this CSC and the former SFB393 preprint series:

- 11-01 P. Benner, M.-S. Hossain, T. Stykel. Low-rank iterative methods of periodic projected Lyapunov equations and their application in model reduction of periodic descriptor systems. February 2011.
- 11-02 G. Of, G. J. Rodin, O. Steinbach, M. Taus. Coupling Methods for Interior Penalty Discontinuous Galerkin Finite Element Methods and Boundary Element Methods. September 2011.
- 12-01 J. Rückert, A. Meyer. Kirchhoff Plates and Large Deformation. April 2012.
- 12-02 A. Meyer. The Koiter shell equation in a coordinate free description. February 2012.
- 12-03 M. Balg, A. Meyer. Fast simulation of (nearly) incompressible nonlinear elastic material at large strain via adaptive mixed FEM. July 2012.
- 13-01 A. Meyer. The Koiter shell equation in a coordinate free description – extended. September 2013.
- 13-02 R. Schneider. With a new refinement paradigm towards anisotropic adaptive FEM on triangular meshes. September 2013.
- 13-03 A. Meyer. The linear Naghdi shell equation in a coordinate free description. November 2013.

The complete list of CSC and SFB393 preprints is available via  
<http://www.tu-chemnitz.de/mathematik/csc/>

Author's addresses:

Arnd Meyer  
TU Chemnitz, Fakultät für Mathematik  
09107 Chemnitz, Germany

unter Mitwirkung von:  
Janine Glänzel, Dr. Martina Weise, Dr. Roman Unger, Dr. Matthias Pester,  
Dr. Michael Weise

<http://www.tu-chemnitz.de/mathematik/>

- [15] R. Unger, *Obstacle Description with Radial Basis Functions for Contact Problems in Elasticity*, Preprint CSC09-01 TUChemnitz (2009).
- [16] J. Glänzel, *Kurzvorstellung der 3D-FEM Software SPC-PM3AdH-XX*, Preprint CSC09-03 TUChemnitz (2009).
- [17] M. Balg, A. Meyer *Numerische Simulation nahezu inkompressibler Materialien unter Verwendung von adaptiver, gemischter FEM*, Preprint CSC10-02 TUChemnitz (2010).
- [18] M. Weise, *Elastic Incompressibility and Large Deformations*, Dissertation, TUChemnitz, (2014).
- [19] M. Weise, A. Meyer, *Grundgleichungen für transversal isotropes Materialverhalten*, Preprint CSC10-03 TUChemnitz (2010).
- [20] M. Balg, A. Meyer, *Fast simulation of (nearly) incompressible nonlinear elastic material at large strain via adaptive mixed FEM*, Preprint CSC12-03 TUChemnitz (2012).
- [21] M. Balg, J. Lang, A. Meyer, G. Rünger, *Array-based reduction operations for a parallel adaptive FEM*, R.Keller et al(Eds.): Facing the Multicore-Challenge III 2012 Lecture Notes in Comp.Sci. Vol.7686, pp.25-36,(2013).

## 1 Vorbemerkungen

Der Teil 1 der Programmbeschreibung zu **SPC\_PM3AdH-XX** behandelte die für alle Varianten gültigen grundlegenden Datenstrukturen und die Organisation zur effektiven Nutzung der adaptiven FEM. Hierzu wurde die Basisimplementierung **A3D\_Original** erläutert, die der Lösung einfacher Reaktions-Diffusionsgleichungen oder linearer Elastizitätsgleichungen mit isotropem Material vorbehalten ist.

Im vorliegenden Teil 2 werden die daraus abgeleiteten Programm-Module zur Lösung komplizierterer Gleichungen erläutert. Hierbei handelt es sich vorrangig um die Lösung von mechanischen Deformationsproblemen mit allgemeinerem Materialverhalten oder Temperaturabhängigkeit. Alle diese Programm-Module sind aus kleinen Änderungen der Grundvariante **A3D\_Original** hervorgegangen.

Die erste Verallgemeinerung betrifft das Materialverhalten der elastischen Deformationen. Der Übergang zu transversal-isotropem Material erfordert im wesentlichen die Benutzung eines Vektorfeldes der „Faserorientierung“ (**A3D-TraIso**).

Zum zweiten wird die Kopplung mit einem weiteren skalaren Feld vorgesehen. Dies ist der hydrostatische Druck bei **A3D-inkLE** (linear elastisches Material mit fast-Inkompressibilität als gemischte FEM) bzw. die Temperatur bei den Varianten der Thermoelastizität (**A3D-ThEl** für isotropes Material, **A3D-TEAni** für die Kopplung von Thermoelastizität mit transversal isotropem Material).

Zum dritten werden Nichtlinearitäten im Materialverhalten (und geometrische Nichtlinearität der großen Verzerrungen) betrachtet in **A3D-LD** (**LD** für „large deformations“) und **A3D-inkLD** (mit zusätzlicher Inkompressibilität).

