

CONSTITUTIVE MODELLING OF SHAPE MEMORY ALLOYS AND
UPSCALING OF DEFORMABLE POROUS MEDIA

A Dissertation

by

PETAR ANGELOV POPOV

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2005

Major Subject: Aerospace Engineering

CONSTITUTIVE MODELLING OF SHAPE MEMORY ALLOYS AND
UPSCALING OF DEFORMABLE POROUS MEDIA

A Dissertation

by

PETAR ANGELOV POPOV

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Dimitris C. Lagoudas
(Chair of Committee)

Jay Walton
(Member)

Theofanis Strouboulis
(Member)

Vikram Kinra
(Member)

Helen L. Reed
(Head of Department)

May 2005

Major Subject: Aerospace Engineering

ABSTRACT

Constitutive Modelling of Shape Memory Alloys and Upscaling of Deformable
Porous Media. (May 2005)

Petar Angelov Popov, Dipl. Sofia University "St. Kliment Ohridski", Bulgaria
Chair of Advisory Committee: Dr. Dimitris C. Lagoudas

Shape Memory Alloys (SMAs) are metal alloys which are capable of changing their crystallographic structure as a result of externally applied mechanical or thermal loading. This work is a systematic effort to develop a robust, thermodynamics based, 3-D constitutive model for SMAs with special features, dictated by new experimental observations. The new rate independent model accounts in a unified manner for the stress/thermally induced austenite to oriented martensite phase transformation, the thermally induced austenite to self-accommodated martensite phase transformation as well as the reorientation of self-accommodated martensite under applied stress. The model is implemented numerically in 3-D with the help of return-mapping algorithms. Numerical examples, demonstrating the capabilities of the model are also presented.

Further, the stationary Fluid-Structure Interaction (FSI) problem is formulated in terms of incompressible Newtonian fluid and a deformable solid. A numerical method is presented for its solution and a numerical implementation is developed. It is used to verify an existing asymptotic solution to the FSI problem in a simple channel geometry. The SMA model is also used in conjunction with the fluid-structure solver to simulate the behavior of SMA based filtering and flow regulating devices.

The work also includes a numerical study of wave propagation in SMA rods. An SMA body subjected to external dynamic loading will experience large inelastic deformations that will propagate through the body as phase transformation and/or

detwinning shock waves. The wave propagation problem in a cylindrical SMA is studied numerically by an adaptive Finite Element Method. The energy dissipation capabilities of SMA rods are estimated based on the numerical simulations. Comparisons with experimental data are also performed.

To my parents, Galia and Angel

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor Dr. Dimitris Lagoudas for his guidance and support during the course of my PhD studies. He taught me the art of being a researcher and the knowledge and experience I gained have been life changing.

I am especially grateful to Dr. Jay Walton, both for serving on my committee and for the opportunity to learn continuum mechanics, both through discussions and classes. A PhD is, above all, an educational experience. Of the many classes I have taken, throughout undergraduate and graduate school, I will certainly remember the one on Continuum Mechanics, taught by Dr. Walton.

I owe special thanks to Dr. Oleg Iliev for his help and mentoring during my visit to the Fraunhofer-Institut für Techno- und Wirtschaftsmathematik (Institute for Industrial Mathematics) in Kaiserslautern, Germany. This internship was essential for the work on fluid-structure interaction problems and I greatly appreciate it. I would also like to thank Dr. Heiko Andrä for guidance and advice during this project.

I owe my knowledge and understanding of homogenization and flow in porous media entirely to Dr. Yalchin Efendiev. While the discussions we had could not materialize into this dissertation, I certainly hope that one day they will. Also, thanks to Dr. Bojan Popov, now I understand much more about the theory of non-linear hyperbolic systems than at the time the research in this thesis was performed.

I would like to acknowledge also the financial support of the Aerospace Engineering Department at Texas A&M University, the Fraunhofer-Institut für Techno- und Wirtschaftsmathematik and, despite being a pacifist, that support of the Air Force Office of Scientific Research and the Office of Naval Research.

Last, but not least, I would like to thank Mughees Khan, Parikshith Kumar, Eric Vandygriff as well as many other colleagues and friends whose advice and sup-

port greatly contributed to this work. Also, special thanks to Luciano Machado and Yetzilah Urthaler for reading and correcting parts of this manuscript. The support and understanding of Yetzilah Urthaler is deeply appreciated.

TABLE OF CONTENTS

CHAPTER		Page
I	INTRODUCTION	1
	1. Properties of martensitic phase transformations	3
	1.1. Shape memory effect	8
	1.2. Pseudoelasticity	10
	2. Modelling of polycrystalline SMAs	16
	3. Literature review of phenomenological models	20
	3.1. Transformation surface	23
	3.2. Detwinning of martensite	24
	3.3. Cyclic loading and transformation induced plastic strain	25
	4. Outline of current research	26
	4.1. Development of an adaptive finite element method for wave propagation problems in SMAs	27
	4.2. Development of a three phase, 3-D constitutive model for SMAs	28
	4.3. Numerical implementation of SMA constitutive model	29
	4.4. Investigation of fluid-structure interaction problems with application to SMA based flow regulating devices	30
II	NONLINEAR WAVE PROPAGATION IN SMA RODS	32
	1. Field equations and constitutive model for the impact prob- lem of SMA rods	37
	1.1. Field equations, initial and boundary conditions	38
	1.2. Uniaxial thermomechanical constitutive model for poly- crystalline SMAs	39
	1.2.1. Stress induced martensite	39
	1.3. Detwinning of martensite	42
	1.4. Isentropic approximation	43
	1.5. Tangent moduli	44
	2. Numerical implementation	44
	2.1. FEM procedure	45
	2.2. Adaptive mesh refinement	48
	3. Numerical examples	49
	3.1. Step loading problem	50

CHAPTER	Page
3.1.1. Analytical solutions to the step loading problem	51
3.1.2. Numerical results for the step loading problem	53
3.2. Square pulse loading problem in pseudoelastic conditions	58
3.3. Detwinning induced by a pulse load	62
4. Comparisons with impact loading experiment of an SMA rod	66
4.1. Experimental data	66
4.2. Numerical simulations of the Hopkinson bar experiment	70
4.2.1. Quasi-static hysteresis and calibration of the SMA model	70
4.2.2. Dynamic hysteresis and rate independence	71
4.2.3. Numerical simulations	72
5. Conclusions	77
III A THREE PHASE, 3-D CONSTITUTIVE MODEL FOR POLYCRYSTALLINE SMAS	79
1. Experimental results on the transformation temperatures of twinned and detwinned martensite to austenite.	80
1.1. Setup and experimental procedure	80
1.2. Successive DSC results	81
2. Modified SMA phase diagram	86
2.1. Austenite to martensite ($A \leftrightarrow M^t, A \leftrightarrow M^d$).	90
2.2. Detwinning of self accommodated martensite ($M^t \rightarrow M^d$)	92
2.3. Combined austenite to detwinned martensite at low stresses	94
3. Description of the constitutive theory	95
3.1. Kinematic assumptions	95
3.2. Energy balance and entropy inequality	98
3.3. Constitutive assumptions for the SMA material	100
3.3.1. Gibbs free energy for a polycrystalline SMA	102
3.4. Transformation surfaces and flow rules	109
4. Determination of material parameters	113
4.1. Uniaxial reduction of the model	114
4.2. The uniaxial transformation strips and the phase diagram	118
4.3. Limitations	122
5. Uniaxial examples	123
5.1. Constrained SMA rod	124
5.1.1. Constrained cooling of a rod	124
5.1.2. Constrained cooling/heating of a rod	127

CHAPTER	Page
IV	NUMERICAL IMPLEMENTATION OF SMA CONSTITUTIVE MODEL 129
	1. Closest point projection algorithm for multiple internal variables 131
	1.1. The loading step 132
	1.2. Closest point projection return mapping algorithm for SMA constitutive model 133
	1.2.1. Thermoelastic prediction 135
	1.2.2. Transformation correction 136
	1.2.3. Active surfaces and other implementation details 139
	1.3. Algorithmic tangent stiffness (Jacobian) 140
	2. Numerical examples 145
	2.1. Constrained heating of a perforated square 146
	2.2. A 3-D structural member 155
V	FLUID-STRUCTURE INTERACTION PROBLEMS WITH APPLICATIONS TO SMA BASED DEVICES 168
	1. The fluid-structure interaction (FSI) Problem 169
	1.1. Solid 169
	1.2. Newtonian fluid at low Reynolds number 172
	1.3. Statement of the coupled FSI problem 173
	1.4. Weak form of the elasticity, Stokes and FSI problems 175
	2. Solution methods for the coupled FSI system 178
	2.1. Dirichlet-Neumann iterative scheme 178
	2.2. FEM approximation of the FSI problem 180
	2.3. Solution methods for the solid and fluid subproblems 182
	2.3.1. Linearized elasticity problem 182
	2.3.2. Stokes problem 185
	2.3.3. Fluid mesh regeneration 188
	3. Model problems with a linearized elastic solid 191
	3.1. Flow through an elastic channel 192
	3.2. Permeability of a long elastic channel 195
	3.3. Flow through a channel with elastic segment 197
	4. Modelling of flow regulating SMA device 200
VI	SUMMARY AND CONCLUSIONS 208
	REFERENCES 211

	Page
APPENDIX A: AN ASYMPTOTIC SOLUTION OF THE FSI SYSTEM FOR AN ELASTIC CHANNEL	230
1. Dimensionless form	231
1.1. Dimensionless Stokes equations	232
1.2. Dimensionless elasticity problem	233
2. Asymptotic expansion	234
2.1. Asymptotic expansion for the Stokes system	235
2.2. Expansion for the solid-fluid interface	236
2.3. Asymptotic expansion for elasticity system	237
APPENDIX B: INTEGRATION OF A RETURN MAPPING ALGORITHM IN A DISPLACEMENT BASED FINITE ELEMENT METHOD	241
APPENDIX C: SMA SUBROUTINE SOURCE CODE LISTING	245
VITA	279

LIST OF TABLES

TABLE		Page
I	Material parameters used in the uniaxial SMA model.	50
II	Execution times for fixed and adaptive meshes.	58
III	Material parameters used in the uniaxial SMA model for detwinning.	71
IV	Material parameters used in the three phase, 3-D SMA model. . . .	124
V	Convergence of the Dirichlet-Neumann iterative scheme for the elastic channel FSI problem.	193
VI	Comparisons of analytical and asymptotic results for a long elastic channel in the L^2 norm.	196
VII	Point-wise comparisons of analytical and asymptotic results of a long elastic channel.	197
VIII	Permeability of a channel with elastic segment as function of the flow rate.	199

LIST OF FIGURES

FIGURE		Page
1	Stress-Temperature diagram showing the relationship of stress and temperature and the austenitic and martensitic domains. . . .	7
2	Stress-Strain-Temperature schematic of the crystallographic changes involved in the Shape Memory Effect.	9
3	Isothermal and isobaric pseudoelastic loading paths	11
4	Schematic of isothermal, pseudoelastic stress-strain curve	13
5	Pseudoelastic response of Ni50Ti50 (% <i>at.</i>) wire specimen during the first 20 cycles $T = 70\text{ }^{\circ}\text{C}$	14
6	Schematic of isobaric, cooling/heating cycle.	15
7	Schematic of the loading portion of a stress-strain relationship and the critical points defining the solution to the problem.	51
8	Difference between adiabatic and isothermal loading.	54
9	Comparisons of FEM solutions with different mesh sizes and time steps.	56
10	Adaptive meshing and numerical dissipation.	56
11	Stress profile at different instances of time for a square pulse in adiabatic loading.	60
12	Magnified view of solution to square pulse adiabatic problem. . . .	60
13	Temperature profile at various times.	61
14	Energy dissipation for a $10\text{ }\mu\text{s}$ square pulse in adiabatic conditions.	63
15	Stress profiles at various times for a square pulse in detwinning conditions.	63

FIGURE	Page
16	Magnified view of the stress profiles. 64
17	Energy dissipation for a 10 μs square pulse (detwinning). 64
18	Geometry and arrangement of strain gauges in Hopkinson apparatus. 67
19	Experimental data on wave propagation in an SMA rod. 68
20	Quasi-static hysteresis of SMA specimen and the model simulation. 70
21	Variation of the wave speed with the strain level as determined from the strain measurements 73
22	Stress-strain response evaluated from one-dimensional wave prop- agation measurements and quasi-static data. 74
23	An adaptive FE analysis of experimental data using SMA consti- tutive model with linear hardening. 75
24	An adaptive FE analysis of experimental data using deformation plasticity model. 75
25	DSC results after a single mechanical loading of an untrained NiTi wire. 82
26	DSC results after a single mechanical loading of a trained NiTi wire. 85
27	The SMA phase diagram used in this work. 87
28	Schematic of the three phases and the possible transitions between them. 97
29	Transformation strips for the $A \leftrightarrow M^t$ and $A \leftrightarrow M^d$ as predicted by the SMA model. 121
30	A constrained cooling path in stress-temperature space. 126
31	Simulated loading path involving simultaneous reverse transformation. 128
32	Schematic of the mechanical boundary conditions applied for the perforated square model problem. 147

FIGURE	Page
33	Schematic of the loading path followed for the boundary conditions. 147
34	Basic mesh used to perform the first loading step, $T = 260$ 149
35	First refinement, $T = 260$, end of first loading step. 150
36	Second refinement, $T = 260$, end of first loading step. 151
37	Third refinement, $T = 260$, end of first loading step. 152
38	Fourth refinement, $T = 260$, end of first loading step. 153
39	Solution for square hole problem at $T = 260$ °K and $T = 280$ °K. 154
40	Solution for square hole problem at $T = 310$ °K 157
41	Solution for square hole problem at $T = 320$ °K. 158
42	Solution for square hole problem at $T = 330$ °K. 158
43	Solution for square hole problem at $T = 335$ °K. 159
44	Solution for square hole problem at $T = 340$ °K. 159
45	Solution for square hole problem at $T = 350$ °K. 160
46	Solution for square hole problem at $T = 360$ °K. 160
47	Schematic of the 3-D SMA structural member geometry. 161
48	Von-Mises stress in the 3-D SMA structural member at $T = 260$ °K. 162
49	M^d volume fraction in the 3-D SMA structural member at $T =$ 260 °K. 163
50	M^d volume fraction in the 3-D SMA structural member at $T =$ 330 °K. 164
51	Von-Mises stress in the 3-D SMA structural member at $T = 360$ °K. 165
52	M^d volume fraction in the 3-D SMA structural member at $T =$ 360 °K. 166

FIGURE	Page
53	Evolution of the effective stress during heating. 167
54	Schematic of the fluid and solid domains of the FSI problem. 173
55	A 2D degenerate mesh. 189
56	A 3D degenerate mesh. 190
57	Geometry and boundary conditions for flow in channel with elastic walls. 193
58	Solution of the FSI problem for $V_{max} = 1/4$ 194
59	Geometry of a channel with an elastic segment. 198
60	Final configuration of the fluid domain Ω^f for $Q = 15$ 199
61	Permeability of a channel with elastic segment. 200
62	Geometry of SMA flow regulating device. 201
63	Geometry of the initial and deformed configuration of the channel while cooling. 203
64	FSI Solution at $T = 320 \text{ }^\circ K$ 205
65	FSI Solution at $T = 300 \text{ }^\circ K$ 206
66	Dependence of the flow rate in the channel as a function of temperature during cooling. 207
67	Schematic of a section of length L a long elastic channel. 230

CHAPTER I

INTRODUCTION

Shape Memory Alloys (SMAs) are metallic alloys that can undergo martensitic phase transformations as a result of applied thermomechanical loads and are capable of recovering permanent strains when heated above a certain temperature. At high temperatures the crystal lattice is in a high symmetry, parent austenitic phase. The key characteristic of all SMAs is the occurrence of a martensitic phase transformation between the austenitic phase and the different variants of the low temperature, low symmetry martensitic phase. The martensitic transformation is a shear-dominant diffusionless solid-state phase transformation occurring by nucleation and growth of the martensitic phase from the parent austenitic phase (Olson and Cohen, 1982). What make SMAs remarkably different from other materials are primarily the *Shape Memory Effect* (SME) and *Pseudoelasticity*, which are associated with the specific way the phase transformation occurs.

When a shape memory alloy undergoes a martensitic phase transformation, it transforms from its high-symmetry, usually cubic austenitic phase to a low-symmetry martensitic phase, such as the monoclinic variants of the martensitic phase in a NiTi SMA. In the absence of applied stresses, the variants of the martensitic phase usually arrange themselves in a self-accommodating manner through twinning, resulting in no observable macroscopic shape change. By applying mechanical loading the martensitic variants are forced to reorient (detwin) into a single variant leading to large macroscopic inelastic strains. After heating above certain temperature, the martensitic phase returns to the austenitic phase, and the inelastic strains are re-

The journal model is Mechanics of Materials.

covered. This behavior is known as the SME. Pseudoelasticity is observed when the martensitic phase transformation is induced by applied thermomechanical loading of the austenitic phase in which case detwinned martensite is directly produced from austenite. The process is associated with large inelastic (transformation) strains which are recovered upon unloading due to the reverse phase transformation (Wayman, 1983). The extensive list of alloys exhibiting SME and pseudoelasticity includes the Ni-Ti alloys, and many copper-, iron-, silver- and gold-based alloys (Nishiyama, 1978).

Martensitic transformations are usually divided into two groups - thermoelastic and nonthermoelastic. The nonthermoelastic transformations occur mainly in ferrous alloys and are associated with non-mobile martensite-parent phase interfaces pinned by permanent defects and proceed by successive nucleation and growth. Due to re-nucleation of austenite during the reverse (martensite to austenite) transformation, these transformations are crystallographically nonreversible in the sense that the martensite cannot revert to the parent phase in the original orientation. The thermoelastic martensitic transformations, on the other hand, are associated with mobile interfaces between the parent and martensitic phases. These interfaces are capable of “backward” movement during the reverse transformation by shrinkage of the martensitic plates rather than nucleation of the parent phase, which leads to a crystallographically reversible transformation (Otsuka and Wayman, 1999). The unique properties of SMAs (i.e. *Shape Memory Effect, Pseudoelasticity*) are the result of thermoelastic martensitic transformation.

For a review of commonly used SMAs, their chemical composition, mechanical properties and the kinematics of phase transformations in single crystals the reader is referred to Patoor et al. (2005). In the next Section 1 some general aspects of the martensitic phase transformation in SMAs are reviewed and the complex thermome-

chanical response of SMAs is described.

1. Properties of martensitic phase transformations

In SMA materials, the martensitic phase transformation is a rate independent, reversible, crystallographic reorientation process between the two stable phases. Summarized below are the main characteristics of martensitic phase transformations that distinguish them among other solid state transformations:

- The transformation is diffusionless, rate-independent (Nishiyama, 1978), inelastic deformation of the crystal lattice; it is a results of cooperative and collective motion of atoms over distances smaller than the lattice parameters (Otsuka and Wayman, 1999).
- Latent heat is generated during the transformation (Jackson et al., 1972; Otsuka and Wayman, 1999); The transformation is thus first order and parent and product phases can coexist (Landau et al., 1976).
- Several variants of martensite may be formed from a single austenitic crystal (De Vos et al., 1978).
- The transformation produces volumetric and shear strains along well defined planes of a crystallographic unit. The shear strain can be an order of magnitude larger than the pure elastic deformation of the crystal lattice (Otsuka and Wayman, 1999).
- The phase transformation is highly sensitive to temperature and applied stresses (Cross et al., 1969; Delaey, 1990; Jackson et al., 1972).

