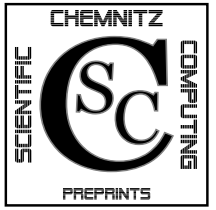
TECHNISCHE UNIVERSITÄT CHEMNITZ

Arnd Meyer

Programmbeschreibung
SPC-PM3-AdH-XX
Teil 2

CSC/14-02



Chemnitz Scientific Computing Preprints

Chemnitz Scientific Computing Preprints – ISSN 1864-0087

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: **S**cientific **P**arallel **C**omputing -
Programm–**M**odul **3D** – **ad**aptiv – **H**exaederelemente.

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

Stand: Ende 2013

CSC/14-02 ISSN 1864-0087 **April 2014**

Inhaltsverzeichnis

1 Vorbemerkungen 1

2 Probleme mit transversal-isotropem Material 1

3 Gleichungen vom Sattelpunktstyp 3

4 Probleme der Thermo-Elastizität 5

5 Nichtlineare Probleme der großen Deformationen 7

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/

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. Kirchhhoff 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/

[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 Nichlinearitäten 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 $\boldsymbol{a}(x)$, das die Vorzugsrichtung („Faserorientierung“) am Punkt x des Gebietes definiert. Die zugrundeliegende Differentialgleichung ist die gleiche wie im isotropen Fall:

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

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

($\boldsymbol{v}$ hat vorgegebene Werte 0 auf Γ_D),

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

mit dem linearen Verzerrungstensor

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

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

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

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

$$\mathfrak{C} = 2\mu\,\mathfrak{I} + \lambda\,(\mathsf{I}\,\mathsf{I}) + \alpha\,(\boldsymbol{aa}\mathsf{I} + \mathsf{I}\boldsymbol{aa}) + 2(\mu_a - \mu)\,\hat{\mathfrak{C}} + \beta\,\boldsymbol{aaaa}$$

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

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 Voigtsche 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 $\boldsymbol{a}(x)$ erzeugt eine technische Schwierigkeit: Nicht alle Problemdaten können (elementorientiert) im Eingabefile gespeichert sein, sondern ein besonderer Funktionsruf der Abb. $x \to \boldsymbol{a}(x)$ ist als Zusatz-Unterprogramm bereitzustellen.

Hierzu wurde *FibreInfo* (in *fibreinfo.f*) neu konzipiert, was (extrem beispielabhängig!) vom Nutzer zu spezifizieren ist. Eine einfachere Version entsteht, wenn $\boldsymbol{a}(x)$ 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, *Betyrachtung 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: Engeneering 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

$$A(\boldsymbol{u})\,\underline{\delta u} = \underline{r}(\boldsymbol{u})$$

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

Die aktuelle Näherung der Knotenwerte von $\boldsymbol{u}$ wird stets auf der Datenstruktur `X(MNode+1...3,j)` $\forall$ `j` gehalten und also nach der Newton-Korrektur $\boldsymbol{u} := \boldsymbol{u} + \delta\boldsymbol{u}$ 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

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

aufzuspalten, wobei

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

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

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

Für die beiden Teile der Materialfunktion wird hier vorläufig

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

benutzt ($\iota = \det\mathsf{C}$ 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

$$\boldsymbol{\sigma} = 2\mu\,\boldsymbol{\epsilon} + \lambda(\operatorname{tr}\boldsymbol{\epsilon})\mathsf{I}$$

auf, wobei die beiden Lamé-Konstanten μ und λ aus E (E-Modul) und ν (Querkontraktionszahl) berechnet werden als

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

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

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

wobei jetzt

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

sind. Die FE-Diskretisierung muss in diesem Fall die LBB-Bedingungen erfüllen, so dass die Benutzung von 2 verschiedenen FE-Ansätzen für $\boldsymbol{u}$ und p notwendig ist. Üblich ist die Benutzung der Taylor-Hood-Elemente (27-Knoten-triquadratische Ansätze für $\boldsymbol{u}$ 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“ $[-1,1]^3 \longleftrightarrow$ „Weltelement“ verantwortlich:

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

mit $\hat{x} \in [-1,1]^3$ und $x^{(i)}$ den `NeX` Knoten des „Weltelements“. Für `NeX` ist 8 oder 27 sinnvoll (dann sind $N_i(\hat{x})$ die trilinearen bzw. triquadratischen Formfunktionen). `NeU` definiert die $\boldsymbol{u}$-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 $\underline{u} \in \mathbb{R}^{\mathtt{Nu}}, \underline{p} \in \mathbb{R}^{\mathtt{Np}}$ 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(`NeU`×`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