## 2 Probleme mit transversal-isotropem Material

SPC-PM3-AdH-TraIso      (in: **A3D-TraIso**)

Dieser Programmmodul ist die Erweiterung der linear elastischen Deformationsberechnung eines 3-dimensionalen Körpers unter äußeren Kräften aus dem Basismodul **A3D\_Original** auf ein nicht mehr isotropes Materialgesetz.

Besonderes Kennzeichen ist die notwendige Bereitstellung eines Vektorfeldes  $\mathbf{a}(x)$ , das die Vorzugsrichtung („Faserorientierung“) am Punkt  $x$  des Gebietes definiert. Die zugrundeliegende Differentialgleichung ist die gleiche wie im isotropen Fall:

finde  $\mathbf{u} \in \mathbb{H}^1(\Omega)^3$  (mit ev. vorgegebenen Werten auf Randstücken  $\Gamma_D \subset \partial\Omega$ ), so dass

$$a(\mathbf{u}, \mathbf{v}) = l(\mathbf{v}) \quad \forall \mathbf{v} \in \mathbb{H}_D^1(\Omega)^3$$

( $\mathbf{v}$  hat vorgegebene Werte 0 auf  $\Gamma_D$ ),

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{v}) \, d\Omega$$

mit dem linearen Verzerrungstensor

$$\boldsymbol{\epsilon}(\mathbf{v}) = ((\nabla \mathbf{v}) + (\nabla \mathbf{v})^T)/2.$$

Der Spannungstensor ist linear vom Verzerrungstensor  $\boldsymbol{\epsilon}(\mathbf{u})$  abhängig als

$$\boldsymbol{\sigma}(\mathbf{u}) = \mathfrak{C} : \boldsymbol{\epsilon}(\mathbf{u})$$

und der Materialtensor 4. Stufe  $\mathfrak{C}$  hat über  $\mathbf{a}(x)$  eine Ortsabhängigkeit wie in [19] ausführlich dargelegt ist:

$$\mathfrak{C} = 2\mu \mathfrak{I} + \lambda (\mathbb{1}) + \alpha (\mathbf{a}\mathbf{a} + \mathbf{l}\mathbf{a}\mathbf{a}) + 2(\mu_a - \mu) \hat{\mathfrak{C}} + \beta \mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}$$

Dabei ist  $\hat{\mathfrak{C}} : \mathbf{E} = \mathbf{a}\mathbf{a} \cdot \mathbf{E} + \mathbf{E} \cdot \mathbf{a}\mathbf{a} \, \forall \mathbf{E}$  und im weiteren bezeichnet stets  $\mathbb{1}$  die identische Abbildung als Tensor 2. Stufe,  $\mathbb{1} \cdot \mathbf{u} = \mathbf{u} \, \forall \mathbf{u}$  und  $\mathfrak{I}$  die identische Abbildung als Tensor 4. Stufe,  $\mathfrak{I} : \mathbf{E} = \mathbf{E} \, \forall \mathbf{E}$ .

Hierbei sind die 5 Materialkonstanten

$$\mu, \mu_a, \lambda, \alpha, \beta$$

zu benutzen, die für jeden Materialbereich konstant sind. Sie werden aus 5 Ingenieurskonstanten erhalten (UP: *getmaterial*), die im Eingabefile *xxx.std* gesetzt werden. (vgl. [19] Formel (17)). Die Implementierung (siehe z. B. File *element.f*) benutzt die Voigtische Notation einer (6x6)-Matrix anstelle des Tensors  $\mathfrak{C}$  (vgl. Teil 1 Abschnitt 6.3).

Folgerichtig sind diejenigen Module gegenüber **A3D-Original** geändert, die die Berechnung des Spannungstensors benötigen (also wesentlich: *element.f*, *GetSigma.f*, *GesEstErr.f* u.ä.).

Die Orstabhängigkeit der Vektorfunktion  $\mathbf{a}(x)$  erzeugt eine technische Schwierigkeit: Nicht alle Problemata können (elementorientiert) im Eingabefile gespeichert sein, sondern ein besonderer Funktionsruf der Abb.  $x \rightarrow \mathbf{a}(x)$  ist als Zusatz-Unterprogramm bereitzustellen.

Hierzu wurde *FibreInfo* (in *fibrefinfo.f*) neu konzipiert, was (extrem beispielabhängig!) vom Nutzer zu spezifizieren ist. Eine einfachere Version entsteht, wenn  $\mathbf{a}(x)$  in Teilgebieten (Materialbereichen) konstant gewählt wird. Dann sind wieder alle Materialdaten im Eingabefile festgelegt.