When the martensitic transformation takes place, numerous physical properties are modified. During the transformation, a latent heat associated with the transformation is absorbed or released based on the transformation direction. The forward, austenite-to-martensite transformation (denoted $A \rightarrow M$) is accompanied by the release of heat corresponding to a change in the transformation enthalpy (exothermic phase transformation). The reverse, martensite-to-austenite transformation (denoted $M \rightarrow A$) is an endothermic phase transformation accompanied by absorption of thermal energy. For a given temperature, the amount of heat is proportional to the volume fraction of the transformed material. The two phases also have different resistance due to their different crystallographic structures, so the phase transformation is associated with a change in the electrical resistivity. These changes allow for the measurement of the transformation temperatures. Generally, differential calorimetry and electrical resistance are used for that purpose. In addition, thermoelectrical power, x-ray analysis, acoustic waves, interior friction, and the measure of Young's modulus are also used.

Transformation in the absence of externally applied stress field. Since the phase transformation is very sensitive to applied external stress, first the case of zero applied stress is considered. The forward $A \rightarrow M$ transformation occurs when the free energy of martensite becomes less than the free energy of austenite. A critical temperature T_0 is assumed to exist, at which the free energies of the two phases are equal. As the temperature is lowered below T_0 , the transformation does not begin exactly at T_0 but, at a temperature M_s (martensite start temperature at zero stress), which is less than T_0 . The free energy necessary for nucleation and growth is responsible for this shift (Delaey, 1990). The transformation continues to evolve as the temperature is lowered until a temperature M_f (martensitic finish temperature at zero stress) is reached. When the SMA is heated from the martensitic phase the reverse

martensite to austenite transformation (denoted $M \rightarrow A$) begins at the temperature A_s (austenitic start temperature at zero stress), higher than T_0 . The transformation continues until a temperature A_f (austenite finish at zero stress) is reached and the material is entirely in the austenitic phase. The equilibrium temperature T_0 is approximately $(M_s + A_f)/2$ (Tong and Wayman, 1974).

For shape memory alloys, the transformation temperatures M_s , M_f , A_s , A_f , and the difference $M_s - M_f$, $A_f - A_s$ are important factors in characterizing shape memory behavior. The transformation temperatures depend mainly on the alloy's composition and processing. Microstructural defects, degree of order in the parent phase, and grain size of the parent phase can also alter the transformation temperatures by several degrees (MacQueron et al., 1991). Note that the existence of exact temperatures M_s , M_f , A_s and A_f is an assumption. In reality, small amounts of either phase can exist beyond the finish temperatures and be formed before the start temperatures.

As will be discussed in Section 1.1, upon cooling of the SMA, the martensite is generated in twinned structures (Otsuka and Wayman, 1999), which accommodate the transformation strain of the individual martensitic variants. Two twinned variants are formed on each side of a mirror plane, so that no observable macroscopic strain is generated. The formation of twins can be hampered by introduction of defects and inhomogeneities introduced during the manufacturing or training of the material (Otsuka and Wayman, 1999)

Effects of externally applied stress. Due to the displacive character of the martensitic transformation, applied stress plays a very important role. The application of stress on a volume of austenite has the effect of orienting the different martensitic variants in the direction of the stress during the $A \rightarrow M$ phase transformation. This leads to the development of large inelastic strains composed of a shearing component that induces shape change and an expansion component that effects volume change.

The shearing component is the dominating one and is oriented in the direction of the stress.

This oriented, or *detwinned* martensite is metallurgically the same as the twinned martensite discussed previously, however, the presence of inelastic strains has significant implications. As a result positive work (the inelastic strains and stress have the same direction) is being done by the material, thus its total energy is reduced. Therefore, the $A \rightarrow M$ phase transformation starts at higher temperature (i.e. earlier) compared to the zero stress case. A more quantitative calculation (see (Otsuka and Wayman, 1999; Roytburd and Pankova, 1985; Wollants et al., 1979) shows that the equilibrium temperature T_0 obeys a linear relationship in stress-temperature space.

The development of large inelastic strains during some types of martensitic deformations and not in others is of primary engineering concern when modeling SMA based components and devices. Moreover, as it will be discussed in the next Section 1.1, twinned martensite, can be forced to reorient or *detwin* if an appropriate stress field is applied. Hence, in this work a distinction is made between the twinned and detwinned martensitic phases. While this distinction is not based on metallurgical considerations it will be very instrumental in developing accurate models for SMAs. In the rest of the thesis, the twinned martensite will be denoted by M^t and the detwinned by M^d , wherever this distinction is necessary.

The phase transformation (both forward and reverse) occur over a strip in temperature - stress space. This is mainly due to inhomogeneities in the stress and strain fields inside the material. These parameters are very sensitive to the internal state of the material and to the loading mode (tension, compression, shear). A convenient way of describing the regions of stability of the different phases is the phase diagram shown Figure 1. The SMA can exist in either austenite and martensite, and the martensite can further be either in twinned or detwinned. There are three

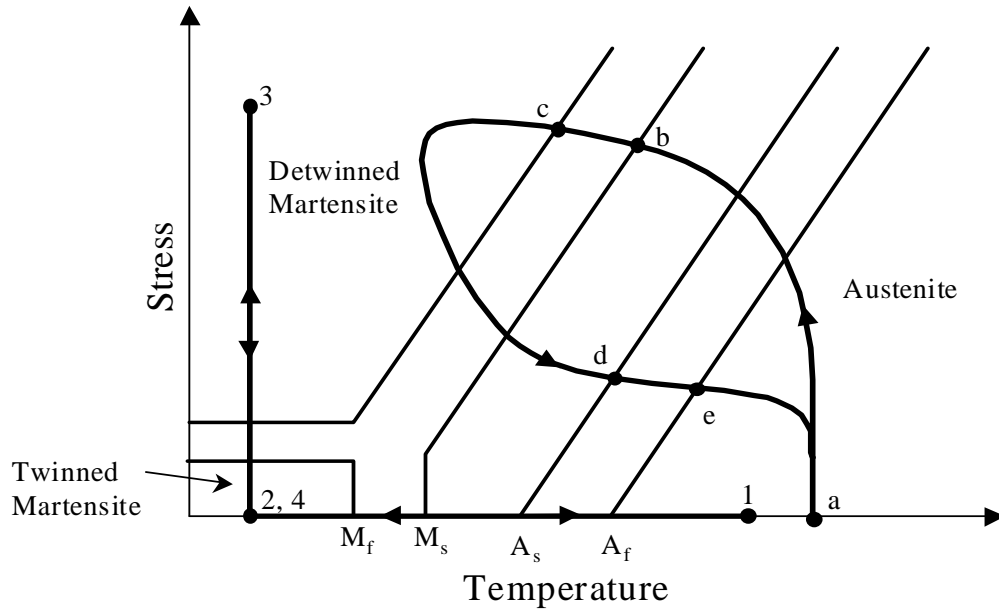


Fig. 1. Stress-Temperature diagram showing the relationship of stress and temperature and the austenitic and martensitic domains.

regions where the material can be in a pure phase along with the transformation lines (surfaces in multiple dimensions) which separate them. When the material crosses a transformation surface it undergoes phase transformation ($A \rightarrow M^t$, $A \rightarrow M^d$, $M^t \rightarrow A$, $M^d \rightarrow A$) or detwinning ($M^t \rightarrow M^d$). In some areas, a mixture of the three phases is also possible. The critical temperatures are denoted on the temperature axis. In addition, the critical uniaxial stresses for beginning σ_s and finish σ_f of the detwinning deformation are shown on the stress axis. The shape memory effects and pseudoelasticity will be discussed in the next two sections with the help of the two loading paths shown on the figure.

This form of the phase diagram has been proposed initially by Brinson and Lammering (1993), based on the experimental observations of Cross et al. (1969); Delaey (1990); Jackson et al. (1972). The diagram has been successively refined by other authors Bekker and Brinson (1997, 1998); Sakamoto (2002), however some ba-

sic questions have remained unanswered. An example is, starting from fully twinned martensitic state (at $T < M_f$), what is the transformation line for the $M^t \rightarrow M^d$ deformation at temperature $M_f < T < A_s$? Based in recent experimental observations performed in the course of this work, the above phase diagram will be extended and modified (Chapter III) to accommodate such cases.

1.1. Shape memory effect

As mentioned earlier, the *Shape Memory Effect* (SME) is a property of SMAs undergoing thermoelastic martensitic transformation. It is exhibited when the SMA is deformed while in the martensitic phase and then unloaded while still at a temperature below M_f . When subsequently heated above A_f it regains its original shape by transforming back into the austenitic phase. A typical loading path $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1$, in which the SME is observed is shown in Figure 1. The same path, schematically plotted in Stress-Strain-Temperature space is shown in Figure 2. During the cooling of the parent phase ($1 \rightarrow 2$) it transforms to twinned martensite. The material is then loaded ($2 \rightarrow 3$) causing stress induced detwinning and development of inelastic strains. Upon unloading ($3 \rightarrow 4$) the material remains in detwinned state and the inelastic strains are not recovered. Finally, when it is heated above A_f ($4 \rightarrow 1$), the SMA returns to its cubic parent phase and the inelastic strains are recovered.

The crystallographic changes during this loading path are explained next. The stress-free cooling of austenite produces self-accommodating growth of the martensitic variants ($1 \rightarrow 2$) such that there is no macroscopic transformation strain (Saburi et al., 1980). The self-accommodated morphology is a characteristic of the crystallography of the alloy used. For example, in copper-based alloys, twenty-four variants of martensite constitute six self-accommodated groups scattered around the $\langle 011 \rangle$ poles of austenite with a typical diamond morphology. The growth of such groups

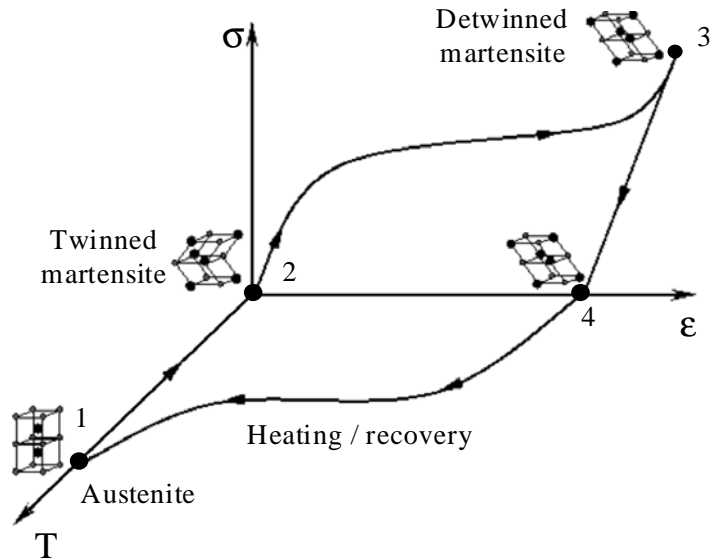


Fig. 2. Stress-Strain-Temperature schematic of the crystallographic changes involved in the Shape Memory Effect.

produces no macroscopic transformation strain, but the multiple interfaces present in these structures (boundaries between the martensite variants and twinning interfaces) are very mobile. This great mobility is at the heart of the shape memory effect. Movement of these interfaces accompanied by detwinning is obtained at stress levels far lower than the plastic yield limit of martensite. This mode of deformation, called *reorientation of variants*, dominates at temperatures lower than M_f .

During the second stage ($2 \rightarrow 3$), the mechanical loading in the martensitic phase leads to reorientation of the variants and results in development of large inelastic strains. This inelastic strain is not recovered upon unloading ($3 \rightarrow 4$). During the last step ($4 \rightarrow 1$), heating the sample above A_f induces the reverse transformation and recovers the inelastic strain. When the material approaches A_f , the martensitic phase becomes unstable in the absence of stress. This results in a complete transformation to the parent phase. Since martensite variants have been reoriented by

stress, the reversion to austenite produces a large transformation strain having the same amplitude but opposite direction to the inelastic strain. As a result, the SMA returns to the original shape it had in the austenitic phase.

1.2. Pseudoelasticity

The *pseudoelastic* behavior of SMAs is associated with stress induced strain recovery upon unloading at temperatures above A_f . Under most general conditions, pseudoelastic thermomechanical loading paths start at zero stress in the austenitic region, then move to the detwinned martensite region and then unload again to the starting point. An example is the path $a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow a$ shown on Figure 1. Other examples are the isothermal and isobaric loading paths shown schematically on Figure 3. For clarity, the initial loading from austenite to achieve the required constant stress for the isobaric path is not shown. Note that isothermal condition can be achieved only by quasi-static (small strain increments) loadings, so that the latent heat generated/absorbed during the phase transformation has time to dissipate. For convenience, in this section mostly isobaric or isothermal loading paths will be considered.

Consider the thermomechanical loading path $a \rightarrow b \rightarrow \dots \rightarrow a$, which starts at zero stress level, above A_f . When the material is loaded at temperatures above A_f , the parent phase (Austenite) undergoes thermoelastic loading up to a critical stress level called the *Transformation stress* ($a \rightarrow b$). At this stress level the material undergoes a stress induced phase transformation ($b \rightarrow c$) from austenite to martensite during which large inelastic strains are developed. Any subsequent loading in the detwinned martensitic region ($c \rightarrow d$) does not produce further phase transformation, although reorientation of the martensitic twins may occur in multi-axial loading conditions. When the point (d) is reached, the reverse transformation begins (martensite-to-

austenite), leading to recovery of the inelastic strains. The material fully transforms to austenite at (e) and the final segment of the loading path $(e \rightarrow a)$ is characterized by recovery of the thermoelastic strains, leading to zero macroscopic strains upon completion of the path. The transformation process results in a hysteresis which reflects the energy dissipated in the cycle.

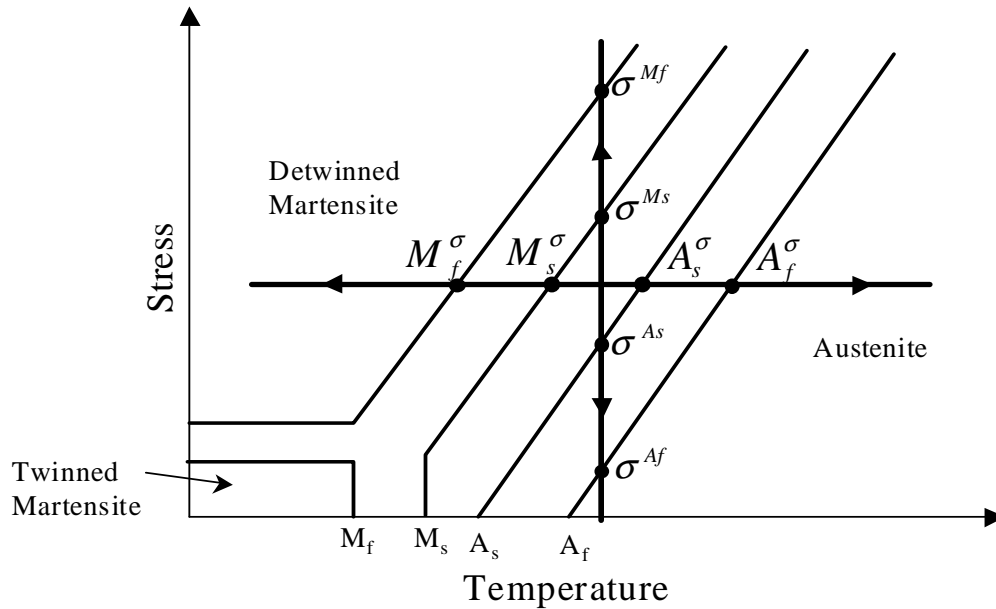


Fig. 3. Isothermal and isobaric pseudoelastic loading paths. The two most commonly encountered pseudoelasticity loading paths - and isothermal and isobaric one. For clarity, the initial loading from austenite to achieve the required constant stress for the isobaric path is not shown.

To better illustrate the uniaxial stress strain response, consider the (quasistatic), isothermal loading path shown on Figure 3, and the corresponding schematic of the stress-strain response in Figure 4. The path begins at zero-stress, zero strain state in the austenitic region and proceeds by monotonously increasing the stress level.

The initial material response is elastic, until the critical stress σ^{Ms} is reached¹. At this point on the stress-temperature diagram, the material hits the forward $A \rightarrow M^d$ transformation strip and begins to develop large inelastic strains. The transformation proceeds until the critical stress σ^{Mf} is reached. The maximal uniaxial transformation strain is denoted by H^t (Figure 4). Further increase in stress leads to elastic material response. The transformation region is characterized by significant decrease in stiffness as can be seen from stress-strain response in Figure 4. Upon unloading, the material response is again linear until the stress decreases to the critical level σ^{As} , when the reverse transformation strip $M^d \rightarrow A$ is reached. A similar decrease in stiffness is observed during the reverse transformation. The completion of the reverse transformation occurs at a critical stress level σ^{Af} . Further unloading is characterized again by elastic response. A typical pseudoelastic SMA response of a NiTi specimen is shown in Figure 5.

The response of an SMA under isobaric cooling/heating exhibits similar hysteretic behavior. Under cooling at constant applied stress σ (Figures 3 and 6) the material initially shows small thermoelastic change in strain until the critical temperature M_s^σ is reached. The forward phase transformation $A \rightarrow M^d$ is characterized again by development of large transformation strain H^t and is complete when the martensitic finish temperature M_f^σ is reached. Further cooling leads to only thermoelastic change in strain. The reverse transformation occurs when the material is heated to temperature A_s^σ and is complete by A_f^σ .

In some cases aging of martensite phase can enable reversal of martensitic twins. This phenomenon of reversible detwinning and twinning of the martensitic variants

¹Note, that the critical stress is a function of temperature, since the $A \rightarrow M^d$ transformation strip is temperature dependent, cf. Figures 1 and 3. In general, the correct notation is $\sigma^{Ms}(T)$, where T is the temperature, however for isothermal examples, like the current one, the temperature dependency can be omitted.

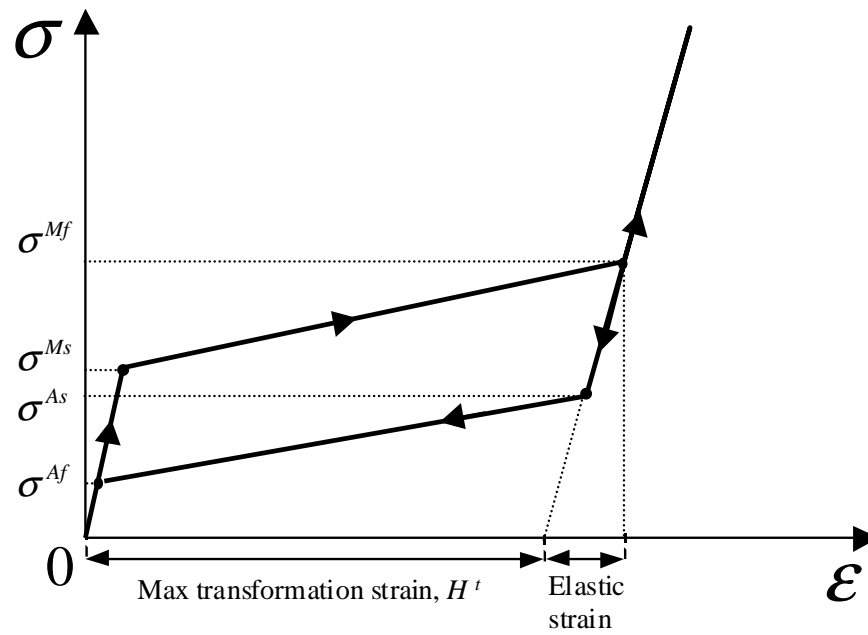


Fig. 4. Schematic of isothermal, pseudoelastic stress-strain curve. The critical stresses σ^{Ms} , σ^{Mf} , σ^{As} and σ^{Af} required for initiation and completion of the forward and reverse transformation are obtained from the phase diagram on Figure 3. The maximum uniaxial transformation strain H^t is also shown.