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

mit dem Elastizitätstensor $\mathfrak{C}$ (bisher nur isotrop: $\mathfrak{C} = 2\mu\mathfrak{I} + \lambda(\mathsf{II})$ 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

$$(1) \qquad a'(\boldsymbol{u};\delta\boldsymbol{u},\boldsymbol{v}) = tl(\boldsymbol{v}) - a(\boldsymbol{u};\boldsymbol{v}) \qquad \forall \boldsymbol{v}$$

und

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

Wenn

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

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 Δt. Dazu wenige Schritte (ev. totale) Netzverfeinerung. Dann wird durch Einschalten der Inkrementierung (Steuerbyte '1') der Test $(*)$ aktiviert und t nach je $2 \ldots 4$ 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 $a'(\boldsymbol{u};\delta\boldsymbol{u},\boldsymbol{v})$, die aufgrund der zweifachen Abhängigkeit der $a(.;.)$-Form von $\boldsymbol{u}$ sich als

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

ergibt. Der “tangentiale” Materialtensor $\mathfrak{C}$ entsteht durch die 2. Ableitung von $\psi(.)$ nach C als

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

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

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

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

$$2\mathsf{E}(\boldsymbol{u};\boldsymbol{v}) = \operatorname{Grad}\boldsymbol{v} + \operatorname{Grad}\boldsymbol{v}^T + \operatorname{Grad}\boldsymbol{u}\cdot\operatorname{Grad}\boldsymbol{v}^T + \operatorname{Grad}\boldsymbol{v}\cdot\operatorname{Grad}\boldsymbol{u}^T.$$

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

$$\mathsf{C} = (\mathsf{I} + \operatorname{Grad}\boldsymbol{u})\cdot(\mathsf{I} + \operatorname{Grad}\boldsymbol{u}^T) = \mathsf{I} + 2\mathsf{E}(\boldsymbol{u}).$$

Diese Abhängigkeit wird durch die Wahl eines Energiefunktionals

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

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

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

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

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

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

$$\psi(a_1,a_2,a_3) = c_{10}\,(a_1 - \ln(\iota) - 3) + D_2\,\ln(\iota)^2, \quad \iota = \det\mathsf{C} = (a_1^3/6 - a_1a_2 + a_3).$$

Die Lösung des nichtlinearen Systems

$$a(\boldsymbol{u};\boldsymbol{v}) = l(\boldsymbol{v}) \qquad \forall\boldsymbol{v}$$

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

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

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

4 Probleme der Thermo-Elastizität

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

SPC–PM3–AdH–ThEl (in: **A3D-ThEl**)

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

SPC–PM3–AdH–TEAni (in: **A3D-TEAni**)

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

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

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

$$\mathfrak{C} = 2\mu\mathfrak{I} + \lambda(\mathsf{I}\,\mathsf{I}) \qquad \text{und} \qquad \mathsf{A} = \alpha\mathsf{I}$$

benutzt mit dem Temperaturausdehnungskoeffizient α (Maßeinheit: 1/K). Also ergibt sich folgender 2-stufiger Prozeß:

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

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

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

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

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

2. Mit dem Feld ϑ wird das Deformationsproblem gelöst:

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

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

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

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

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

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

$$\int_\Omega \vartheta\, \alpha\, (2\mu + 3\lambda)\ \mathrm{div}\, \boldsymbol{v}\, d\Omega.$$

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

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

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 ($l_2(\boldsymbol{v})$, wenn $\boldsymbol{v}$ alle $\boldsymbol{u}$-Ansatzfunktionen durchläuft) und $\vartheta_0 \equiv 0$. Bei einer nicht konstanten Initialtemperatur muss $\underline{b}$ noch aus den entsprechenden Elementintegralen

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

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

Bemerkung zum Eingabefile:

Neben der Festlegung von `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 ν, Wärmeleitkoeffizient κ, Temperaturausdehnungskoeffizient α, Volumenlast (f_1, f_2, f_3), Wärmequelle q.

Für den anisotropen Fall (**A3D-TEAni**) kommen weitere Parameter zur Faser-Information 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

$$\mathrm{div}\, \boldsymbol{\sigma} + \boldsymbol{f} = 0$$

wird im deformierten Gebiet Ω_t verstanden ($x \in \Omega_t,\ x = X + \boldsymbol{u},\ X \in \Omega_0$) und ins Ausgangsgebiet Ω_0 "zurückgezogen". Dies ergibt die zu lösende nichtlineare Variationsgleichung zur Definition der Verschiebung $\boldsymbol{u}(X)$ als:

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

mit

$$a(\boldsymbol{u};\boldsymbol{v}) = \int_{\Omega_0} \mathsf{T}(\mathsf{C}(\boldsymbol{u})) : \mathsf{E}(\boldsymbol{u};\boldsymbol{v})\, d\Omega_0.$$

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

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 $\mathsf{E}(\boldsymbol{u};\boldsymbol{v})$ ist die Symmetrisierung von $(\mathsf{I}+\mathrm{Grad}\,\boldsymbol{u})\cdot\mathrm{Grad}\,\boldsymbol{v}^T$ bzw.