## Literatur

- [1] J. Bramble, J. Pasciak, and J. Xu, *Parallel multilevel preconditioners*, Math. Comput., 55:1-22, (1990).
- [2] S. Zhang. *Multilevel schwarz methods* Numer. Math. 63:512-539,(1992).
- [3] A. Meyer, *Projected PCGM for Handling Hanging Nodes in Adaptive Finite Element Procedures*, Preprint SFB393 99-25 TUChemnitz (1999).
- [4] A. Meyer, P.Steinhorst, *Betrachtung zur Spektraläquivalenz für das Schur-Komplement im Bramble-Pasciak-CG bei piezoelektrischen Problemen*, Preprint CSC07-08 TUChemnitz (2008).
- [5] A. Meyer, P.Steinhorst, *Überlegungen zur Parameterwahl im Bramble-Pasciak-CG für gemischte FEM*, Preprint SFB393 05-07 TUChemnitz (2005).
- [6] P. Steinhorst, *Anwendung adaptiver FEM für piezoelektrische und spezielle mechanische Probleme*, Dissertation, TU Chemnitz, 2009
- [7] S. Beuchler, A. Meyer, M. Pester, *SPC-PM3AdH v 1.0 - Programmer's Manual*, Preprint SFB393 01-18 TUChemnitz (2001), (revised version, August 2003).
- [8] A. Meyer, *Projection Techniques embedded in the PCGM for Handling Hanging Nodes and Boundary Restrictions* in: Engineering Computational Technology B.H.V.Topping and Z.Bittnar, (Eds.), Saxe-Coburg Publ., Stirling, Scotland, 147-165, (2002).
- [9] M. Pester, *Bibliotheken zur Entwicklung paralleler Algorithmen - Basisroutinen für Kommunikation und Grafik*, Preprint SFB393 02-01 TUChemnitz (2002).
- [10] M. Pester, *Visualization Tools for 2D and 3D Finite Element Programs - Users Manual*, Preprint SFB393 02-02 TUChemnitz (2002).
- [11] A. Meyer, R. Unger, *Projection methods for contact problems in elasticity*, Preprint SFB393 04-04 TUChemnitz (2004).
- [12] R. Unger, *Unterraum-CG-Techniken zur Bearbeitung von Kontaktproblemen*, Dissertation TUChemnitz (2006).
- [13] A. Meyer, *Grundgleichungen und adaptive Finite-Elemente-Simulation bei „Großen Deformationen“*, Preprint CSC07-02 TUChemnitz (2007).
- [14] A. Meyer, *Error Estimators and the Adaptive Finite Element Method on Large Strain Deformation Problems*, Math.Meth.Appl.Sci.32:2148-2159,(2009)

Newton-System ist dann im Diskreten das lineare Gleichungssystem

$$A(\mathbf{u}) \delta \mathbf{u} = \mathbf{r}(\mathbf{u})$$

mit dem Residuum  $\mathbf{r}(\mathbf{u})$  (analog zu  $tl(\mathbf{v}) - a(\mathbf{u}; \mathbf{v}) \forall \mathbf{v}$ ) auf der rechten Seite. Abermals geschieht die Lösung mit BPX-vorkonditioniertem CG und die Matrixmultiplikation mit  $A(\mathbf{u})$  als Summe der Elementanteile.

Die aktuelle Näherung der Knotenwerte von  $\mathbf{u}$  wird stets auf der Datenstruktur  $\mathbf{X}(\text{MNode}+1 \dots 3, j) \forall j$  gehalten und also nach der Newton-Korrektur  $\mathbf{u} := \mathbf{u} + \delta \mathbf{u}$  rückgespeichert (im Hauptprogramm nach dem Aufruf von *DoNewtonCor*).

Der Programmmodul

**SPC-PM3-AdH-inkLD** (in: **A3D-inkLD**)

ist gewissermaßen die Vereinigung von **SPC-PM3-AdH-inkLE** mit **SPC-PM3-AdH-LD**. Bei den nichtlinearen Materialgesetzen wird sich die (fast-)Inkompressibilität des Materials ebenso wie im linear elastischen durch einen sehr großen Materialparameter (z.B.  $D_2$  in der obigen Materialfunktion) zeigen. Deshalb ist es üblich die Materialfunktion in die Summe von 2 Anteilen

$$\psi(\mathbf{C}) = \psi_D(\mathbf{C}_D) + \kappa(\psi_V(\iota))^2$$

aufzuspalten, wobei

$$\mathbf{C}_D = \mathbf{C} \iota^{-(1/3)}, \quad \iota = \det \mathbf{C}$$

der „deviatorische“ Anteil des rechten Cauchy-Green-Tensors ist. Bei zunehmender Inkompressibilität gilt nun  $\kappa \rightarrow \infty$ , also  $\psi_V(\iota) \rightarrow 0$  (was gleichbedeutend mit  $\det \mathbf{C} \rightarrow 1$  ist). Deshalb ist die zu Abschnitt 4 analoge Substitution hier

$$p = \kappa \psi_V(\iota).$$

Für die beiden Teile der Materialfunktion wird hier vorläufig

$$\psi_D(\mathbf{C}) = c_{10}(a_1 \iota^{-(1/3)} - 3), \quad \psi_V(\iota) = \ln(\iota)$$

benutzt ( $\iota = \det \mathbf{C}$  wie oben), aber weitere Verallgemeinerungen sind wieder leicht durch Änderung des UP *MaterialFunction* zu erhalten. Die genauen Zusammenhänge, die resultierende Pseudo-Sattelpunktsgleichung und die Newton-Linearisierung zu einer Folge von echten Sattelpunktsgleichungen sind in der Dissertation von Martina Weise [18] ausführlich beschrieben.