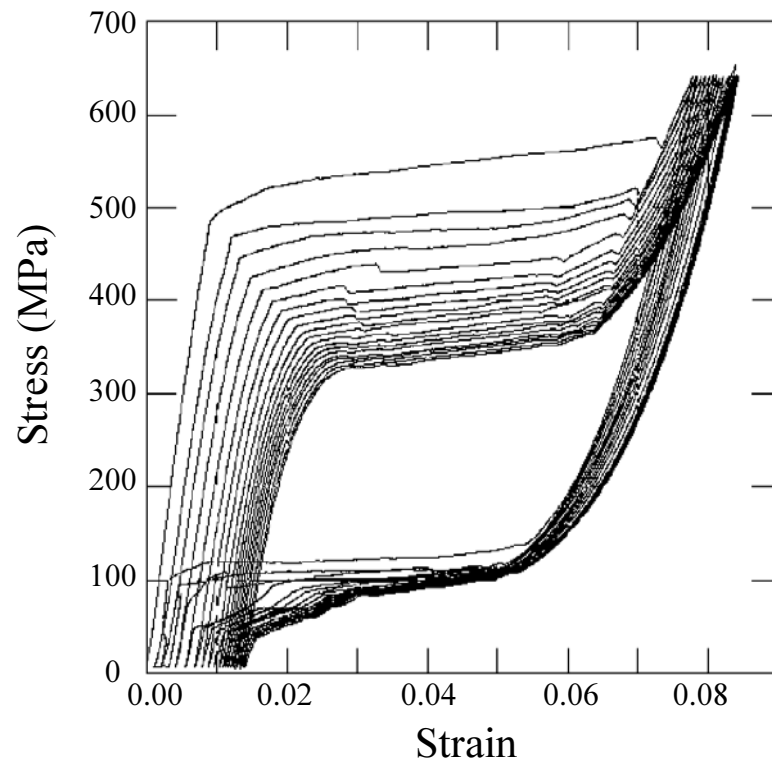


Fig. 5. Pseudoelastic response of Ni50Ti50 (%*at.*) wire specimen ($A_f = 65\text{ }^\circ\text{C}$) during the first 20 cycles $T = 70\text{ }^\circ\text{C}$. After the first few cycles the hysteresis of the material stabilizes (Kumar et al., 2003).

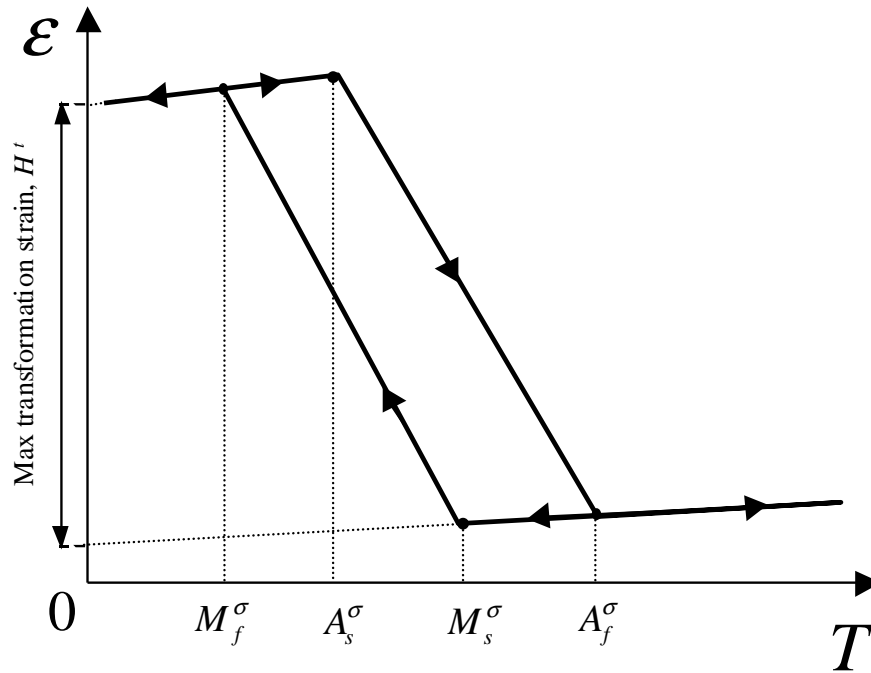


Fig. 6. Schematic of isobaric, cooling/heating cycle. The critical temperatures M_s^σ , M_f^σ , A_s^σ and A_f^σ required for initiation and completion of the forward and reverse transformation are obtained from the phase diagram on Figure 3.

creates a S-S curve similar to the pseudoelastic curve. This phenomenon is called *rubber-like* effect (Otsuka and Wayman, 1999). This effect, first observed in a Au-Cd alloy (Olander, 1932), constitutes the first studied manifestation of shape memory effect. This type of behavior is a characteristic of the martensitic phase ($T < M_f$). It is observed in specific SMA materials, and unlike the superelastic phenomenon, this involves reorientation within the same phase. Since twin boundaries are very mobile, the critical stress required to move them is very small (a few MPa). Temperature plays only a secondary role in this behavior since there is no phase change. When the motion of twin boundaries is not reversible, the loading/unloading path results in an inelastic strain. However, when the twin boundary motion is reversible during unloading (e.g., in Au-Cd alloys), the macroscopic strain thus obtained is composed of the usual elastic strain and a reversible component associated with the movement of these interfaces. Some authors (Otsuka et al., 1976; Otsuka and Wayman, 1999) use the term *pseudoelasticity* to denote both austenite to detwinned martensite phase transformations and the *rubber-like* effect of reversible detwinning of twinned martensite. In order to distinguish between the two, the term *superelasticity* is used for the first process. In this work the rubber-like effect is not considered, so the term *pseudoelasticity* will be used throughout in the sense of *superelasticity* as defined by Otsuka et al. (1976); Otsuka and Wayman (1999).

2. Modelling of polycrystalline SMAs

During the last two decades the area of constitutive modeling of *polycrystalline* SMAs has been a topic of many research publications and significant advancements have been reported. The majority of the constitutive models reported in the literature can be formally classified to belong to one of the two groups:

- micromechanics-based models
- phenomenological models.

The micromechanical models attempt to predict the SMA response by taking into account the granular microstructure of polycrystalline SMAs. Theoretically, if the micro-structure is well-known it is possible to use the well developed knowledge of single crystal SMA behavior and solve boundary value problems in a polycrystalline material. Anand and Gurtin (2003) for example, use a model for single crystal SMA and perform representative numerical calculations for the response of polycrystals with random orientation of the single crystals. In practice, however one neither has exact representation of the micro-structure, nor is it possible to solve numerically problems involving sufficient number of grains, as would happen in realistic, three-dimensional boundary value problems. It is therefore necessary to use homogenization techniques in order to obtain representative thermomechanical properties of a polycrystalline material.

These type of problems first arose in the context of homogenizing the macroscopic properties of heterogeneous composite materials. Many homogenization methods have been developed over the years for elastic and elastoplastic materials, among which averaging methods (Christensen, 1991; Hershey, 1954; Hill, 1965; Kröner, 1958; Mori and Tanaka, 1973; Nemat-Nasser and Hori, 1993) which typically consider ellipsoid inclusion in a Representative Volume Element (RVE) and asymptotic expansion methods, initially developed for periodic microstructure (Bakhvalov and Panasenko, 1990; Bensoussan et al., 1978; Gaymonat et al., 1993; Sanchez-Palencia, 1980; Zhikov et al., 1994). Most of the micromechanical models for polycrystalline SMAs are based on a self consistent type of averaging methods (Falk, 1990; Gao and Brinson, 2002; Lagoudas and Bhattacharya, 1997; Lu and Weng, 1998; Patoor et al., 1987, 1996).

Such an approach was developed for the elastoplastic behavior of heterogeneous materials by Berveiller et al. (1994). Since micromechanical modelling of SMAs will not be adopted here, the reader is referred to (Patoor et al., 2005; Roubicek, 2004) for a detailed review of the subject.

The other class of models for polycrystalline SMAs are phenomenological ones which rely on continuum thermomechanics with internal variables to account for the changes in the microstructure due to phase transformation. These type of models usually assume a macroscopic energy function that depends on state and internal variables used to describe the degree of phase transformation. Evolution equations are then postulated for the internal variables. The macroscopic energy and the evolution equations are assumed to have a certain functional form, which must be compatible with thermodynamics. The fundamental structure of all these models is very similar, and can be classified as that of constitutive models with internal state variables (Hill, 1967; Kestin and Bataille, 1978; Kestin and Rice, 1970; Rice, 1971). The resulting phenomenological models do not directly depend on material parameters at the microscopic level, but on a set of parameters at the macroscopic level which are determined by experimental observations. Such models can be very simple, for example modeling the uniaxial pseudoelastic response of an SMA by a piecewise linear function, or can be very complex, as in 3-D models, involving a number of material parameters which have to be determined by extensive experimentation and often do not have obvious physical interpretation. Phenomenological models are easily implementable in numerical methods for the solution of boundary value problems on the structural (macroscopic) level, and, depending on the application, one has the flexibility to make numerous trade-offs between accuracy and complexity. An alternative to using internal variables and defining evolution equations, as in classical plasticity, are the energy minimization methods. In one of the early examples, Ball and James

(1987) modeled the SMA as a nonlinear elastic material and postulated a two-well free energy function. By determining energy minimizing deformations with two coherent and macroscopically unstressed variants of martensite it is possible to find a microstructure which corresponds to the loading conditions. Energy minimization methods will not be discussed in this work.

Other approaches to phenomenological SMA models have also been proposed. The work of Brocca et al. (2002) presents a three-dimensional model for SMAs which is based on the microplane model by Bažant (1984). The main idea of the model is to deduce the macroscopic constitutive behavior of an SMA by describing the response of the SMA along planes of different orientations, called *microplanes*. The SMA constitutive behavior on the microplanes is described by a one-dimensional model. First, the normal and shear components of the stress on each microplane are defined in terms of the unit normal and tangential vectors of the plane and the macroscopic stress tensor. Next, the normal and shear components of the strain are calculated based on the constitutive model for the microplane. Finally, the components of the macroscopic strain tensor are calculated from the normal and shear strain components for a set of microplanes using the principle of virtual work. The particular SMA constitutive model on the microplane implemented in the work by Brocca et al. (2002) is the one presented by Bekker and Brinson (1998), however, it is noted that any other model can easily be implemented. The effect of the hydrostatic pressure and the tension/compression asymmetry are also taken into account by modifying the critical stress values for phase transformation and the transformation temperatures. Various results demonstrating the capabilities of the microplane model are presented and compared with experimental data.

In this work, the phenomenological approach is selected for the purpose of modeling complicated loading paths involving both twinned and detwinned martensite.

This is done for two reasons. First, there is an extensive volume of phenomenological models aimed at separately predicting the stress-induced martensitic transformations, and, to a lesser degree, the detwinning of self-accommodated martensite. In this respect the 3-D model proposed in the current work is a combination of these two separate regimes of SMA constitutive behavior which is exhibited often in complex thermomechanical loading paths. Secondly, the very complex microstructural changes which occur during a complicated loading path make it intrinsically difficult to use micromechanical methods. A review of phenomenological SMA models is presented next.

3. Literature review of phenomenological models

Significant effort has been devoted over the past decade to establish phenomenological constitutive models describing the macroscopic thermomechanical response of polycrystalline SMAs (Bekker and Brinson, 1997, 1998; Berveiller et al., 1991; Bo and Lagoudas, 1999a,b,c; Boyd and Lagoudas, 1994b, 1996a,b; Brinson, 1993; Graesser and Cozzarelli, 1991; Juhasz et al., 2002; Lagoudas and Bo, 1999; Lagoudas et al., 1996, 1994; Lagoudas and Entchev, 2004; Lagoudas and Shu, 1999; Leclercq and Lexcellent, 1996; Lexcellent et al., 2000; Liang and Rogers, 1990, 1992; Marketz and Fischer, 1995; Marketz et al., 1995; Ortin and Planes, 1988, 1989; Patoor et al., 1987, 1988; Raniecki and Lexcellent, 1994; Sato and Tanaka, 1988; Sun and Hwang, 1993a,b; Sun et al., 1991; Tanaka, 1986; Tanaka et al., 1992, 1995). Most of the constitutive models adopt a thermodynamic structure and select the martensitic volume fraction as an internal state variable to account, on the average, for the influence of the microstructure.

The early constitutive models (Boyd and Lagoudas, 1994b, 1996a; Brinson, 1993;

Liang and Rogers, 1990, 1992; Tanaka, 1986; Tanaka et al., 1986, 1995) have been used to derive the pseudoelastic response of SMAs and their main difference is the hardening function selected to model the stress-strain response during the stress induced martensitic phase transformation. A unified framework for these early constitutive models has been presented by Lagoudas et al. (1996). Further improvements in the accuracy of SMAs models was achieved by Lexcellent et al. (2002); Qidwai and Lagoudas (2000b); Raniecki and Lexcellent (1998), who proposed different transformation functions in order to capture the asymmetric response that SMAs exhibit in tension and compression. Qidwai and Lagoudas (2000b) also studied the consequences the principle of maximum dissipation during phase transformation has on the transformation surface and flow rules. The one-dimensional model of Brinson (1993) was one of the first to include modeling of detwinning of martensite. The work was based on a phase diagram approach and used two volume fractions of martensite to model pseudoelasticity and detwinning separately. It was further refined by Bekker and Brinson (1997, 1998). A thermodynamics based model of detwinning has been proposed by Juhasz et al. (2002); Lagoudas and Shu (1999); Leclercq and Lexcellent (1996). Reorientation of martensite during non-proportional loading has been taken into account by Boyd and Lagoudas (1994a), who used a non-associative flow rule during the reverse transformation. Juhasz et al. (2002) addressed this issue by using both the martensitic volume fraction and the transformation strain as separate internal variables. Cyclic loading and transformation induced plasticity in SMAs has also been a major research topic in SMA modeling (Bo and Lagoudas, 1999a,b,c; Lagoudas and Bo, 1999; Lagoudas and Entchev, 2004; Lexcellent and Bourbon, 1996; Lexcellent et al., 2000; Tanaka et al., 1995). In the works of Lexcellent and Bourbon (1996); Tanaka et al. (1995) it is assumed that a portion of the martensite does not recover after each cycle, which leads to observable unrecoverable strain, which even-

tually saturates with the number of cycles. In the one dimensional model of Bo and Lagoudas (1999a,b,c) and Lagoudas and Bo (1999) the stress-induced transformation is modelled by allowing both transformation and plastic strains to develop simultaneously as a result of the applied load. The work was later extended by Lagoudas and Entchev (2004) to three dimensions.

The models reviewed above are rate independent ones, having a stress-strain response dependant only on the loading path. Rate dependent constitutive models have also been proposed in the literature. An early example is the model by Achenbach (1989), who uses two-well potentials for the free energy and statistical physics to justify transition probabilities between two different variants of martensite (one in tension and one in compression) and austenite. The formulation of the model allows for a softening stress-strain relationship. In a recent paper this model is extended by Govindjee and Hall (2000), who used multi-well potentials and an arbitrary number of martensitic variants. Other authors have directly coupled a nonlinear thermoelastic potential and a kinetic relation to solve wave propagation problems in one-dimensional SMA rods. Abeyaratne and Knowles (1994a,b, 1997) have solved the Riemann problem in both isothermal and adiabatic settings for an SMA with softening behavior. A kinetic relation defining the speed of propagation of the phase front is introduced as a constitutive relation in order to enforce uniqueness of the Riemann problem.

The numerical implementation of phenomenological models has also been an active area of research (Auricchio, 2001; Auricchio et al., 1997; Govindjee and Miehe, 2001; Qidwai and Lagoudas, 2000a). While the computational methods have their roots in algorithms used in computational plasticity (cf. Ortiz and Popov (1985); Simo and Hughes (1998)), the complex behavior of SMAs requires the development of specialized algorithms.

Describing the complex characteristics involved in the phase transitions in poly-

crystalline SMAs has been a significant challenge to researchers. These include modeling the hardening during phase transformation; the asymmetric response that SMAs exhibit in tension and compression; the modeling of detwinning of martensite and, more generally, complicated thermomechanical paths beyond isobaric or isothermal ones; two-way shape memory effect; the effects of reorientation and the accumulation of plastic strains during cyclic loading. Historically, the first subject addressed by researchers was the choice of transformation hardening functions. These topics have been covered in the review paper by Birman (1997). A detailed account can also be found in Lagoudas et al. (1996) and thus will not be discussed here. The remaining topics are discussed in detail below.

3.1. Transformation surface

The topic of transformation surfaces and tension-compression asymmetry of the SMA response has been investigated by Raniecki and Lexcellent (1998) who presented a model for pseudoelasticity of SMAs. A distinct feature of the model is its capability to take into account the difference between the tension and compression loading. This is accomplished by using a $J_2 - J_3$ transformation surface. The model uses exponential hardening functions. It was used in a later work by Raniecki et al. (2001) to study bending of SMA beams undergoing pseudoelastic loading. In this particular work the tension-compression difference was neglected. The authors were able to determine the distribution of the martensitic volume fraction along the thickness of the beam during both loading and unloading. Additional results included plots of the beam curvature versus the applied moment. Rejzner et al. (2002) have further extended the work on pseudoelastic beams, by including the effect of tension-compression asymmetry in the analysis and comparing the results with experimental data. It was found, however, that the tension-compression asymmetry does not have a significant influence on the

macroscopic beam response.

The comprehensive study of Qidwai and Lagoudas (2000b) focused on the choice of different transformation functions and their effect on the material response. In particular, the asymmetry of the material behavior under tension and compression, as well as the volumetric transformation strain, can be modeled by choosing an appropriate functional form of transformation function. Qidwai and Lagoudas (2000b) proposed a transformation function, based on the J_2 , J_3 and I_1 stress invariants which can account for the observed asymmetry. The subject of the form of the transformation function has been revisited in a recent paper by Lexcellent et al. (2002). Multiaxial experiments on CuZnAl and CuAlBe polycrystalline SMAs have been performed to determine the initial transformation surface. The experiments have revealed tension-compression asymmetry, consistent with the results found in the literature. Motivated by the experimental results, Lexcellent et al. (2002) have proposed an analytical expression for the transformation function, based on the J_2 and J_3 stress invariants.

3.2. Detwinning of martensite

Another important aspect of the SMA response is the detwinning of martensite. The one-dimensional model of Brinson (1993) was one of the first to include modeling of detwinning of self-accommodated martensite. The work was based on a phase diagram approach and used two volume fractions of martensite to model pseudoelasticity and detwinning separately. It was further refined by Bekker and Brinson (1997, 1998) by incorporating different hardening functions and minor transformation loops. While an important step in the incorporation of twinned martensite into SMA modelling, these models however use a phase diagram which is inconsistent with available experimental data. As a result certain modifications are needed in order to correctly predict the

detwinning of twinned martensite for temperatures between M_s and A_s .

Leclercq and Lexcellent (1996) have presented a thermodynamics model formulated in a 3-D framework; however, only 1-D implementation and numerical results have been provided. Comparisons are made with experimental data for both pseudoelastic mechanical loading as well for isobaric thermally-induced transformation. It has been found that the results are in reasonably good agreement, with the largest discrepancies observed for the case of isobaric thermally-induced transformation. In another study, Lagoudas and Shu (1999) have proposed a 3-D model with three internal variables but again with only 1-D numerical implementation and results. The main drawback of this model is its inability to simulate repeated SME loading paths, due to the specific selection of hardening function for the detwinning of twinned martensite. Juhasz et al. (2002) proposed a thermodynamics model consistent with the phase diagram of Brinson (1993) which takes into account the detwinned martensite by using both the twinned martensitic volume fraction and the transformation strain as separate internal variables.

3.3. Cyclic loading and transformation induced plastic strain

One of the important problems recently addressed by the researchers is the behavior of SMAs under cycling loading (Abeyaratne and Kim (1997); Bo and Lagoudas (1999a,b,c); Fischer et al. (1998); Lagoudas and Entchev (2004); Lexcellent and Bourbon (1996); Lexcellent et al. (2000); Tanaka et al. (1995)). During cyclic phase transformation a substantial amount of plastic strains is accumulated. In addition, the transformation loop evolves with the number of cycles and TWSME is developed. Based on the experimental observations researchers have attempted to create models able to capture the effects of cycling loading. One-dimensional models for the behavior of SMA wires under cycling loading have been presented by Lexcellent and

Bourbon (1996); LExcellent et al. (2000); Tanaka et al. (1995) and Abeyaratne and Kim (1997), among others.

A series of papers by Bo and Lagoudas (1999a,b,c) and Lagoudas and Bo (1999) studies the cyclic behavior of SMA wires in one dimension. The work focuses on the modeling of stress-induced transformation, where both transformation and plastic strains occur simultaneously as a results of the applied stress. The resulting model is able to account for simultaneous development of transformation and plastic strains during phase transformation under applied loads. In addition to the plastic strain, the changes in the material response are also modeled by introducing evolution equations for the material parameters. Finally, minor hysteresis loops are also modeled by Bo and Lagoudas (1999c). This is accomplished by modifying the transformation criterion and the hardening parameters during a minor loop. All of the above-mentioned features of the model have been demonstrated and the results have been compared with experimental data for NiTi SMA wires. The results have been found to be in very good agreement.

A three-dimensional model for transformation induced plasticity has been presented by Fischer et al. (1998). In contrast to the work by Bo and Lagoudas (1999b), separate phase transformation condition and plasticity yield condition are used by Fischer et al. (1998). The theory is presented in general terms, but the identification of the material parameters and the implementation are not discussed.

4. Outline of current research

The research in this thesis consists of four separate topics listed below.

4.1. Development of an adaptive finite element method for wave propagation problems in SMAs

Shape Memory Alloys have recently been considered for dynamic loading applications for energy absorbing and vibration damping devices. An SMA body subjected to external dynamic loading will experience large inelastic deformations that will propagate through the body as phase transformation and/or detwinning shock waves. The wave propagation problem in a cylindrical polycrystalline SMA rod induced by an impact loading is considered in Chapter II. The constitutive model of Boyd and Lagoudas (1994b) is used. The model, originally designed for pseudoelasticity, is reinterpreted to model detwinning of SMAs. Numerical solutions for various boundary conditions are presented for stress induced martensite and detwinning of martensite. The numerical simulations utilize an adaptive Finite Element Method (FEM) based on the Zienkiewicz-Zhu (ZZ) error estimator. Selected results are compared to known analytical solutions to verify the adaptive FEM approach. The energy dissipation in an SMA rod is evaluated for a square pulse stress input applied at various temperatures involving both stress induced martensite and detwinning of martensite. Comparisons with available experimental data is also given.

The reinterpretation of the constitutive model (Boyd and Lagoudas, 1994b) used for the one-dimensional wave propagation problem cannot be generalized to cases where significant changes in the temperature occur and the SMA exhibits detwinning and pseudoelasticity, either simultaneously or at different instances of time. While the numerical simulations were successfully applied to a range of wave propagation problems, many practical problems require a comprehensive SMA model, capable of accounting for all phase transitions, both simultaneously and sequentially.

4.2. Development of a three phase, 3-D constitutive model for SMAs

The analysis of the existing models and their comparison to the experimental results has shown that current SMA constitutive models can handle successfully different types of thermomechanical loading paths but have difficulties doing so in a unified manner. While the models which take into account the development of stress-induced martensite have reached a high level of sophistication, generally they lack the ability to handle other loading paths, involving detwinning and reorientation of martensite. Some of the models presented in the literature address the above problem but in a limited way, usually restricted to a one-dimensional description and/or cannot handle certain classes of thermomechanical loading paths (Bekker and Brinson, 1997, 1998; Brinson, 1993; Lagoudas and Shu, 1999).

The current applications of SMAs in multifunctional smart structures have matured beyond the simple, one-dimensional actuators. For a comprehensive review of early SMA based devices, the reader is referred to Birman (1997); Funakubo (1987). Recent application designs, however, involve complex SMA structures undergoing non-proportional thermomechanical loading paths. Devices such as SMA micro-grips (Kohl et al., 2002) or self-expanding medical stents (Jung et al., 2004), to name a few, have complex geometry and undergo complicated thermal and mechanical paths. In addition to the common and well studied development of stress-induced martensite, parts of the thermomechanical loading process often involve generation of twinned martensite, detwinning of twinned martensite and various simultaneous transformation of twinned, detwinned martensite and austenite. Thus, in order to further advance existing applications and promote the development of new ones, it is necessary to have a comprehensive three-dimensional model for SMAs, which can handle a wide variety of loading paths and reliably predict the thermomechanical response of SMA

actuators and devices.

In this work (Chapter III), a thermodynamics based model with three internal variables is formulated for the simultaneous modeling of pseudoelasticity and detwinning of self-accommodated martensite associated with the SME. The model is formulated in three dimensions and is made consistent with a uniaxial phase diagram in stress-temperature space. The phase diagram incorporates new experimental results, presented in Section 1, which demonstrate that twinned and detwinned martensite transform to austenite at different temperatures. The phase diagram is constructed in Section 2 based on these new observations, as well as, on a careful reexamination of published experimental data on detwinning of twinned martensite and the conversion of twinned martensite to austenite. This modified phase diagram is important in deriving a robust model which can correctly predict the material behavior for a wide range of paths. Specifically, paths which generate a mixture of the austenitic, twinned martensitic and detwinned martensitic phases and may cycle repeatedly between different regions of the phase diagram. The model is utilized to simulate two uniaxial problems which illustrate its capabilities over a wide temperature/stress range.

4.3. Numerical implementation of SMA constitutive model

The numerical implementation of the constitutive model is developed and presented in Chapter IV. The implementation is designed for displacement based Finite Element Methods (FEM). The approach of Qidwai and Lagoudas (2000a), based on return mapping algorithms (Ortiz and Popov, 1985; Ortiz and Simo, 1986; Simo and Hughes, 1998) and specifically designed for pseudoelastic SMA materials, has been generalized to multi-surface inelasticity and multiple simultaneous transformations. Appropriate algorithmic tangent moduli are derived so that Newton-type iteration methods can be utilized to solve the nonlinear system of algebraic equations, arising from the FEM

discretization. The implementation is coded in the C++ programming language in order to facilitate integration with modern FE analysis software.

Several nontrivial model problems are also solved. Two of them, one for a perforated square in plane strain (Chapter IV, Section 2.1) and one for a 3D structural member (Chapter IV, Section 2.2) are simulated. The boundary conditions lead the material through a complicated loading path starting from low temperature and going through $M^t \rightarrow M^d$, $M^t \rightarrow A$, $M^d \rightarrow A$ in both successive and simultaneous manner, thus testing extensively the 3-D numerical implementation of constitutive model.

The third model problem is a coupled fluid-structure interaction problem with a Stokes fluid flowing through a channel with an SMA membrane (Chapter V, Section 4). It is performed in conjunction with a Fluid-Structure Interaction (FSI) solve, developed in Chapter V. The example is designed to test the feasibility of temperature actuated flow regulating devices.

4.4. Investigation of fluid-structure interaction problems with application to SMA based flow regulating devices

The current work also involves modeling of fluid-structure interaction problems. The effort was motivated by the recent advent of Porous SMAs which have various potential applications such as bone replacements, filters in the automotive and chemical industries, and light-weight structures for aerospace applications. The introduction of a fluid passing through an open pore SMA can be used to significantly extend the functionality of such devices. In order to investigate these possibilities a Fluid-Structure Interaction (FSI) numerical solver was developed for the fully coupled FSI problem (Chapter V). The FSI problem is formulated for Stokes flow and the solid can be either linear or include material nonlinearities. The solver utilizes mesh regeneration schemes to support large displacements of the solid phase.

The numerical method was implemented and tested on a range of numerical examples (Chapter V, Section 3). It was further used to verify an existing asymptotic solution to the FSI problem in a simple channel geometry. As a precursor to porous based SMA flow regulating devices, the FSI solver was used to simulate a simple temperature regulated flow device (Chapter V, Section 4). The device consist of a SMA membrane embedded in a rigid channel. The simulation involves a temperature sweep to determine the flow rate at different temperatures. The model developed in Chapter III was especially suited for simulating such devices since it captures correctly the phenomena occurring in SMAs over a wide temperature range.

CHAPTER II

NONLINEAR WAVE PROPAGATION IN SMA RODS

Shape Memory Alloys (SMAs) have recently been considered for dynamic loading applications as energy absorbing and vibration damping devices. An SMA body subjected to external dynamic loading will experience large inelastic deformations that will propagate through the body as phase transformation and/or detwinning shock waves. There are many areas of applications which can successfully utilize the unique properties of SMAs. The engineering research presented in this chapter relates directly to the design of SMA components capable of absorbing dynamic loads. Such components can be integrated into critical parts of structures that may need protection from impact loads. Examples include joints that connect the hull of an underwater vehicle with its internal structure, tank armor or blast resistant cargo containers. Another promising field of application includes various active or passive vibration damping devices. Many different SMA devices have been proposed among which nonlinear hysteretic SMA springs (Graesser, 1995; Yiu and Regelbrugge, 1995), wires (Fosdick and Ketema, 1998; Thomson et al., 1995) or rods (Feng and Li, 1996). In a recent series of papers (Khan et al., 2004; Lagoudas et al., 2004) the authors investigate numerically the vibration damping capabilities of SMAs.

The main focus of this chapter is the study of the one-dimensional dynamic problem of loading an SMA rod under conditions of pseudoelasticity and detwinning. Both the pseudoelastic response and detwinning of self-accommodated martensite in classical SMAs is characterized by strictly monotonous stress-strain curves. As a result the field equation describing the impact of an SMA rod (Section 1) form a second order, nonlinear, strictly hyperbolic problem. It is a well-known fact that solutions to nonlinear hyperbolic problems possess self-inducing discontinuities, that

is, given an arbitrarily smooth initial data, after some finite time, the solution may become discontinuous in space (see, for example Godlewsky and Raviart, 1996; Renardy and Rogers, 1996). The speed of these shocks is dependent on the strain level and shocks travelling at different speed may further interact. The problem therefore is very complicated and few wave-propagation results in SMA materials are known.

Classical rate-independent plasticity theory is not sufficient to describe the behavior of SMA materials. While it is still capable of partially predicting the shape memory effect (without capturing the strain recovery upon heating), it cannot model the pseudoelastic response. However, for rate independent models of SMAs both theoretical and experimental developments of dynamic elasto-plasticity can be used for guidance. Theoretical developments on elasto-plastic wave propagation in long rods dates back to the works of Rakhmatulin (1945); Taylor (1958); Von Karman and Duwez (1950). Extensive experiments on elasto-plastic wave propagation have been carried out by Bell (1962); Bodner and Clifton (1967); Chiddister and Malvern (1963); Clifton and Bodner (1966); Kolsky (1949) using a split-Hopkinson bar apparatus. The split bar technique itself was introduced by Kolsky (1949). The reader is referred to classical texts on wave propagation such as Kolsky (1963) and Graff (1975) for additional information.

There has been a limited amount of experimental work done on characterizing the dynamic response of SMAs. An experimental study on the propagation of shear waves in single crystal Cu-Al-Ni shape memory alloy has been done by Escobar and Clifton (1993). Phase transition shocks are not observed directly due to their low propagation speed. Instead, their presence is inferred from the measurements of the elastic waves at the rear end of the specimen. An analytical attempt to model these experiments is presented in Abeyaratne and Knowles (1997).

In a recent paper (Chen and Lagoudas, 2000) the rate independent model for

polycrystalline SMAs (Lagoudas et al., 1996) is employed to obtain solutions to the coupled thermomechanical problem for SMA materials. The authors take into account the latent heat generation and assuming adiabatic conditions they solve the problem by the method of characteristics together with jump conditions that yield unique solutions. A similar study (Bekker et al., 2002), but for different constitutive models has been carried out for both isothermal and adiabatic conditions. In a different setting Oberaigner et al. (1996) investigates numerically the coupled problem of wave propagation and heat transfer in an SMA rod. The authors focus on stress pulses of low magnitude that cause only elastic deformations. The temperature at one end of the SMA rod is chosen as a function of time in such a way as to utilize the phase change due to the shape memory effect in order to maximize the damping characteristics of the rod.

The dynamics of phase transformation in piecewise linear elastic materials with non-monotone hysteresis is also studied by Abeyaratne and Knowles (1991). A unique solution is obtained with the use of a kinetic relation controlling the rate of the phase transformation together with a nucleation condition for the initiation of the transformation. In later work the same authors extend the analysis to account for thermal effects (Abeyaratne and Knowles, 1994a,b). In a general setting Pence (1986) considers wave propagating in a nonlinear elastic bar with a non-monotonic stress-strain relationship subjected to a monotonically increasing load. It is found that for sufficiently high loads a strain discontinuity associated with phase transformation is being created.

The complex nature of most constitutive models for SMA materials makes direct integration of even the simplest uniaxial transient initial boundary value problems (IBVP) very complicated. Closed form solutions can usually be obtained for simple boundary conditions, e.g. impact step loading (Chen and Lagoudas, 2000) or by

simplifying the constitutive model so that the stress can be obtained as an explicit function of strain (Bekker et al., 2002). Numerical solutions of the impact loading problem have been obtained by (Bekker et al., 2002; Jimenez-Victory, 1999) by mainly using the Lax-Friedrichs finite difference scheme. This FD scheme has been found to produce a considerable amount of numerical dissipation which makes the distinction between a self-contained nonlinear shock and a rarefaction wave difficult. In this chapter numerical simulations of step and pulse shock loading both for stress induced phase transformation and detwinning of martensite are performed using the FEM method. An adaptive meshing technique based on the ZZ error estimator (Zienkiewicz and Zhu, 1987) is utilized in order to improve the accuracy of the method and decrease computational time. Comparisons with analytical solutions are made whenever such solutions are available. Based on the simulation results, the energy dissipation of SMA rods for pulse loads are discussed. Finally recent experimental data (Lagoudas et al., 2003) on impact induced detwinning in a polycrystalline NiTi SMA rod is analyzed and compared with numerical simulations.

Summarized below are the key points addressed in this chapter:

- An adaptive Finite Element Method (FEM) based on the Zienkiewicz-Zhu (ZZ) error estimator is developed and implemented for wave propagation problems in SMA rods. Numerical solutions for various boundary conditions are presented for stress induced martensite and detwinning of martensite. Selected results are compared to known analytical solutions to verify the adaptive FEM approach.
- A one-dimensional adaptation of the constitutive model of (Boyd and Lagoudas, 1994b; Lagoudas et al., 1996) is chosen, assuming rate independence in the constitutive thermomechanical response of SMAs.
- The energy dissipation in an SMA rod is evaluated for a square pulse stress

input applied at various temperatures involving both stress induced martensite and detwinning of martensite.

- Actual experimental observations from a split Hopkinson bar test on an SMA rod in detwinning conditions are analyzed. Due to experimental limitations the stress levels in the bar could not reach high enough levels to cause pseudoelasticity, so wave propagation data only from detwinning of martensite was available.
- The quasi-static and dynamic stress-strain hysteretic response of the SMA, both due to detwinning, are found to be nearly identical, validating the assumption of rate independence.
- Strain history records obtained by strain gauges placed at different locations along the SMA rod are compared with numerical simulations for a square pulse stress input. The quasi-static tests are used to calibrate the rate independent constitutive model. The numerical simulations are found to match the experimental observations reasonably well.

As it was mentioned above, the rate independent constitutive model of Boyd and Lagoudas (1994b); Lagoudas et al. (1996) is used. This model was originally developed for purely pseudoelastic response of SMAs. Dictated by actual experimental constraints, it is shown that the model can be adapted to simulate detwinning of SMAs by reinterpreting the material constants and using a high-order polynomial hardening function. This made possible the simulations of the available experimental data. Such an approach however, is applicable only in cases when the entire loading history of the SMA component is confined to detwinning phenomena. If, for example, due to energy dissipation sufficient amount of heat is released during the impact inside

the material, or the temperature is changed through the boundary conditions then the material may undergoes pseudoelastic and detwinning deformations during the duration of the loading. A model capable of handling both phenomena is therefore required to simulate the dynamic response under more general conditions. The work presented in this chapter thus serves as a real-life motivation for the comprehensive SMA model presented in Chapter III.

The chapter continues with Section 1 which is devoted to a brief overview of the field equations, boundary conditions and constitutive model defining the problem. The implementation of the FEM for the NiTi SMA is outlined in Section 2.1. The adaptive strategy is presented in Section 2.2. In order to verify the implementation of the adaptive FEM a boundary value problem with a step-function stress boundary condition is solved in Section 3.1. This specific boundary condition allows for the construction of analytical solutions which can be used to verify the numerical solution methodology. Then, a square pulse IBVP is solved for conditions of stress induced martensite (Section 3.2) and detwinning (Section 3.3). Expected values for energy dissipation as the pulse propagates through the rod are presented. Section 4 describes the split-Hopkinson bar experiment and discusses the dynamic characterization of SMA materials. Finally, in Section 4.2 the numerical schemes developed in this chapter are utilized to simulate available experimental results.

1. Field equations and constitutive model for the impact problem of SMA rods

A cylindrical SMA rod of uniform cross-section and length L is considered. A coordinate cover is associated with the centroidal axis of the rod spanning the interval $0 \leq x \leq L$. The rod which is initially stress free and at rest is subjected to an impact load at its left end ($x = 0$). The right end ($x = L$) is assumed to remain traction

free. The field equations, initial and boundary conditions are presented next followed by a description of the thermomechanical constitutive model for SMAs.

1.1. Field equations, initial and boundary conditions

The rod is assumed to be long compared to its diameter so it is under uniaxial stress state and the stress $\sigma(x, t)$ depends only on the axial position and time. The axial component of the displacement is denoted by $u(x, t)$. Linearized strain is further assumed so the axial component of the strain $\varepsilon(x, t)$ is related to the displacement by $\varepsilon(x, t) = \partial u / \partial x$. Finally, the density of the material ρ is assumed constant. The local form of the balance of linear momentum and energy then read (Graff, 1975; Malvern, 1969):

$$\rho \frac{\partial^2 u}{\partial t^2} = \frac{\partial \sigma}{\partial x}, \quad (2.1)$$

$$\rho \frac{\partial}{\partial t} \left(\mathcal{U} + \frac{1}{2} \left(\frac{\partial u}{\partial t} \right)^2 \right) = \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial t} \sigma - q \right), \quad (2.2)$$

where \mathcal{U} is the internal energy per unit mass and $q(x, t)$ is the heat flux.