Zur Finite Element Diskretisierung wird wieder das Taylor-Hood-Element bevorzugt benutzt. Hier gelten analoge Angaben wie in Abschnitt 4.

### 3 Gleichungen vom Sattelpunktstyp

**SPC-PM3-AdH-inkLE** (in: **A3D-inkLE**)

Bei linear elastischen Deformationsproblemen tritt im isotropen Fall das Materialgesetz

$$\boldsymbol{\sigma} = 2\mu \boldsymbol{\epsilon} + \lambda(\text{tr } \boldsymbol{\epsilon})\mathbf{I}$$

auf, wobei die beiden Lamé-Konstanten  $\mu$  und  $\lambda$  aus  $E$  (E-Modul) und  $\nu$  (Querkontraktionszahl) berechnet werden als

$$2\mu = \frac{E}{1 + \nu}, \quad \lambda = (2\mu) \frac{\nu}{1 - 2\nu}.$$

Fast inkompressible Materialien haben  $\nu$  nahe an 0.5, weshalb  $\lambda \rightarrow \infty$  zu Fehlern in der FE-Approximation führt („volume locking“). Abhilfe schafft die Einführung des hydrostatischen Druckes  $p = \lambda \text{tr } \boldsymbol{\epsilon} = \lambda \text{div } \mathbf{u}$  und der Übergang zur Sattelpunktsformulierung:

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) + b(p, \mathbf{v}) &= l(\mathbf{v}) & \forall \mathbf{v} \in \mathbb{H}_D^1(\Omega)^3 \\ b(q, \mathbf{u}) - c(p, q) &= 0 & \forall q \in L_2(\Omega), \end{aligned}$$

wobei jetzt

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} 2\mu \boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{v}) \, d\Omega, \\ b(p, \mathbf{u}) &= \int_{\Omega} p \text{div } \mathbf{u} \, d\Omega, \\ c(p, q) &= \int_{\Omega} \frac{1}{\lambda} p q \, d\Omega \end{aligned}$$

sind. Die FE-Diskretisierung muss in diesem Fall die LBB-Bedingungen erfüllen, so dass die Benutzung von 2 verschiedenen FE-Ansätzen für  $\mathbf{u}$  und  $p$  notwendig ist. Üblich ist die Benutzung der Taylor-Hood-Elemente (27-Knoten-triquadratische Ansätze für  $\mathbf{u}$  und 8-Knoten-trilineare Ansätze für  $p$ ). Es ist aber auch triquadratisch+elementkonstant denkbar. Deshalb wurde das kleine Unterprogramm *ElemDefine* eingeführt, das die 3 INTEGER-Werte **NeX**, **NeU**, **NeP** liefert. Hierbei ist **NeX** für die gewählte Geometrie-Abbildung

„Masterelement“  $[-1, 1]^3 \longleftrightarrow$  „Wertelement“ verantwortlich:

$$x(\hat{x}) = \sum_{i=1}^{\text{NeX}} N_i(\hat{x}) x^{(i)}$$

mit  $\hat{x} \in [-1, 1]^3$  und  $x^{(i)}$  den **NeX** Knoten des „Wellelements“. Für **NeX** ist 8 oder 27 sinnvoll (dann sind  $N_i(\hat{x})$  die trilinearen bzw. triquadratischen Formfunktionen). **NeU** definiert die  $\mathbf{u}$ -Ansatzfunktionen im Element, vorrangig ist **NeU=27·3** für die triquadratischen Ansatzfunktionen. **NeP** definiert die skalaren Ansatzfunktionen für das Feld  $p$  und darf **NeP=8** (Taylor-Hood-Elemente) oder **NeP=1** (elementkonstanter Ansatz) sein.

Die Sattelpunktsstruktur zieht die Berechnung von 3 Elementmatrizen  $A_{el}, B_{el}, C_{el}$  nach sich, die theoretisch zu den jeweiligen Gesamtsteifigkeitsmatrizen assembliert würden und das Gleichungssystem

$$\begin{pmatrix} A & B \\ B^T & -C \end{pmatrix} \begin{pmatrix} \underline{u} \\ \underline{p} \end{pmatrix} = \begin{pmatrix} \underline{b} \\ 0 \end{pmatrix}$$

für die Vektoren  $\underline{u} \in \mathbb{R}^{3n}$ ,  $\underline{p} \in \mathbb{R}^{np}$  der entsprechenden Knotenwerte liefern. Abermals wird  $A, B, C$  nicht assembliert, sondern die Matrix×Vektor-Multiplikationen im Löser summieren über alle Element-Anteile.

Die beiden symmetrischen Matrizen  $A_{el}, C_{el}$  sind als oberes Dreieck in die Datenstruktur **Solid(JELMatA, .)** bzw. **Solid(JELMatC, .)** eingepackt, die Rechteckmatrix  $B(\text{NeU} \times \text{NeP})$  als volle Matrix auf **Solid(JELMatB, .)** (vgl. UP *assem.f*).

Zum schnellen Lösen des obigen Gleichungssystems kommt der Bramble-Pasciak-CG zur Anwendung. Grundlegende Informationen hierzu, sowie einige Verbesserungen zur Effizienzsteigerung finden sich in [6], [18].

#### SPC-PM3-AdH-Piezo (in: A3D-Piezo)

Bei der Betrachtung von piezo-elektrischen Materialien entsteht ein gleichartiges Gleichungssystem vom Sattelpunktstyp. Hier ist  $p$  das skalare elektrische Potential und

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} \boldsymbol{\epsilon}(\mathbf{v}) : \mathfrak{C} : \boldsymbol{\epsilon}(\mathbf{u}) \, d\Omega \\ b(p, \mathbf{u}) &= \int_{\Omega} \boldsymbol{\epsilon}(\mathbf{u}) : \mathfrak{E} \cdot \nabla p \, d\Omega \\ c(p, q) &= \int_{\Omega} \nabla q \cdot \boldsymbol{\kappa} \cdot \nabla p \, d\Omega \end{aligned}$$

mit dem Elastizitätstensor  $\mathfrak{C}$  (bisher nur isotrop:  $\mathfrak{C} = 2\mu\mathfrak{J} + \lambda(\mathbb{I})$ ) benutzt und einem Tensor 3. Stufe  $\mathfrak{E}$  der Kopplung des Gradienten von  $p$  zur mechanischen Deformation. Genaueres zu den definierenden Eingangsgrößen ist in [6] ausgeführt. Durch obige Definition ist die Matrix  $C$  stets gut positiv definit, weshalb die

Damit muss folgende Iterationsvorschrift ausgeführt werden:

Löse

$$(1) \quad a'(\mathbf{u}; \delta\mathbf{u}, \mathbf{v}) = t l(\mathbf{v}) - a(\mathbf{u}; \mathbf{v}) \quad \forall \mathbf{v}$$

und

$$\mathbf{u} := \mathbf{u} + \delta\mathbf{u}.$$

Wenn

$$(*) \quad \frac{\|\delta\mathbf{u}\|}{\|\mathbf{u}\|} < \epsilon_{\text{Newton}}, \quad \text{dann } t := t + \Delta t$$

solange bis  $t = 1$  erreicht ist, sonst weiter bei (1).

Die Steuerung der Newton-Schritte übernimmt die Routine *DoNewtonCor* (Datei *aNetzQ/donewtoncor.f*) sowie die Nutzer-Abfrage im Hauptprogramm.

Übliche Vorgehensweise ist folgende:

Mehrfaches Lösen der Newton-Gleichung mit genügend kleinem festen  $\Delta t$ . Dazu wenige Schritte (ev. totale) Netzverfeinerung. Dann wird durch Einschalten der Inkrementierung (Steuerbyte '1') der Test (\*) aktiviert und  $t$  nach je 2...4 Newton-Schritten auf festgehaltenem Netz vergrößert. Dies kann automatisch durch das Steuerbyte 'A' erfolgen bis zum Erreichen von  $t = 1$ , wo durch Eingabe des Steuerbytes '1' die Inkrementierung wieder ausgeschaltet wird und nun wieder unter Nutzersteuerung die Newton-Konvergenz (mehrmals Byte '0') abgewartet werden sollte. Dann kann weitere Netzverfeinerung (mit konstantem  $t = 1$ ) vorgenommen werden.

Die Finite Elemente Diskretisierung dieses Prozesses erfordert zunächst die Berechnung von  $a'(\mathbf{u}; \delta\mathbf{u}, \mathbf{v})$ , die aufgrund der zweifachen Abhängigkeit der  $a(\cdot; \cdot)$ -Form von  $\mathbf{u}$  sich als

$$a'(\mathbf{u}; \delta\mathbf{u}, \mathbf{v}) = \int_{\Omega_0} \mathbf{E}(\mathbf{u}; \mathbf{v}) : \mathfrak{C} : \mathbf{E}(\mathbf{u}; \delta\mathbf{u}) \, d\Omega_0 + \int_{\Omega_0} \text{Grad}\mathbf{v}^T : (\mathbb{T} \cdot \text{Grad}\delta\mathbf{u}) \, d\Omega_0$$

ergibt. Der "tangente" Materialtensor  $\mathfrak{C}$  entsteht durch die 2. Ableitung von  $\psi(\cdot)$  nach  $\mathbf{C}$  als

$$\mathfrak{C} = 4 \left( \sum_{k,j=1}^3 \frac{\partial^2 \psi}{\partial a_k \partial a_j} [\mathbf{C}^{k-1} \mathbf{C}^{j-1}] + \frac{\partial \psi}{\partial a_2} \mathfrak{J} + \frac{\partial \psi}{\partial a_3} \hat{\mathbf{C}} \right)$$