The timescale of the impact problem is on the order of micro- to milliseconds. The physically meaningful IBVP is an adiabatic one because such time-intervals are too short for heat conduction to take place as well as for convection to remove heat through the surface of the rod. In the *adiabatic* approximation, therefore, the heat conduction term q in (2.2) can be neglected so the balance of energy in conjunction with (2.1) yields

$$\rho \frac{\partial \mathcal{U}}{\partial t} = \sigma \frac{\partial^2 u}{\partial x \partial t}. \quad (2.3)$$

Equation (2.1) and (2.3) involve the field variables u , σ and \mathcal{U} . Through appropriate constitutive assumptions to be discussed in the following section only $u(x, t)$ and the temperature $T(x, t)$ will become the independent variables.

For the field variables the following initial and boundary conditions are assumed:

$$u|_{t=0} = 0, \quad \frac{\partial u}{\partial t}|_{t=0} = 0, \quad T|_{t=0} = T_R, \quad (2.4)$$

$$\sigma|_{x=0} = \sigma_0(t), \quad \sigma|_{x=L} = 0, \quad (2.5)$$

The initial conditions indicate that the rod is at rest and its temperature is equal to the ambient temperature T_R . The boundary conditions specify the traction $\sigma_0(t)$ applied¹ to the left end of the rod. The right end is kept traction free.

1.2. Uniaxial thermomechanical constitutive model for polycrystalline SMAs

The field equations (2.1), (2.3) and initial and boundary conditions (2.4), (2.5) alone are not sufficient to form a complete IBVP. A thermomechanical constitutive model that captures the key characteristics of pseudoelasticity and detwinning of the SMA response is needed.

1.2.1. Stress induced martensite

The constitutive model used is formulated in terms of the Gibbs free energy G and employs the volume fraction of detwinned martensite ξ formed from austenite as an internal variable (Lagoudas et al., 1996). The specific form of G in the one dimensional case is:

$$\begin{aligned} G = G(\sigma, T, \xi) = & -\frac{1}{2\rho}\mathcal{S}\sigma^2 - \frac{1}{\rho}\sigma(\alpha(T - T_R) + \varepsilon^t) \\ & + c\left((T - T_R) - T \ln\left(\frac{T}{T_R}\right)\right) - s_0T + u_0 + f(\xi), \end{aligned} \quad (2.6)$$

and it is linked to the internal energy \mathcal{U} by a Legendre transformation:

$$\mathcal{U} = G + Ts + \frac{1}{\rho}\sigma\varepsilon. \quad (2.7)$$

¹There is no continuity requirement on $\sigma_0(t)$ i.e. impact loads are allowed

The definition of G includes the inelastic transformation strain ε^t associated with the phase transformation. The function $f(\xi)$ is taken to be a quadratic polynomial in ξ and is responsible for the transformation hardening:

$$f(\xi) = \begin{cases} \frac{1}{2}\rho b^M \xi^2 + (\mu_1 + \mu_2)\xi, & \dot{\xi} > 0 \\ \frac{1}{2}\rho b^A \xi^2 + (\mu_1 - \mu_2)\xi, & \dot{\xi} < 0 \end{cases}, \quad (2.8)$$

where material constants ρb^A , ρb^M , μ_1 and μ_2 define the transformation surfaces and the hardening during the forward and reverse transitions (Qidwai and Lagoudas, 2000b). In the above $\dot{\xi} > 0$ denotes the forward transformation and $\dot{\xi} < 0$ the reverse. The remaining material properties in (2.6) are the effective compliance \mathcal{S} , effective thermal expansion coefficient α , effective specific heat c , effective specific entropy at the reference state s_0 and effective specific internal energy at the reference state u_0 for the SMAs which is composed of a mixture of austenite and martensite. They are approximated by the following averaging expressions, which are good approximations for polycrystalline SMAs with random orientation distributions of the martensitic grains (Boyd and Lagoudas, 1994b):

$$\begin{aligned} \mathcal{S} &= \mathcal{S}(\xi) = \mathcal{S}^A + \xi \Delta \mathcal{S}, & \Delta \mathcal{S} &:= \mathcal{S}^M - \mathcal{S}^A, \\ \alpha &= \alpha(\xi) = \alpha^A + \xi \Delta \alpha, & \Delta \alpha &:= \alpha^M - \alpha^A, \\ c &= c(\xi) = c^A + \xi \Delta c, & \Delta c &:= c^M - c^A, \\ s_0 &= s_0(\xi) = s_0^A + \xi \Delta s_0, & \Delta s_0 &:= s_0^M - s_0^A, \\ u_0 &= u_0(\xi) = u_0^A + \xi \Delta u_0, & \Delta u_0 &:= u_0^M - u_0^A. \end{aligned} \quad (2.9)$$

Quantities with subscript A denote the appropriate material constant for the austenite phase and those with subscript M for the martensite phase. Following a standard

thermodynamic procedure the following constitutive relations are obtained:

$$s = -\frac{\partial G}{\partial T}, \quad (2.10)$$

$$\varepsilon = -\rho \frac{\partial G}{\partial \sigma}, \quad (2.11)$$

$$\pi = -\rho \frac{\partial G}{\partial \xi}, \quad (2.12)$$

where s is the entropy and π is the driving force for the transformation. Using (2.11) the following constitutive relation is obtained:

$$\sigma = E(\xi)(\varepsilon - \alpha(\xi)(T - T_R) - \varepsilon^t), \quad (2.13)$$

where $E(\xi) = 1/\mathcal{S}(\xi)$ is the effective elastic modulus. The evolution of the inelastic variable ξ is given by a consistency condition derived from a transformation criterion (Lagoudas et al., 1996). The evolution of ε^t follows that of ξ and in the one dimensional case can be integrated explicitly to yield:

$$\varepsilon^t = H \operatorname{sgn}(\sigma) \xi. \quad (2.14)$$

Here H is a positive material constant corresponding to the maximum transformation strain. The principle of maximum transformation dissipation in conjunction with the second law of thermodynamics leads to the following transformation surface:

$$\pi = \pm Y^*, \quad (2.15)$$

where $Y^* = -\frac{1}{2}\rho\Delta s_0(A_f - M_s) - \frac{1}{4}\rho\Delta s_0(M_s - M_f - A_f + A_s)$. The $+Y^*$ at the right hand side stands for the forward ($A \rightarrow M$) transformation surface and $-Y^*$ for the reverse ($M \rightarrow A$) transformation surface.

For detailed description of the transformation rule and conditions for the forward and reverse phase transformation the reader is referred to the original paper by

Lagoudas et al. (1996). The next section describes how detwinning is incorporated into the constitutive model.

1.3. Detwinning of martensite

The detwinning deformation will be accounted for by adapting the constitutive model. The material constants for twinned and detwinned martensite are the same. Consequently, the initial response and the response after the completion of detwinning will both be elastic with the slope being the modulus of elasticity of martensite E^M . The deformation is irreversible upon unloading which, consequently, will also be elastic.

The material constants in the constitutive model can be reinterpreted, replacing the ones for the austenitic phase with the ones for martensite. This will ensure the same elastic response prior to the onset of detwinning and after its completion. The internal variable ξ should be interpreted as the volume fraction of detwinned with respect to self-accommodated martensite and H is the maximum inelastic strain. From equation (2.15) the transformation surface will have the following simple form:

$$\sigma H - \frac{\partial f}{\partial \xi} = 0. \quad (2.16)$$

The hardening function in this case may be expressed as follows:

$$f(\xi) = \frac{1}{2} \rho b^d \xi^2 + Y^d \xi, \text{ for } \dot{\xi} > 0, \quad (2.17)$$

where $Y^d = \sigma_s H$ and $\rho b^d = \sigma_f H - Y^d$. For convenience, the critical stress level σ_s for the onset and σ_f for the completion of the detwinning deformation are introduced as material constants. Note that for the detwinning case $\dot{\xi}$ can only be positive since the unloading is entirely elastic. This adaptation of the model allows for the modelling of detwinning deformations when no stress induced martensite is being produced.

1.4. Isentropic approximation

The adiabatic heat equation can be simplified in order to facilitate the numerical treatment of the impact problem. Using the Legendre transformation (2.7) the internal energy can be eliminated from equation (2.3):

$$\rho T \frac{\partial s}{\partial t} = \pi \frac{\partial \xi}{\partial t}. \quad (2.18)$$

Further, upon combining (2.6) and (2.10) an explicit expression for the entropy is obtained

$$s = \alpha \sigma / \rho + c \ln(T/T_R) + \Delta s_0 \xi + s_0^A. \quad (2.19)$$

On substituting (2.19) into (2.18) the balance of energy becomes:

$$\rho c \frac{\partial T}{\partial t} = -T \frac{\partial}{\partial t} (\alpha \sigma + \rho \Delta s_0 \xi) + \pi \frac{\partial \xi}{\partial t}. \quad (2.20)$$

According to Cory and McNichols (1985); McNichols (1987) $\pi \ll \rho \Delta s_0 T$ for most SMAs. For NiTi the precise values yield $\pi / \rho \Delta s_0 T \ll 0.013$ so equation (2.20) can be approximated by

$$\rho c \frac{\partial T}{\partial t} = -T \frac{\partial}{\partial t} (\alpha \sigma + \rho \Delta s_0 \xi), \quad (2.21)$$

which is equivalent to the isentropic condition $\frac{\partial s}{\partial t} = 0$. The heat capacity c can be assumed constant for the two phases (i.e. $c^A = c^M$). Then equation (2.21) can be integrated directly, yielding:

$$T = T_R e^{-\frac{1}{\rho c} (\alpha(\xi)\sigma + \rho \Delta s_0 \xi)}. \quad (2.22)$$

Consequently, the differential equation (2.3) is replaced by the algebraic equation (2.22). The impact problem then reduces to solving the balance of linear momentum (2.1) for the only field variable $u(x, t)$. The remaining field variables σ and T are coupled with the strain ε and the internal variable of the constitutive model ξ by

equations (2.13) and (2.22).

1.5. Tangent moduli

A nonlinear displacement-based FEM solver utilizing the Newton-Raphson iteration to resolve the nonlinearity requires partial derivatives of the stress with respect to an increment of the strain. An increment in the strain causes increments in both stress (equation (2.13)) and temperature (equation (2.22)):

$$\frac{d\sigma}{d\varepsilon} = \frac{\partial\sigma}{\partial\varepsilon} + \frac{\partial\sigma}{\partial T} \frac{dT}{d\varepsilon}. \quad (2.23)$$

In order to find the total derivative $\frac{d\sigma}{d\varepsilon}$ a closed form expression for $\frac{\partial T}{\partial\varepsilon}$ is needed. This is done by differentiating equations (2.13) and (2.22) with respect to the strain and combining the result to obtain:

$$\frac{dT}{d\varepsilon} = - \left(\alpha \frac{\partial\sigma}{\partial\varepsilon} + (\sigma\Delta\alpha + \rho\Delta s_0) \frac{\partial\xi}{\partial\varepsilon} \right) / \left(\frac{\rho c}{T} + \alpha \frac{\partial\sigma}{\partial T} + (\sigma\Delta\alpha + \rho\Delta s_0) \frac{\partial\xi}{\partial T} \right). \quad (2.24)$$

Second order approximations for the partial derivatives $\frac{\partial\sigma}{\partial\varepsilon}$, $\frac{\partial\sigma}{\partial T}$, $\frac{\partial\xi}{\partial\varepsilon}$ and $\frac{\partial\xi}{\partial T}$ are developed in (Qidwai and Lagoudas, 2000a) and thus all the quantities in (2.23) can be computed numerically.

2. Numerical implementation

The numerical techniques used to implement the constitutive laws are described first. For given strain increment $\Delta\varepsilon$ and temperature increment ΔT the stress σ given by equation (2.13) is computed with the help of the cutting plane return-mapping algorithm described in (Qidwai and Lagoudas, 2000a). A displacement based FEM provides strain increments. In the impact problem both stress and temperature de-

pend on the strain increment $\Delta\varepsilon$, that is for given strain both (2.13) and (2.22) have to be satisfied simultaneously. This is done via an iterative process.

The process starts with given values $\varepsilon^{(0)}$, $\sigma^{(0)}$, $T^{(0)}$ for strain, stress and temperature which satisfy (2.13) and (2.22). Given a strain increment $\Delta\varepsilon$ the pair (σ, T) corresponding to strain $\varepsilon = \varepsilon^{(0)} + \Delta\varepsilon$ is found through the iteration:

$$\sigma^{(n+1)} = E \left(\varepsilon - \alpha (T^{(n)} - T_R) - \varepsilon^t(n) \right), \quad (2.25)$$

$$T^{(n+1)} = T_R e^{-\frac{1}{\rho C} (\alpha \sigma^{(n+1)} + \rho \Delta s_0 \xi^{(n+1)})}. \quad (2.26)$$

The first equation (2.25) uses the return-mapping algorithm to compute a new value $\sigma^{(n+1)}$ for the stress based on the old temperature $T^{(n)}$. The second equation (2.26) attempts to enforce the isentropic heat equation by computing a corrected temperature $T^{(n+1)}$. The process is terminated when there is no further progress, i.e. when $|\sigma^{(n+1)} - \sigma^{(n)}|$ and $|T^{(n+1)} - T^{(n)}|$ both become smaller than certain tolerance. The algorithm showed linear convergence in the test cases, however a detailed theoretical study is required to establish its properties.

2.1. FEM procedure

A standard semi-discrete Galerkin approximation is used to generate the weak form of the problem. In this chapter only linear elements will be used. Let $P^1([0, L]) \subset H^1([0, L])$ be the set of piecewise linear functions over each element and $\{\psi_i\}_{i=1}^N$ be the usual basis of $P^1([0, L])$. The weak form of (2.1) then reads:

Find $u^h(x, t) = \sum_{i=1}^N U_i(t) \psi_i(x)$ such that for $\forall v^h \in P^1([0, L])$:

$$\rho \int_0^L \frac{\partial^2 u^h}{\partial t^2} v^h dx + \int_0^L \sigma \frac{\partial v^h}{\partial x} dx = -\sigma v^h \Big|_{x=0}. \quad (2.27)$$

As usual the number of nodes is N (i.e. $N - 1$ elements) and the nodal values for the displacement are denoted by $U_i(t)$. Whenever appropriate, vector notation will be

used, that is $\mathbf{U} = (U_1, \dots, U_N)^t$. Problem (2.27) is reduced to a second order nonlinear system of ODEs (cf., e.g., Reddy, 1993):

$$\mathbf{M}\ddot{\mathbf{U}} = \mathbf{F}(\mathbf{U}), \quad (2.28)$$

where \mathbf{M} is the mass matrix and $\mathbf{F}_{\xi(t)}(\mathbf{U})$ is the forcing term. The subscript $\xi(t)$ stands to indicate that $\mathbf{F}_{\xi(t)}(\mathbf{U})$ does not depend on the displacement only but on the whole loading history. However, for any given loading history the stress and hence $\mathbf{F}_{\xi(t)}(\mathbf{U})$ can be viewed as well defined single valued functions. Thus, without loss of generality the subscript $\xi(t)$ will be dropped in the discussion that follows. The mass matrix and load vector are given by:

$$M_{ij} = \rho \int_0^L \psi_i \psi_j dx, \quad (2.29)$$

$$F_i(\mathbf{U}) = - \int_0^L \sigma \frac{\partial \psi_i}{\partial x} dx. \quad (2.30)$$

It is also useful to introduce the forcing term $\tilde{\mathbf{F}}(\mathbf{U})$ due to inelastic strains and the stiffness matrix $\mathbf{K}(\mathbf{U})$ which are given by²:

$$K_{ij}(\mathbf{U}) = \int_0^L E(\xi) \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} dx, \quad (2.31)$$

$$\tilde{F}_i(\mathbf{U}) = \int_0^L E(\xi) [\varepsilon^t(\xi) + \alpha(\xi)(T - T_R)] \frac{\partial \psi_i}{\partial x} dx. \quad (2.32)$$

Note that the decomposition $\mathbf{F}(\mathbf{U}) = \tilde{\mathbf{F}}(\mathbf{U}) - \mathbf{K}(\mathbf{U})\mathbf{U}$ holds and (2.28) can be rewritten as:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{K}(\mathbf{U})\mathbf{U} = \tilde{\mathbf{F}}(\mathbf{U}). \quad (2.33)$$

The time integration in (2.28) (or (2.33)) is done by the backward difference method, a member of the Newmark family (Newmark, 1959; Reddy, 1993). For

²Similarly, a more precise notation for \mathbf{K} and $\tilde{\mathbf{F}}$ would be $\mathbf{K}_{\xi(t)}(\mathbf{U})$ and $\tilde{\mathbf{F}}_{\xi(t)}(\mathbf{U})$, respectively.

$t = t_s$ the Newmark scheme is defined by³:

$$\mathbf{U}_{s+1} = \mathbf{U}_s + \tau \dot{\mathbf{U}}_s + \frac{1}{2} \tau^2 \left((1 - \gamma) \ddot{\mathbf{U}}_s + \gamma \ddot{\mathbf{U}}_{s+1} \right), \quad (2.34)$$

$$\dot{\mathbf{U}}_{s+1} = \dot{\mathbf{U}}_s + \tau \left((1 - \alpha) \ddot{\mathbf{U}}_s + \alpha \ddot{\mathbf{U}}_{s+1} \right). \quad (2.35)$$

The backward difference method is obtained by setting $\alpha = \frac{3}{2}$ and $\gamma = 2$. It is easy to show (see e.g. (Reddy, 1993)) that the above difference equations lead to the following system of nonlinear algebraic equations for U_{s+1} :

$$\frac{2}{\gamma \tau^2} \mathbf{M} \mathbf{U}_{s+1} = \mathbf{F}(\mathbf{U}_{s+1}) + \mathbf{G}_s, \quad (2.36)$$

or, equivalently, to

$$\left(\frac{2}{\gamma \tau^2} \mathbf{M} + \mathbf{K}(\mathbf{U}_{s+1}) \right) \mathbf{U}_{s+1} = \tilde{\mathbf{F}}(\mathbf{U}_{s+1}) + \mathbf{G}_s, \quad (2.37)$$

where $\mathbf{G}_s = \mathbf{M} \left(\frac{2}{\gamma \tau^2} \mathbf{U}_s + \frac{2}{\gamma \tau} \dot{\mathbf{U}}_s + \frac{1-\gamma}{\gamma} \ddot{\mathbf{U}}_s \right)$. The nonlinear problem (2.36) is solved by linearizing the right-hand side

$$F_i(\mathbf{U} + \Delta \mathbf{U}) \simeq F_i(\mathbf{U}) + \sum_{j=1}^N \frac{\partial F_i(\mathbf{U})}{\partial U_j} \Delta U_j$$

and using the chain rule to obtain:

$$L_{ij}(\mathbf{U}) := \frac{\partial F_i(\mathbf{U})}{\partial U_j} = \int_0^L \frac{\partial \sigma}{\partial U_j} \frac{\partial \psi_i}{\partial x} dx = \int_0^L \frac{d\sigma}{d\varepsilon} \frac{\partial \psi_j}{\partial x} \frac{\partial \psi_i}{\partial x} dx. \quad (2.38)$$

The solution \mathbf{U}_{s+1} is found through a Newton-Raphson iterative process. Set the initial guess to $\mathbf{U}_{s+1}^{(0)} = \mathbf{U}_s$ and for $n = 1, 2, \dots$ until convergence compute:

$$\mathbf{U}_{s+1}^{(n+1)} = \left(\frac{2}{\gamma \tau^2} \mathbf{M} - \mathbf{L}(\mathbf{U}_{s+1}^{(n)}) \right)^{-1} \left(\mathbf{F}(\mathbf{U}_{s+1}^{(n)}) - \mathbf{L}(\mathbf{U}_{s+1}^{(n)}) \mathbf{U}_{s+1}^{(n)} + \mathbf{G}_s \right). \quad (2.39)$$

The cutting plane method (Qidwai and Lagoudas, 2000a) which is used to re-

³The usual notation $\mathbf{U}_s := \mathbf{U}(t_s)$ is used

solve the nonlinear behavior of the material also provides second order numerical approximation for the derivative $d\sigma/d\varepsilon$ which results in a quasi-Newton algorithm. Since the Newton algorithm is only locally convergent in the cases when it diverges the simple iteration was applied to (2.37). Again, set $\mathbf{U}_{s+1}^{(0)} = \mathbf{U}_s$ and for $n = 1, 2, \dots$ until convergence compute:

$$\mathbf{U}_{s+1}^{(n+1)} = \left(\frac{2}{\gamma\tau^2} \mathbf{M} + \mathbf{K}(\mathbf{U}_{s+1}^{(n)}) \right)^{-1} \left(\tilde{\mathbf{F}}(\mathbf{U}_{s+1}^{(n)}) + \mathbf{G}_s \right). \quad (2.40)$$

In all numerical examples tested the later iteration demonstrated global linear convergence.