(hier ist  $\hat{\mathbf{C}} : \mathbf{E} = \mathbf{C} \cdot \mathbf{E} + \mathbf{E} \cdot \mathbf{C} \quad \forall \mathbf{E}$ ). Deshalb werden im UP *MaterialFunction* die 2. Ableitungen von  $\psi$  bereitgestellt. Der Aufbau der  $(6 \times 6)$ -Materialmatrix (Voigtsche Notation) aus diesen Ableitungen erfolgt im UP *MaterialTensor*.

Indem für  $\delta\mathbf{u}$  und  $\mathbf{v}$  alle FE-Formfunktionen des Elements eingesetzt werden, entsteht hieraus die Elementsteifigkeitsmatrix  $A_{el}(\mathbf{u})$  wie in [13] beschrieben. Das



die Richtungsableitung des Verzerrungstensors  $\mathbf{E}(\mathbf{u})$  (der großen Deformation) in Richtung  $\mathbf{v}$  :

$$2\mathbf{E}(\mathbf{u}; \mathbf{v}) = \text{Grad } \mathbf{v} + \text{Grad } \mathbf{v}^T + \text{Grad } \mathbf{u} \cdot \text{Grad } \mathbf{v}^T + \text{Grad } \mathbf{v} \cdot \text{Grad } \mathbf{u}^T.$$

Das Materialgesetz manifestiert sich in der nichtlinearen Abhängigkeit des 2. Piola-Kirchhoffschen Spannungstensors  $\mathbf{T}$  vom rechten Cauchy-Green-Tensor  $\mathbf{C}$

$$\mathbf{C} = (\mathbf{I} + \text{Grad } \mathbf{u}) \cdot (\mathbf{I} + \text{Grad } \mathbf{u}^T) = \mathbf{I} + 2\mathbf{E}(\mathbf{u}).$$

Diese Abhängigkeit wird durch die Wahl eines Energiefunktionalis

$$W(\mathbf{C}) = \int_{\Omega_0} \psi(a_1, a_2, a_3) d\Omega_0$$

festgelegt. Die Materialfunktion  $\psi$  (Energiedichte) hängt von den Invarianten des rechten Cauchy-Green-Tensors ab, die wiederum einfache Funktionen der 3 Pseudoinvarianten  $a_k = \frac{1}{k} \text{tr } \mathbf{C}^k$  sind. Deshalb ergibt sich

$$\mathbf{T} = 2 \frac{\partial \psi}{\partial \mathbf{C}} = 2 \sum_{k=1}^3 \frac{\partial \psi}{\partial a_k} \mathbf{C}^{k-1}$$

als leicht angebar, indem eine Routine (*MaterialFunction*) aus den 3 Werten  $a_k$  alle ersten und zweiten Ableitungen von  $\psi$  generiert. Diese Routine muss bei Änderungen des Materialgesetzes ausgetauscht werden, hat aber den Vorteil, dass nur die skalaren Abbildungen

$$(a_1, a_2, a_3) \mapsto \left( \frac{\partial \psi}{\partial a_k}, \frac{\partial^2 \psi}{\partial a_k \partial a_j} \right)$$

zu implementieren sind. Wegen der Struktur von  $\mathbf{T}$  sind keine weiteren Änderungen im Programm bei Änderung des Materialgesetzes nötig. Beispielsweise ist die Funktion  $\psi$  beim gern benutzten Neo-Hooke-Material für gummi-ähnliche Materialien einfach

$$\psi(a_1, a_2, a_3) = c_{10} (a_1 - \ln(\iota) - 3) + D_2 \ln(\iota)^2, \quad \iota = \det \mathbf{C} = (a_1^3/6 - a_1 a_2 + a_3).$$

Die Lösung des nichtlinearen Systems

$$a(\mathbf{u}; \mathbf{v}) = l(\mathbf{v}) \quad \forall \mathbf{v}$$

erfolgt mit Hilfe eines inkrementellen Newton-Verfahrens. Mit  $t := \Delta t$  (nach Eingabe  $\Delta t = 1/M$ ) wird eine Folge von Newton-Iterationen generiert.

Hierzu sei  $a'(\mathbf{u}; \delta \mathbf{u}, \mathbf{v})$  die erste Ableitung von  $a(\mathbf{u}; \mathbf{v})$  nach  $\mathbf{u}$

$$(\text{Newton-Linearisierung: } a(\mathbf{u} + \delta \mathbf{u}; \mathbf{v}) = a(\mathbf{u}; \mathbf{v}) + a'(\mathbf{u}; \delta \mathbf{u}, \mathbf{v}) + \mathcal{O}(\|\delta \mathbf{u}\|^2)).$$

LBB-Bedingung entfällt und gleichgradige Polynome für  $\mathbf{u}$  und  $p$  erlaubt sind. Deshalb kann *ElemDefine* im Programmmodul **SPC-PM3-AdH-Piezo** auch einfach  $\text{NeP}=\text{NeX}=\text{NeIN}$  u.  $\text{NeU}=3 \cdot \text{NeX}$  setzen.

## 4 Probleme der Thermo-Elastizität

Dieser Programmmodul dient zur Simulation von linearen thermoelastischen Problemen mit stationärem Temperaturfeld. In

$$\text{SPC-PM3-AdH-ThEl} \quad (\text{in: } \mathbf{A3D-ThEl})$$

wird die einfache Isotropie im Temperaturverhalten als auch in der mechanischen Deformation benutzt. Der Programmmodul

$$\text{SPC-PM3-AdH-TEAni} \quad (\text{in: } \mathbf{A3D-TEAni})$$

dient der anisotropen Erweiterung. Das skalare Feld  $\vartheta(x)$  steht für die Temperaturverteilung im Festkörper  $\Omega$ . Dann ist die Verformung durch Temperaturexpansion zusätzlich zu äußeren Kräften aus der Elastizitätsgleichung berechenbar, in die jetzt ein temperaturabhängiger Term der Dehnungs-Spannungs-Abbildung eingeht:

$$\boldsymbol{\sigma}(\mathbf{u}, \vartheta) = \boldsymbol{\mathfrak{C}} : (\boldsymbol{\epsilon}(\mathbf{u}) - (\vartheta - \vartheta_0)\mathbf{A}).$$

Im Allgemeinfall steht  $\boldsymbol{\mathfrak{C}}$  (Tensor 4. Stufe) für das Materialgesetz und der Tensor 2. Stufe  $\mathbf{A}$  für die ev. anisotrope Temperaturexpansion. Im isotropen Fall wird wieder

$$\boldsymbol{\mathfrak{C}} = 2\mu \mathcal{J} + \lambda(\mathbf{I}\mathbf{I}) \quad \text{und} \quad \mathbf{A} = \alpha \mathbf{I}$$

benutzt mit dem Temperaturexpansionskoeffizient  $\alpha$  (Maßeinheit: 1/K). Also ergibt sich folgender 2-stufiger Prozeß:

1. Berechnung einer Temperaturverteilung  $\vartheta(x)$  aus:

$$c(\vartheta, v) = l_1(v) \quad \forall v \in \mathbb{H}_D^1(\Omega)$$

( $\vartheta$  hat ev. Dirichlet-Randbedingungen auf Teilen des Randes  $\Gamma_{D,1}$  und  $\partial\Omega = \Gamma_{D,1} \cup \Gamma_{N,1}$ )

$$c(\vartheta, v) = \int_{\Omega} \kappa \nabla \vartheta \cdot \nabla v d\Omega$$

$$l_1(v) = \int_{\Omega} q(x) v(x) d\Omega + \int_{\Gamma_{N,1}} g(x) v(x) dS$$

2. Mit dem Feld  $\vartheta$  wird das Deformationsproblem gelöst:

$$a(\mathbf{u}, \mathbf{v}) = b(\vartheta, \mathbf{v}) + l_2(\mathbf{v}) \quad \forall \mathbf{v} \in (\mathbb{H}_D^1(\Omega))^3$$

( $\mathbf{u}$  hat ev. Dirichlet-Randbedingungen auf Teilen des Randes  $\Gamma_{D,2}$  und  $\partial\Omega = \Gamma_{D,2} \cup \Gamma_{N,2}$ )

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \boldsymbol{\epsilon}(\mathbf{v}) : \mathbf{C} : \boldsymbol{\epsilon}(\mathbf{u}) d\Omega$$

$$b(\vartheta, \mathbf{v}) = \int_{\Omega} \vartheta \boldsymbol{\epsilon}(\mathbf{v}) : \mathbf{C} : \mathbf{A} d\Omega$$

$$l_2(\mathbf{v}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\Omega + \int_{\Gamma_{N,2}} \mathbf{g} \cdot \mathbf{v} dS - \int_{el} \vartheta_0 \boldsymbol{\epsilon}(\mathbf{v}) : \mathbf{C} : \mathbf{A} d\Omega$$

Im isotropen Fall vereinfacht sich  $b(\vartheta, \mathbf{v})$  zu

$$\int_{\Omega} \vartheta \alpha (2\mu + 3\lambda) \operatorname{div} \mathbf{v} d\Omega.$$