2.2. Adaptive mesh refinement

Let σ_n^h be the stress at the completion of the Newton iterations for given time step n , i.e. $t = t_n$. Since there is no risk of confusion the subscript n will be dropped. For linear elements σ^h is a piecewise constant function. Let $\bar{\sigma}^h$ be the continuous, piecewise linear function in $[0, L]$ which assumes the averaged value of σ^h at each nodal point. The error indicator $\eta_\sigma(e)$ is defined locally over each element e by (Zienkiewicz and Zhu, 1987):

$$\eta_\sigma(e) = \|\bar{\sigma}^h - \sigma^h\|_{0,e}, \quad (2.41)$$

where $\|\cdot\|_{0,e}$ is the L_2 norm. An element e is refined if

$$\eta_\sigma(e)/\sigma_{max} > TOL1, \quad (2.42)$$

where σ_{max} is the absolute value of the maximum attainable stress in the rod, which for impact problems is known in advance. Two neighboring elements e_i and e_{i+1} are merged into one if

$$\eta_\sigma(e_i)/\sigma_{max} < TOL2, \quad \eta_\sigma(e_{i+1})/\sigma_{max} < TOL2. \quad (2.43)$$

Two aspects of the actual implementation details of the FE analysis should be emphasized. The linear system (2.36) (or (2.37)) is tridiagonal and poses no computational problems. Secondly, the most time-consuming parts of the FE procedure are the assembly of the stiffness matrix at each Newton step (because of the nonlinear dependence of the stiffness on the strain) and the assembly of the force vector. They require the execution of the stress update procedure via the return-mapping algorithm which is a computationally expensive operation and is performed once for each element at each Newton step.

Clearly a global uniform h-refinement strategy used to achieve satisfactory spatial discretization will impose severe restrictions on the problem size due to the assembly time issues. In order to avoid this the local criterion (2.42) is applied to each element at the completion of the Newton iteration to refine or coarsen the mesh. If there is no further need to refine the mesh the algorithm proceeds to the next time step. It was found that this approach works very well for the class of SMA hysteretic materials under consideration.

3. Numerical examples

The implementation of the FEM was tested in three different numerical examples. The step loading problem under conditions of pseudoelasticity ($T > A_f$) presented in the next section is used to compare the numerical solution to existing analytical solutions (Bekker et al., 2002; Chen and Lagoudas, 2000). It is also used to demonstrate the capabilities of the adaptive mesh refinement strategy. Secondly, a problem with pulse boundary conditions is solved, again under pseudoelastic conditions. The third problem also features a pulse boundary condition but at a lower temperature ($T < M_s$) so only detwinning of martensite is involved.

Table I. Material parameters used in the uniaxial SMA model.

Material constant	Value	Material constant	Value
E^A	$70 \times 10^9 \text{ Pa}$	$\frac{d\sigma}{dT}$	$7.0 \times 10^6 \text{ Pa}/(m^3K)$
E^M	$30 \times 10^9 \text{ Pa}$	M_f	$275 \text{ }^\circ K$
α^A	$22 \times 10^{-6}/K$	M_s	$291 \text{ }^\circ K$
α^M	$10 \times 10^{-6}/K$	A_s	$295 \text{ }^\circ K$
H	0.05	A_f	$315 \text{ }^\circ K$

The material properties (Table I) for all model problems are taken from (Qidwai and Lagoudas, 2000a) and represent generic NiTi SMA properties. In addition to that for all numerical simulations the length of the rod was taken to be 0.5m. All calculations were performed on a 933 Mhz PIII machine running Windows NT.

3.1. Step loading problem

The fixed impact stress initial-boundary value problem⁴ is defined by setting the boundary condition to be the step function:

$$\sigma_0(t) = \begin{cases} 0 & \text{for } t \leq 0 \\ \sigma_0 & \text{for } t > 0 \end{cases}. \quad (2.44)$$

The strain level ε_0 which causes the constant impact stress σ_0 can be found from equation (2.13). This particular boundary condition is chosen because it is a natural starting point for nonlinear hyperbolic equations and because there are existing analytical solutions for it.

⁴When the same initial boundary value problem is reformulated as an initial problem on an infinite domain with the initial condition being a step function it is usually referred to as the *Riemann* problem.

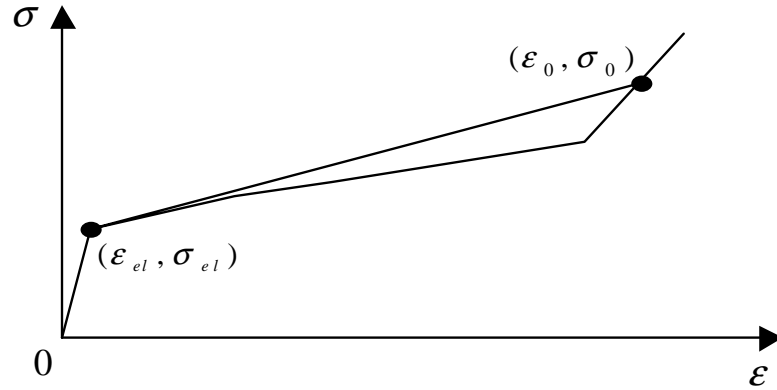


Fig. 7. Schematic of the loading portion of a stress-strain relationship and the critical points defining the solution to the problem.

3.1.1. Analytical solutions to the step loading problem

The structure of the solution depends strongly on the impact stress σ_0 . Let the pair $(\varepsilon_{el}, \sigma_{el})$ be the point on the hysteresis curve that corresponds to the start of the phase transformation. In this example σ_0 it is taken to be sufficiently high so that full phase transformation has occurred. It is also required that the value of σ_0 be high enough, so that the graph of the stress strain relationship of the SMA is below the line connecting the points $(\varepsilon_{el}, \sigma_{el})$ and $(\varepsilon_0, \sigma_0)$ (see Figure 7).

Following Bekker et al. (2002); Chen and Lagoudas (2000) it can be shown that for material with initial linear stress-strain relationship prior to the onset of phase transformation the solution has the following two-shock structure:

$$\sigma(x, t) = \begin{cases} \sigma_0 & \text{for } 0 \leq x/t \leq V_{ph} \\ \sigma_{el} & \text{for } V_{ph} < x/t \leq V_{el} \\ 0 & \text{for } V_{el} < x/t \end{cases}, \quad (2.45)$$

$$T(x, t) = \begin{cases} T_0 & \text{for } 0 \leq x/t \leq V_{ph} \\ T_{el} & \text{for } V_{ph} < x/t \leq V_{el} \\ 0 & \text{for } V_{el} < x/t \end{cases} , \quad (2.46)$$

where T_0 is the temperature corresponding to the impact stress σ_0 and T_{el} is the temperature just prior to the onset of the phase transformation. The faster shock is a linear thermoelastic elastic shock and has velocity

$$V_{el} = \sqrt{\frac{\sigma_{el}}{\rho \varepsilon_{el}}} . \quad (2.47)$$

This shock is due to the shock type of the boundary condition and the initial linear stress-strain response.

The second, slower shock, is a transformation shock which travels with velocity

$$V_{ph} = \sqrt{\frac{\sigma_0 - \sigma_{el}}{\rho(\varepsilon_0 - \varepsilon_{el})}} . \quad (2.48)$$

This shock occurs not only because of the boundary condition but also because of the convex-down nature of the stress-strain relationship for $\varepsilon > \varepsilon_{el}$. Higher stress levels travel with higher velocity than lower stress levels which make the shock self sustained and independent of the boundary condition (see (Godlewsky and Raviart, 1996, pg. 83-97) for a general discussion as well as (Bekker et al., 2002; Chen and Lagoudas, 2000) for solutions specific to SMA materials). The phase transformation shock specifies the point of abrupt phase transition. For material points with $x \leq V_{ph}t$ the material is in the martensitic phase and the region $x \geq V_{ph}t$ is still in the austenitic phase.

Note that the adiabatic heat equation (2.22) does not provide for a completely linear initial response. However, prior to the onset of phase transformation, $\xi = 0$

and the heat equation (2.22) can be linearized as follows:

$$T = T_R \left(1 - \frac{\alpha}{\rho c} \sigma \right) + \mathcal{O} \left(\left(\frac{\alpha \sigma}{\rho c} \right)^2 \right). \quad (2.49)$$

By neglecting the higher order terms in (2.49) the remaining linear part can be substituted in (2.13) to obtain a completely linear adiabatic stress-strain response. The linear approximation in (2.49) is justified in the thermoelastic range before commencement of phase transformation because $\frac{\alpha \sigma_0}{\rho c} \approx 10^{-3}$. If equation (2.22) is not linearized the elastic shock will be replaced by a continuous function with very high gradient. The velocity of the points on the graph of this function will deviate from the velocity V_{el} of the elastic shock by $\approx 10^{-5}$.

3.1.2. Numerical results for the step loading problem

For all numerical simulations the impact stress level is $\sigma_0 = -400 MPa$ corresponding to impact strain of $\varepsilon_0 = -0.0635$. The reference temperature is $T_R = 320 \text{ }^\circ K$. The FEM solver was set to use the backward difference time integration scheme and the Newton-Raphson method to solve the nonlinear system (2.36). The Newton-Raphson iteration showed quadratic convergence at all time steps except for the first few ones when the shock were forming. In the cases when it was diverging the alternative direct iteration (2.40) approach was used.

Significant computational savings can be obtained if isothermal instead of adiabatic conditions are assumed. In an *isothermal* problem the temperature is held constant $T = T_R$ and the balance of energy (2.2) is not considered. Thus the quasi-static hysteresis of the material is used instead of solving equations (2.13) and (2.22). For a NiTi SMA with the material data from Table I the difference between the adiabatic and isothermal hysteresis is shown in Figure 8. The shape of the hysteresis is the same and the differences in the transformation portion will not affect the structure

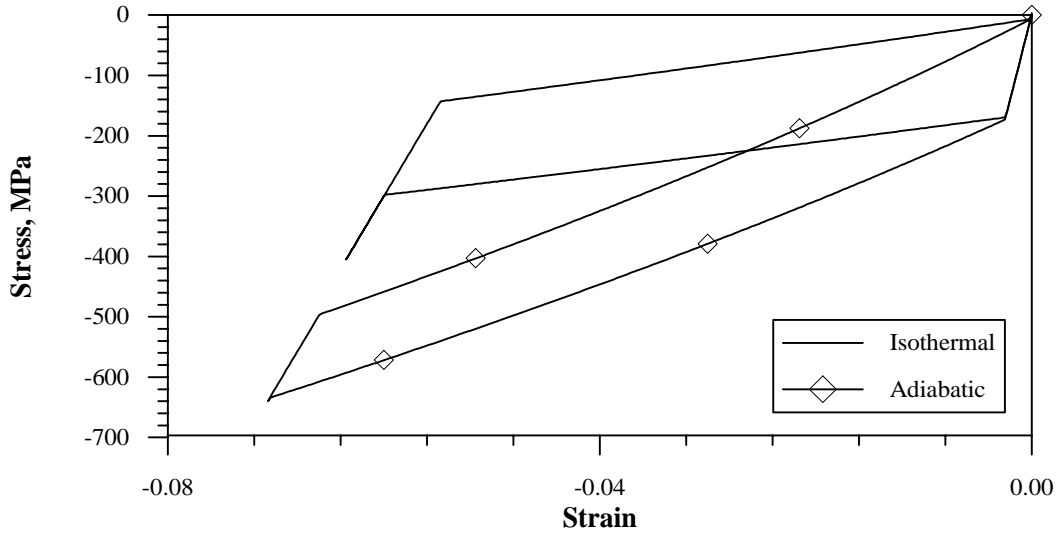


Fig. 8. Difference between adiabatic and isothermal loading. Shown are an adiabatic and an isothermal path at $T_R = 320 \text{ }^\circ\text{K}$ for the material data in Table I. Under adiabatic conditions higher stress levels are required to complete the phase transformation compared to isothermal hysteresis loops.

of the solution provided that σ_0 is well above the stress level required to finish the transformation. Consequently, no matter whether isothermal or adiabatic conditions are assumed the shock speeds V_{ph} and V_{el} will only depend on the values for ε_{el} , σ_{el} , ε_0 , σ_0 . From a computational point of view this simplification avoids the iteration process (2.25),(2.26) (typically 6-7 iterations) which results in a significant reduction in computational time. While the structure of the solution is not compromised very fine spatial meshes can be explored for the purposes of comparing analytical and numerical solutions.

For the isothermal hysteresis (Figure 8) an impact stress of $\sigma_0 = 400\text{MPa}$ is sufficient for the full completion of the phase transformation under isothermal conditions. The onset of phase transformation begins at $\sigma_{el} = -195\text{MPa}$ for a strain $\varepsilon_{el} = 2.78 \times 10^{-3}$. Given this, the speed of the two shocks (2.48) and (2.47) are found

to be:

$$V_{ph} = 723m/s \quad (2.50)$$

$$V_{el} = 3294m/s \quad (2.51)$$

Based on the first few numerical results (Figure 9) and (Figure 10) several observations can be made. First, all numerical solutions have the expected two shocks - one elastic and another corresponding to the phase transformation. Fixed meshes with coarse spatial discretizations have oscillations close to the phase shock location. A comparison of the two meshes in Figure 9, both for a fixed time-step of $\tau = 0.1\mu s$ at time $t = 30\mu s$ shows that oscillations can be eliminated by refining the mesh. Secondly, the backward difference scheme which was used in these computations, introduced numerical dissipation which is most pronounced at the elastic shock. Several other members of the Newmark family were tested. Explicit methods as well as the constant acceleration scheme were found to be unconditionally unstable producing highly oscillatory solutions that were diverging with time. Of those methods that were able to converge the backward difference was found to dampen the high frequency oscillations (Figure 9(a)) in the most efficient manner and was subsequently chosen for all future computations. The numerical dissipation can be decreased by appropriately decreasing the time step. The quasi-Newton method used to solve the nonlinear system (2.36) showed quadratic convergence at all time steps but the first few ones when the shock were forming. In that case the alternative direct iteration (2.40) approach was used.

Quantitatively the results obtained by both the fixed and adaptive FEM are in agreement with the analytical solution. In regions away from the shocks the relative difference in the values of the stress for the numerical and the exact solution is less

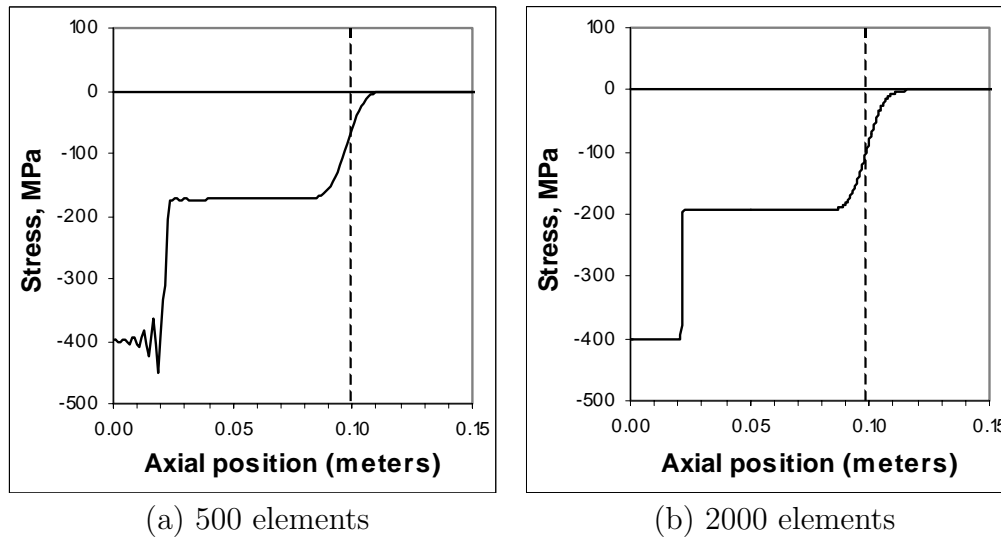


Fig. 9. Comparisons of FEM solutions with different mesh sizes and time steps. Shown are the stress profile at $30\mu s$ for a fixed mesh with 500 (a) and 2000 elements (b). The position of the elastic shock is marked by a dashed line. Numerical oscillations are eliminated for the finer spatial discretization.

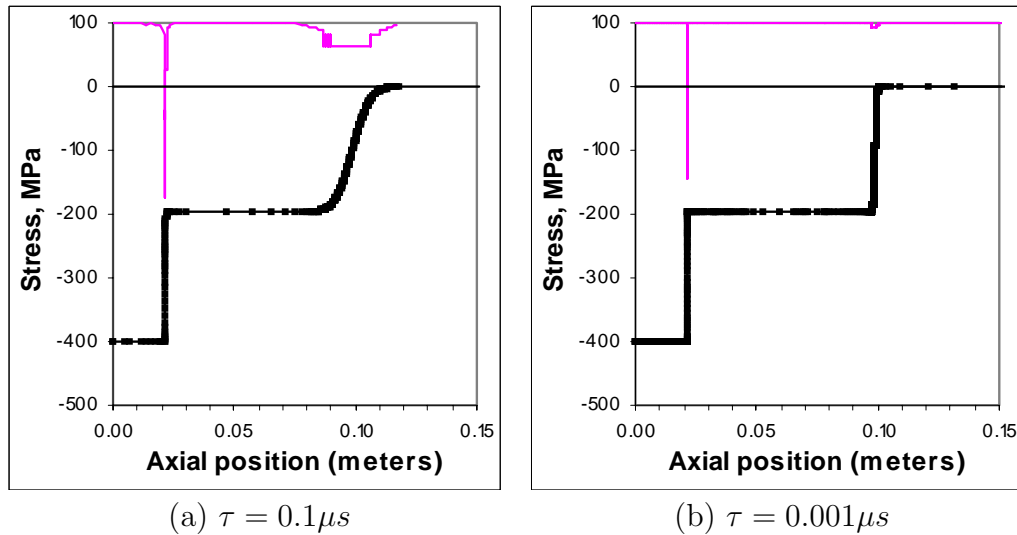


Fig. 10. Adaptive meshing and numerical dissipation. Shown are stress profile at $30\mu s$ for an adaptive mesh with two different time steps – $\tau = 0.1\mu s$ and $\tau = 0.001\mu s$. Mesh nodes are marked with black squares and the thin line at the top shows the density of elements. The linear shock is smeared for the larger time step (a) and is much sharper when a finer step (b) is used.

then 10^{-4} . The accuracy of the solutions therefore is determined based on the quality of the numerical solution close to the shock locations. The interval covering a shock (phase or elastic) where the numerical values for the stress differ from the exact ones by more than 1% is assumed to be the range of uncertainty for the numerical value of the shock location. Consequently, the left and right end of this interval are assumed to be bounds for the position of the shock of the numerical solution.