Die FEM-Diskretisierung beider Gleichungen benutzt für  $\mathbf{u}$  und  $\vartheta$  die gleichen Formfunktionen pro Element (also  $\mathbf{NeIN}=8, 20$  oder  $27$ ) und im UP *ElemDefine* wird wieder  $\mathbf{NeX} = \mathbf{NeP} = \mathbf{NeIN}$  und  $\mathbf{NeU} = 3 * \mathbf{NeIN}$  gesetzt (isoparametrische Elemente). Somit werden die 3 Elementmatrizen  $A_{el}, B_{el}$  und  $C_{el}$  (zu den 3 Bilinearformen  $a(\mathbf{u}, \mathbf{v}), b(\vartheta, \mathbf{v})$  und  $c(\vartheta, v)$  gehörend) analog zum Programmmodul **SPC-PM3-AdH-inkLE** aufgebaut. Allerdings wird der PCG-Löser für beide Teilgleichungen einzeln aufgerufen:

1.  $C\vartheta = q$  ergibt  $\vartheta$ ,
2.  $A\mathbf{u} = B\vartheta + \underline{b}$  ergibt  $\mathbf{u}$ .

Hier sind  $A, B, C$  wieder die (theoretischen) Assemblierungen der Elementmatrizen  $A_{el}, B_{el}, C_{el}$ . Der Knotenvektor  $\underline{b}$  ist z.Z. nur aus den äußeren Kräften gebildet ( $l_2(\mathbf{v})$ , wenn  $\mathbf{v}$  alle  $\mathbf{u}$ -Ansatzfunktionen durchläuft) und  $\vartheta_0 \equiv 0$ . Bei einer nicht konstanten Initialtemperatur muss  $\underline{b}$  noch aus den entsprechenden Elementintegralen

$$- \int_{el} \vartheta_0 \boldsymbol{\epsilon}(\mathbf{v}) : \mathbf{C} : \mathbf{A} d\Omega$$

( $\forall$  Formfunktionen  $\mathbf{v}$  des Elements) assembliert werden.

Bemerkung zum Eingabefile:

Neben der Festlegung von  $\mathbf{NDof} = 4$  (Freiheitsgrade 1...3 sind die Verschiebungen und 4 die Temperatur) sind mehr Materialparameter nötig als in der Grundvariante. Deshalb sind im Eingabefile unter dem Schlüsselwort **#MATERIAL** in folgender Reihenfolge die folgenden Parameter (einfacher Fall: **A3D-ThEl**) anzugeben:

E-Modul, Querkontraktionszahl  $\nu$ , Wärmeleitkoeffizient  $\kappa$ , Temperatureausdehnungskoeffizient  $\alpha$ , Volumenlast  $(f_1, f_2, f_3)$ , Wärmequelle  $q$ .

Für den anisotropen Fall (**A3D-TEAni**) kommen weitere Parameter zur Faserinformation hinzu.

## 5 Nichtlineare Probleme der großen Deformationen

Der Programmmodul

**SPC-PM3-AdH-LD** (in: **A3D-LD**)

ist die Erweiterung von **A3D-Original** auf nichtlineare Deformationsprobleme. Diese sind gekennzeichnet durch die Beachtung der sog. "großen" oder "finiten" Verzerrungen sowie durch ein nichtlinear-elastisches Materialgesetz. Die theoretischen Grundlagen sind ausführlich in [13] dargelegt. Das Kräftegleichgewicht

$$\operatorname{div} \boldsymbol{\sigma} + \mathbf{f} = 0$$

wird im deformierten Gebiet  $\Omega_t$  verstanden ( $x \in \Omega_t, x = X + \mathbf{u}, X \in \Omega_0$ ) und ins Ausgangsgebiet  $\Omega_0$  "zurückgezogen". Dies ergibt die zu lösende nichtlineare Variationsgleichung zur Definition der Verschiebung  $\mathbf{u}(X)$  als:

$$a(\mathbf{u}; \mathbf{v}) = l(\mathbf{v}) \quad \forall \mathbf{v} \in (\mathbb{H}_D^1(\Omega_0))^3$$

mit

$$a(\mathbf{u}; \mathbf{v}) = \int_{\Omega_0} \mathbb{T}(\mathbf{C}(\mathbf{u})) : \mathbf{E}(\mathbf{u}; \mathbf{v}) d\Omega_0.$$

Wieder hat  $\mathbf{u}$  eventuell vorgegebene Dirichlet-Randdaten auf Randteilen  $\Gamma_D$  ( $\partial\Omega_0 = \Gamma_D \cup \Gamma_N$ ), dort ist  $\mathbf{v} \equiv 0$ .

Zur Markierung von nichtlinearen Abhängigkeiten wird das Semikolon verwendet, indem alle Größen vor dem Semikolon „nichtlinear“ und alle nach dem Semikolon linear eingehen.

Der Tensor 2. Stufe  $\mathbf{E}(\mathbf{u}; \mathbf{v})$  ist die Symmetrisierung von  $(1 + \operatorname{Grad} \mathbf{u}) \cdot \operatorname{Grad} \mathbf{v}^T$  bzw.