Based on this measure of error, for a time step of $\tau = 0.1\mu s$ the phase shock is found to travel with velocity in the range $693 - 900m/s$. The velocity of the elastic front is calculated to be in the range $3316 \pm 420m/s$. These results are the same for a fixed (Figure 9(b)) and adaptive mesh (Figure 10(a)). This indicates that the adaptive and fixed FEM converge to the same solution.

The smearing of the stress profile in the region of the elastic shock is due to the time-integration scheme. When the time step is decreased the slope becomes steeper and eventually converges to the shock. For an adaptive solution with a time step $\tau = 0.001\mu s$ (the same computation for a fixed mesh was time prohibitive) the calculated values for the phase shock are now in the range $723 - 733m/s$ and the elastic shock is within the bounds $3256 - 3366m/s$ (Figure 10(b)). This indicates that the lower bound for the transformation shock is very close to the actual value (2.50) and that the elastic shock (2.51) is virtually in the middle of the suggested numerical range. The relative error in the predicted value for the phase shock velocity decreases from 24% for $\tau = 0.1\mu s$ down to 1.3% for $\tau = 0.001\mu s$. The error in the elastic shock speed decreases from 12% to 1.1% which is a clear indication that the FEM algorithm is converging to the exact solution.

An inspection of Figure 9 reveals that there are large regions in the bar with no variation in the stress. This is fully utilized by the adaptive approach. Figure 10(a) shows an adaptive FE solution with the same time step as the solution on Figure 9(b)

Table II. Execution times for fixed and adaptive meshes.

Time	Fixed Mesh		Adaptive Mesh	
	Elements	Time (min)	Elements	Time (min)
10 μs	16000	56	161	1:12
20 μs	16000	113	199	2:37
40 μs	16000	226	256	6:10
80 μs	16000	451	301	15

and an adaptive tolerance (see (2.42)) set to 10^{-4} . This accuracy is comparable to the one of a fixed mesh with 2000 elements. The maximum number of elements that the adaptive mesh contained was 305. The order of magnitude fewer number of elements in the adaptive meshes induced a corresponding order of magnitude decrease in the computational time.

A comparison in the performance of the fixed and adaptive FE methods is given in Table II. The time step is $\tau = 0.01\mu s$. The number of elements for the fixed FEM is 16000. The adaptive solution was chosen so that it had comparable accuracy with the one for the fixed mesh solution. A comparison of the execution times for the fixed and adaptive methods shows that the adaptive procedure delivers an order of magnitude improvement in performance.

3.2. Square pulse loading problem in pseudoelastic conditions

A more realistic initial-boundary value problem is one for which, instead of step loading, the boundary condition is a square pulse, that is

$$\sigma_0(t) = \begin{cases} 0 & \text{for } t \leq 0 \\ \sigma_0 & \text{for } 0 < t < t_{pulse} \\ 0 & \text{for } t \geq t_{pulse} \end{cases} \quad (2.52)$$

where t_{pulse} is the duration of the pulse. Due to the complicated constitutive response and boundary conditions there is no analytical solution to be compared with. Moreover, there are unresolved questions regarding the uniqueness of the weak solution for times $t > t_{pulse}$ when unloading takes place.

The stress level used for the numerical simulation is $\sigma_0 = 800MPa$ and the initial temperature is $T_R = 320 \text{ }^\circ K > A_f$. The simulation is done for adiabatic conditions, utilizing both equations (2.13) and (2.22) to calculate the adiabatic response of the SMA. The stress level is chosen so that the full adiabatic hysteresis loop can be realized (see Figure 8). The pulse length is $t_{pulse} = 10\mu s$ and the time step is $t = 0.001\mu s$.

The evolution of the stress and temperature in the rod up to $90\mu s$ is shown in Figures 11, 12 and 13. As predicted by (2.45) the two-shock solution for the stress is clearly visible at the end of the pulse load at $t = 10\mu s$ (Figure 11). The temperature profile (Figure 13) also has two shocks (equation (2.46)). The maximum temperature $T_0 = 378.8 \text{ }^\circ K$ is achieved in the region of full phase transformation. The jump in the elastic shock is $T_{el} - T_R = 0.66 \text{ }^\circ K$ and for this reason it is not clearly visible in the figure.

The most noticeable feature observed in Figure 12 is the structure of the unloading pulse. Again a two wave shock structure is seen that corresponds to the initial elastic unloading and the following reverse transformation $M^t \rightarrow A$ as can be seen from the stress profile at 10 and $20\mu s$. Both unloading shocks travel faster than the forward phase transformation shock. When the faster unloading front catches up with the forward phase transformation shock ($t \approx 22\mu s$) a left-travelling reflection is

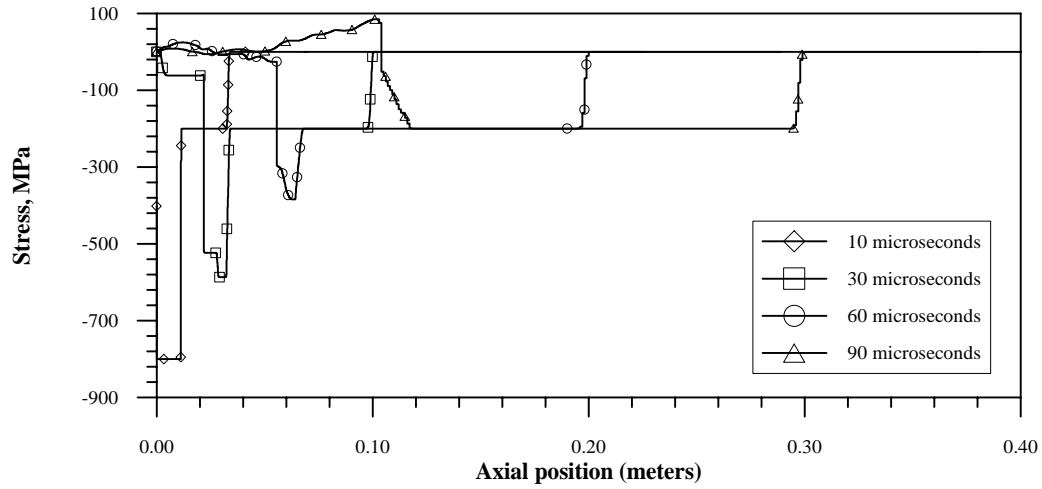


Fig. 11. Stress profile at different instances of time for a square pulse in adiabatic loading.

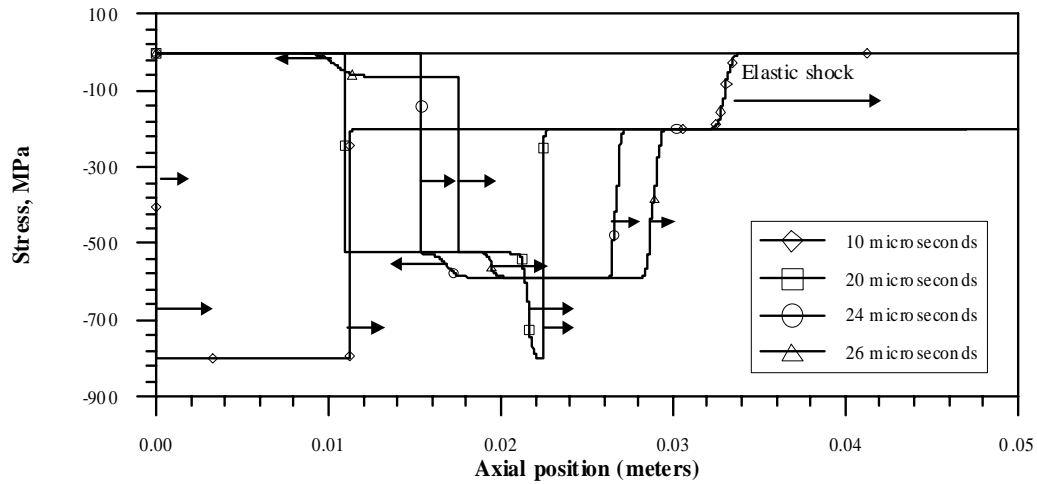


Fig. 12. Magnified view of solution to square pulse adiabatic problem. The unloading ($10\mu s$) produces two right-travelling shock waves ($20\mu s$). The faster unloading wave reflects off the transformation shock ($\approx 21\mu s$) and forms a left-travelling wave ($24\mu s$). What follows is a series of complicated reflections that gradually kill the initial non-linear shock.

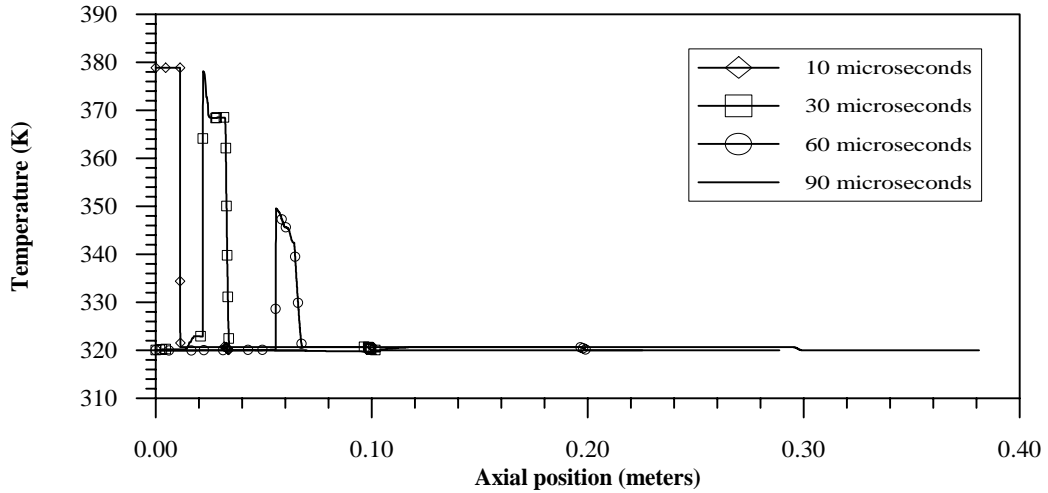


Fig. 13. Temperature profile at various times. The jump at the forward transformation shock is $T_0 - T_{el} = 58.2 \text{ }^\circ\text{K}$. The elastic shock is not visible clearly because of its small magnitude of $T_{el} - T_R = 0.66 \text{ }^\circ\text{K}$.

generated. The left-travelling wave, as seen for $t = 24\mu\text{s}$, partially reflects from the slower unloading shock and partially continues ($t = 26\mu\text{s}$) until it reflects off the left end of the rod. A complicated series of reflection waves follows. The first reflection results in approximately 34% decrease of the peak stress level ($t = 24\mu\text{s}$). The picture becomes even more complicated when the slower unloading shock eventually catches up with the forward travelling phase transformation shock. Eventually the peak stress levels are reduced to values below σ_{el} , the critical stress corresponding to the onset of phase transformation. The temperature profile at $t = 90\mu\text{s}$ is hardly visible because the material is entirely in the elastic range and the temperature in the rod is very close to the reference temperature. The large amounts of latent heat generated during the initial loading phase are gradually consumed in the reverse transformation as the stress is reduced within the elastic limits.

For pulse loading it is physically meaningful to compute the energy dissipation due to the phase transformation. If $P(\tau)$ is the work done by the external forces at

the left end of the rod from $t = 0$ up to $t = \tau$, $\mathcal{K}(\tau)$ is the kinetic energy of the rod at time $t = \tau$ and $\mathcal{W}(\tau)$ is the stored elastic energy of the rod then the energy dissipation is defined by

$$D(\tau) = \frac{P(\tau) - (\mathcal{K}(\tau) + \mathcal{W}(\tau))}{P(\tau)}. \quad (2.53)$$

The quantities P , \mathcal{K} and \mathcal{W} given by

$$\begin{aligned} P(\tau) &= \int_0^\tau \sigma(0, t)v(0, t)dt, \\ \mathcal{W}(\tau) &= \frac{1}{2} \int_0^L \sigma(x, \tau)\varepsilon^e(x, \tau)dx, \\ \mathcal{K}(\tau) &= \frac{1}{2} \int_0^L \rho(v(x, \tau))^2 dx, \end{aligned}$$

can be easily computed numerically at each time step.

The calculations show (Figure 14) that the dissipation level goes from 40% at the end of the pulse ($t = 10\mu s$) to 64% at $t \approx 22\mu s$ when the faster unloading wave reflects off the forward travelling transformation wave. The high stress levels are then gradually reduced within the elastic limits. The energy dissipation reaches approximately 84% at $100 \mu s$, shortly before the elastic front reaches the right end.

3.3. Detwinning induced by a pulse load

In this numerical simulation the same boundary condition (2.52) as in the previous section is used. The initial temperature is set to $T_R = 295 \text{ }^\circ K$ which is in the detwinning range and the material is initially in the M^t state. The stress pulse has magnitude $\sigma_0 = 400 MPa$ which is sufficient to complete the detwinning and then obtain the elastic response of the martensite phase.

There is no latent heat generation during the detwinning deformation. If it is assumed that all the work dissipated through inelastic deformations is transformed

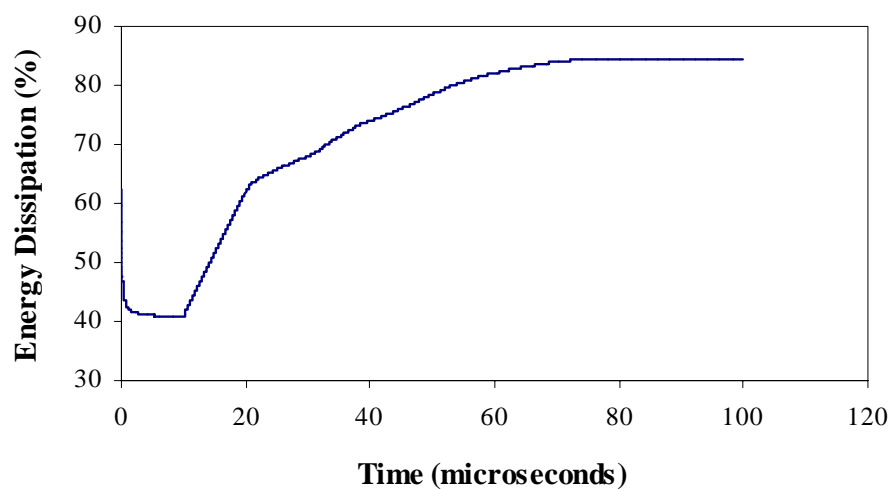


Fig. 14. Energy dissipation for a $10 \mu s$ square pulse in adiabatic conditions.

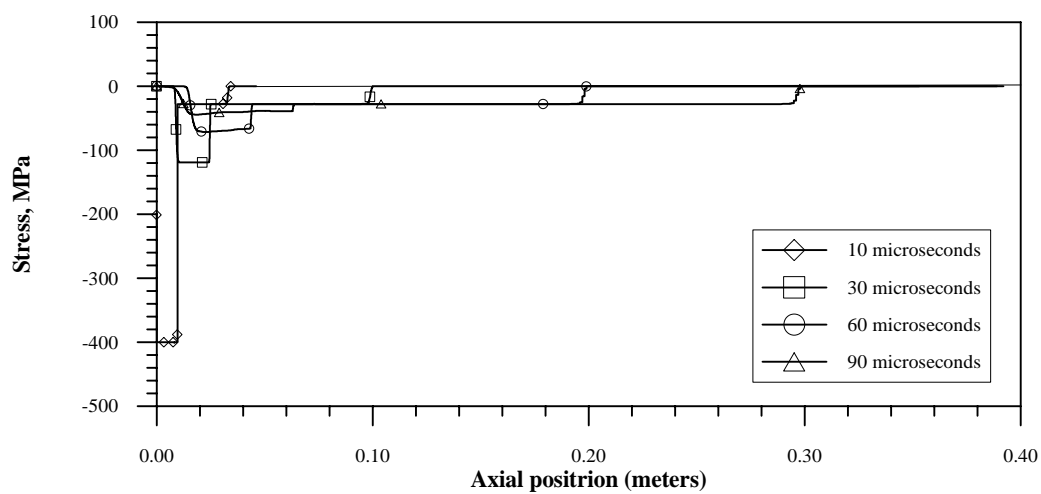


Fig. 15. Stress profiles at various times for a square pulse in detwinning conditions. The attenuation of the stress to values within the elastic material response is clearly visible.

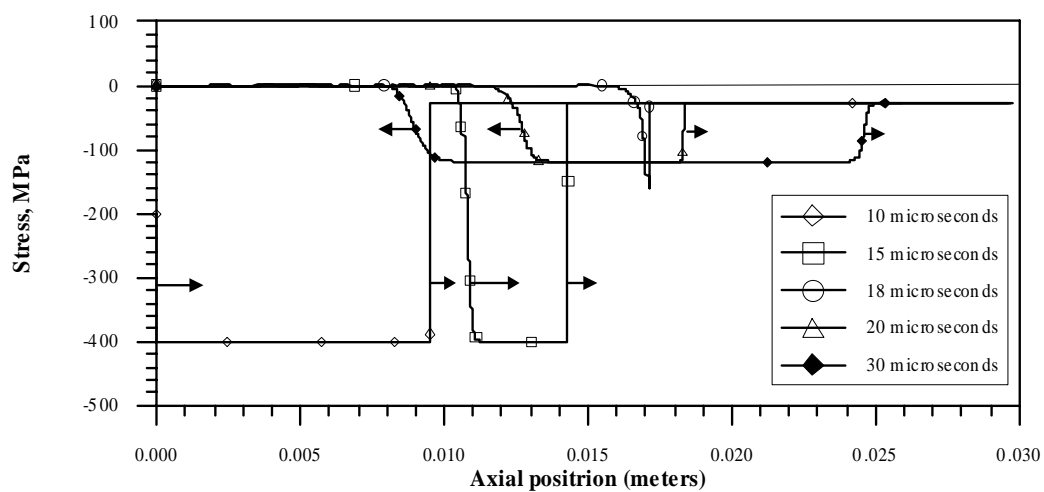


Fig. 16. Magnified view of the stress profiles. The stress profiles shown in Figure 15 in the region close to the left end of the rod are magnified. Direction of the shock velocities are indicated by arrows.

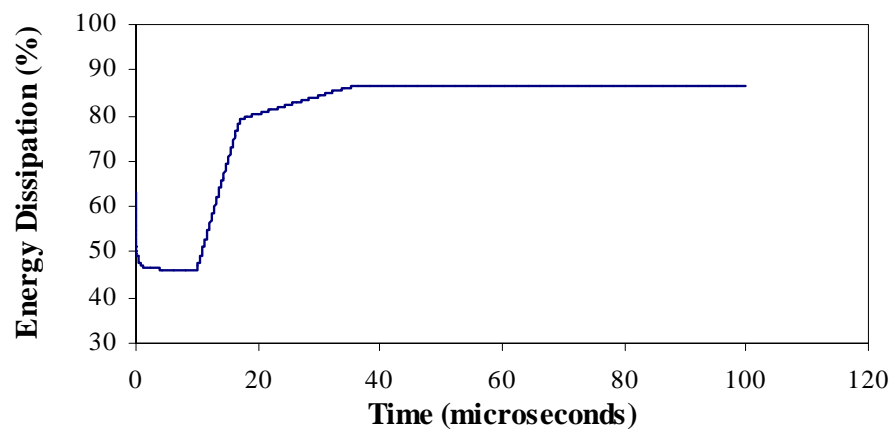


Fig. 17. Energy dissipation for a 10 μs square pulse (detwinning).

into heat, then the change in temperature would be $\approx 2 \text{ }^\circ K$. Therefore it is both physically and computationally justified to perform the simulation in an isothermal setting. The loading part of the hysteresis is of the same type as the loading part ($A \rightarrow M^d$) of the stress-strain relationship for stress induced martensite. Therefore for the duration of the pulse a two-shock structure for the stress distribution can be expected (see equations (2.45), (2.50) and (2.51)). This is observed clearly for the stress profile at $t = 10\mu s$ in Figure 15.

The unloading is completely elastic and a single linear shock forms, travelling at the speed of the forward elastic shock (both the initial loading response and unloading are linear with the elastic modulus of martensite). The unloading shock is therefore fast enough to catch up with the nonlinear shock caused by the detwinning. This is followed by a series of reflections between the left end (which is traction free after the pulse is over) and the forward propagating detwinning shock. A magnified view of the stress profile at several different instances of time is presented in Figure 16.

The energy dissipation (Figure 17) in the rod follows a similar path as in the previous numerical simulation. The first significant rise in the dissipation levels occurs immediately after unloading, at $t = 10\mu s$. After the unloading wave reaches the forward propagating detwinning front at $t \approx 18\mu s$ a new rise in the dissipation occurs leading to final levels of approximately 86%. It should be noted that this case is not equivalent to the pulse load in pseudoelastic conditions because of different initial stress levels. Another difference with the pseudoelastic case is that the material is permanently deformed and in order to recover its shape the rod has to be reheated.

4. Comparisons with impact loading experiment of an SMA rod

The computational studies in the previous section indicate that adaptive FEM methods can be successfully applied to impact problems in SMA materials. In this section the numerical scheme is used to simulate recent experimental data on impact induced detwinning an SMA bar.

The experimental data used here is reported by Lagoudas et al. (2003). The dynamic response of a nearly equiatomic NiTi alloy rod is obtained in a split Hopkinson bar test. The testing itself was performed at the Research Center for Mechanics of Solids, Structures and Materials at the University of Texas at Austin by Prof. Ravi-Chandar and Dr. Khalid Sarh. The details of the apparatus and the specimen preparation can be found in (Lagoudas et al., 2003). For the sake of completeness, they are summarized in the next section along with the test results.

4.1. Experimental data

Hopkinson bar apparatus has become standard in the characterization of the dynamic response of materials and detailed descriptions are provided in many handbooks and textbooks (Graff, 1975; Kolsky, 1963). A schematic of the impact device is given in Figure 18.

The apparatus consists of a striker bar, an input bar and an output bar, all of diameter $d = 15.5mm$ and all made of a 4340 steel, quenched and tempered to a martensitic state. The yield strength of these bars is about 1.8 GPa and they remain elastic during the impact experiments. The density of the bars is $\rho = 7800kg/m^3$, the measured bar wave speed $C_b = \sqrt{E_b/\rho} = 5300m/s$ and E_b is the modulus of elasticity of the steel bar. The striker bar (18) of length L is propelled from an air gun at speeds in the range of 10 to 40 m/s . This striker impacts the input bar which is 1.7m long.

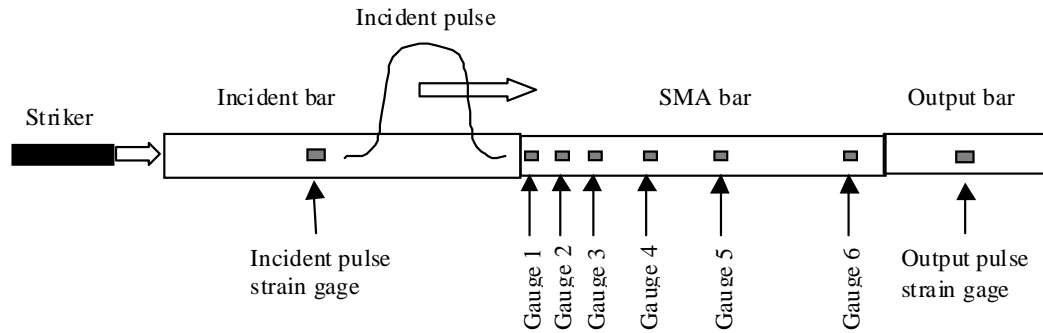


Fig. 18. Geometry and arrangement of strain gauges in Hopkinson apparatus (Figure not drawn to scale).

A one dimensional compression wave propagates into both bars. Since the striker bar is short, the reflected tension pulse arrives at the striker-input bar interface at a time $t_{pulse} = 2L/C_b$. At this point, the striker comes to a stop and is disengaged from the input bar. Hence, a compression pulse of duration t_{pulse} is propagated down the length of the input bar. This wave is coupled into the specimen which is in contact with the far end of the input bar. Due to the impedance mismatch between the specimen and the input bar, part of the pulse is reflected back into the input bar and part of the pulse propagates into the specimen. A strain gauge mounted at about the middle of the input bar is used to monitor the incident compressive pulse and the reflected tensile pulse propagating in the input bar. The wave propagating through the specimen, gets coupled into the output bar, again with a reflected component due to the impedance mismatch. The output bar is free at the far end and so a tensile pulse reflects from the far end of the output bar and is unable to transmit into the specimen. Hence the specimen is loaded only once. A strain gage mounted at the middle of the output bar is used to monitor the strain pulses, in particular the first transmitted pulse, in the output bar.

Experimental data is available for a single SMA specimen, 345 mm long and

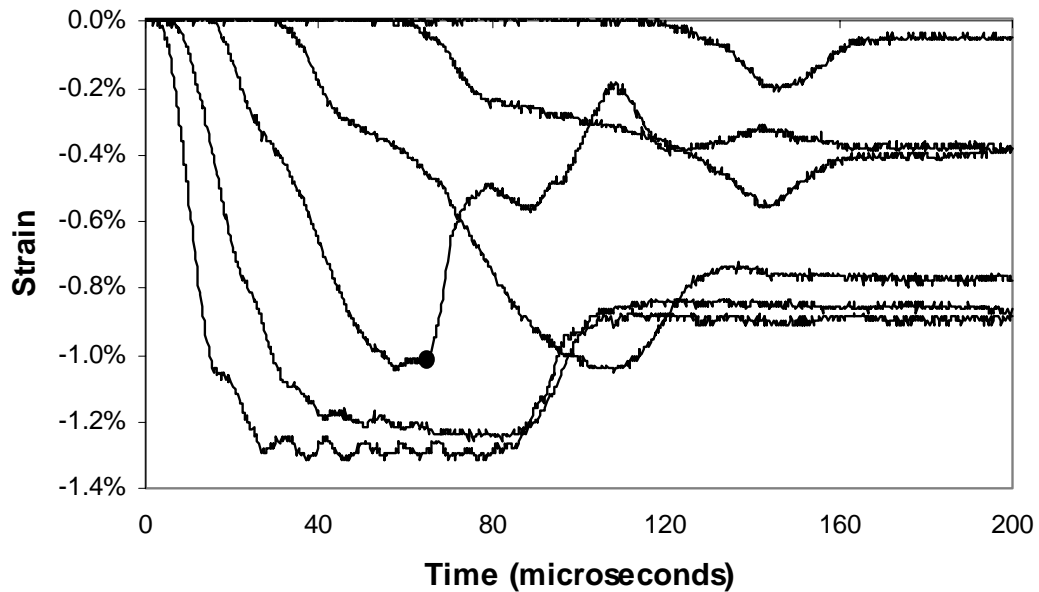


Fig. 19. Experimental data on wave propagation in an SMA rod (Lagoudas et al., 2003). Strains are measured by the gauges mounted on the SMA bar. Gauge 3 suffered a partial debond at the point indicated by the dark circle and hence the data beyond this time should not be interpreted.

tested at room temperature (nominally $20\text{ }^{\circ}\text{C}$). The initial preparation of the specimen ensured that it was in the twinned martensitic state with all previous deformation history deleted by appropriate annealing. (Lagoudas et al., 2003). The data consist of strain history obtained by six strain gauges placed at distances 10 mm , 20 mm , 40 mm , 80 mm , 160 mm and 320 mm from the impact end. The output from these gauges is shown in Figure 19. Strain gauge number 3 (40 mm) suffered a partial debond during the test and hence the results from this gauge are not meaningful beyond the point marked by the dark dot in the figure. The elastic wave in the input bar was not recorded due to an error in the device; all other gauges worked well and recorded the strain profile as the wave propagated down the length of the SMA rod.

As seen in Figure 19, the strain in the first two gauges increases rapidly to a

level of about 1.3% and levels off as the load from the input bar levelled off. The oscillations seen in these gauges near the plateau are Pochhammer-Chree oscillations that appear in bars. At around 290 μs the unloading wave from the end of the loading pulse reaches the first two gauges and the strain begins to decrease; however, because the strains beyond 0.3% were the result of detwinning (see the quasi-static results in Figure 20), these strains are not recovered and a permanent strain of about 1% is left at these locations. The signal in gauge 4 clearly indicates the dispersion of the wave - higher strain levels propagate at significantly slower speeds and arrive later at the gauge location. Hence a broadening of the strain pulse can be seen - the peak in the strain at gauge 4 occurs 75 μs after elastic wave arrival while it occurs in about 20 μs in gauge 1. This delay also results in the peak strain not being sustained for too long as the elastic unloading pulse reaches the gauge quickly; once again a residual strain of peak strain - 0.3% is left at this gauge location. The same behavior is seen in gauge 5 where due to its distance from the impact end, and due to the slowness of the inelastic waves, the peak strain reached is only about 0.5%. Once again a residual strain is left in this location. In gauge 6, the reflected wave from the end of the SMA rod (left free in this experiment) causes unloading of the gauge; a very small, but measurable permanent strain or detwinning is observed in this location. Subsequent to the test, the rod was heat treated through a temperature cycle taking it above A_f first, holding for 1 hour and then cooling below M_f and warming back to room temperature. All strain gauges recovered their original state indicating full recovery of the specimen.

4.2. Numerical simulations of the Hopkinson bar experiment

4.2.1. Quasi-static hysteresis and calibration of the SMA model

In order to obtain preliminary information on the mechanical behavior of this material quasi-static compression test was performed on a short SMA specimen in a standard testing machine. The specimen used was from the same batch as the long rod used in the impact test. The testing was performed by Eric Vandygriff at the Active Materials Lab at Texas A&M University. Since the dynamic test involved only detwinning of martensite the quasi-static tests were done at room temperature. These tests were used to obtain the stiffness of the martensitic phase E^M and the critical stresses σ_s and σ_f for onset and finish of detwinning. The material constants used for the detwinning model are summarized in Table III. The hysteresis simulated by the model (Section 1.3) and the actual hysteresis from the quasi-static test are given in Figure 20.

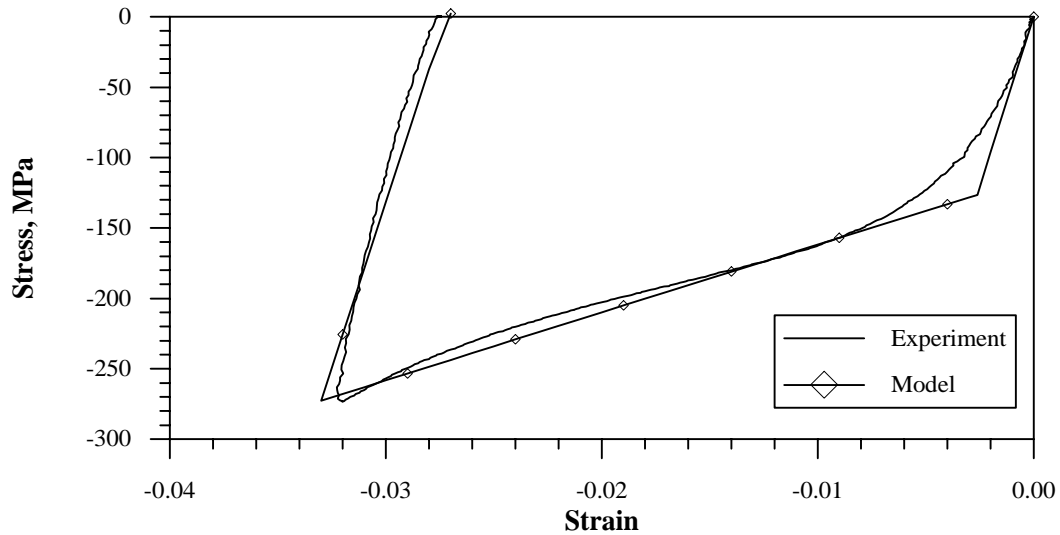


Fig. 20. Quasi-static hysteresis of SMA specimen and the model simulation.

Table III. Material parameters used in the uniaxial SMA model for detwinning.

Material constant	Value	Description
E^M	42×10^9 Pa	Modulus of elasticity in martensite
H	0.027	Maximum detwinning strain
σ_s	-125MPa	Start of $M^t \rightarrow M^d$ deformation
σ_f	-273MPa	Completion of $M^t \rightarrow M^d$ deformation
Y^d	$\sigma_s H$	
ρb^d	$\sigma_f H - Y^d$	

4.2.2. Dynamic hysteresis and rate independence

The results of the Hopkinson bar experiment can be used to extract the dynamic stress-strain response by applying the theory of one-dimensional wave propagation in plastic rods due to (Rakhmatulin, 1945; Taylor, 1958; Von Karman and Duwez, 1950). The idea is a simple extension of the rod theory for elastic waves. Let us assume that stress is only a function of strain, i.e. $\sigma = \sigma(\varepsilon)$. Then the balance of linear momentum (2.1) can be written in the form

$$u_{tt} = \frac{\sigma'(\varepsilon)}{\rho} u_{xx}. \quad (2.54)$$

Note that this is not an incremental theory, but a total strain theory; therefore unloading cannot be considered here. The wave speed $C(\varepsilon)$ of disturbances is no longer a constant as in the linear elastic case, but a function of strain:

$$C(\varepsilon) = \sqrt{\frac{\sigma'(\varepsilon)}{\rho}}. \quad (2.55)$$

The main result of this one dimensional theory is that a given strain (or stress) level will propagate into the rod with a characteristic speed given by equation (2.55).

If the propagation speed of strain waves in a one-dimensional rod is known (measured with strain gauges as in the experiment discussed above), equation (2.55) can be inverted to determine the stress-strain behavior of the material:

$$\sigma(\varepsilon) = \int_0^\varepsilon \sigma'(\zeta) d\zeta = \rho \int_0^\varepsilon C^2(\zeta) d\zeta. \quad (2.56)$$

This representation of the wave speed is used to extract the constitutive behavior of the material (Bell, 1960; Kolsky and Douch, 1962).

(Lagoudas et al., 2003) calculated the propagation speeds of different strain levels from the results shown in Figure 19. The time of arrival of different strain levels at each one of the five gauges were determined from the strain measurements. The speed of each strain level $C(\varepsilon)$ was then determined from the known distances between the gauges (Figure 21). The results are averaged for all the strain gauges and a smooth approximation of $C(\varepsilon)$ is obtained (for details, see Lagoudas et al., 2003). After numerical integration of equation (2.56), the stress strain relationship associated with the detwinning deformation in the SMA rod. It is shown in Figure 22 and for comparison, the quasi-static hysteresis is also plotted. An important observation that can be made from this figure is that the quasi-static and dynamic stress-strain response of the SMA in detwinning conditions are identical, thus justifying the assumption of rate-independence used throughout the SMA model.

4.2.3. Numerical simulations

A numerical simulation was performed and results were compared with the experimental data. As indicated earlier, due to a trigger failure, the signal in the input bar was lost so only the readings of the six strain gauges on the specimen were available. In order to supply proper boundary conditions the signal from the first strain gauge (at 10mm) was used and the remaining gauges were simulated. Gauge number 3 was

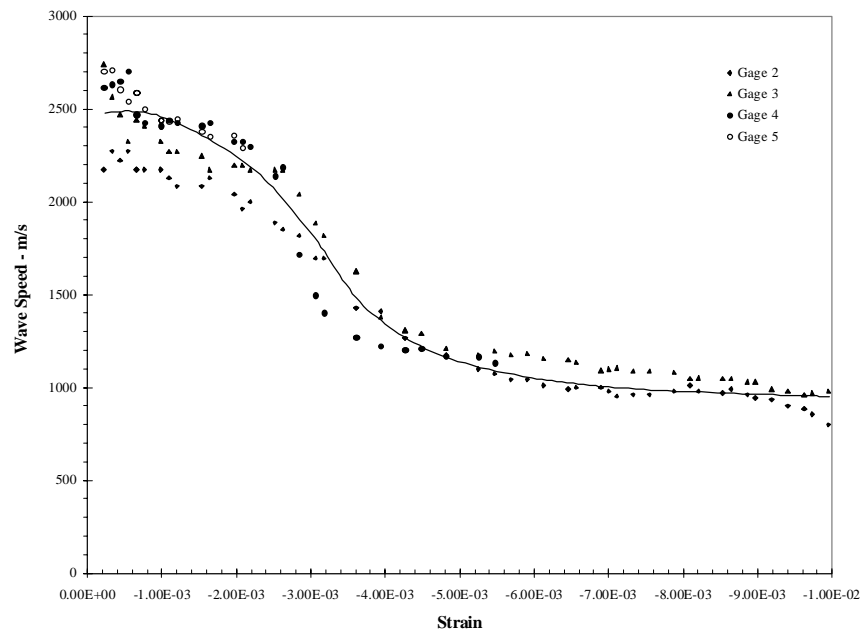


Fig. 21. Variation of the wave speed with the strain level as determined from the strain measurements. The line is an eyeball fit to indicate the data trend. Cubic fits over short segments were used to determine the wave speed corresponding to each strain level in the determination of the stress-strain behavior (Lagoudas et al., 2003).

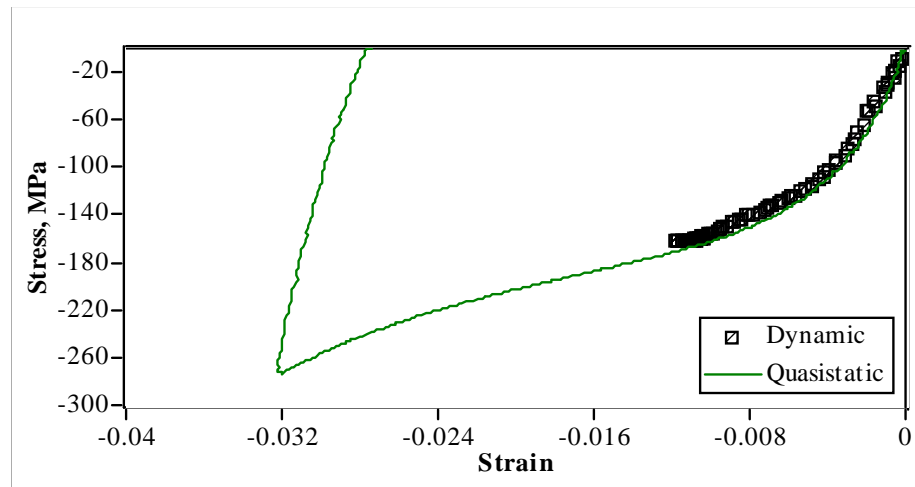


Fig. 22. Stress-strain response evaluated from one-dimensional wave propagation measurements (scatter plot) and quasi-static data (solid line). The close match between the two indicated that the material response is rate independent (Lagoudas et al., 2003).

not included in the modelling because it unglued during the test.

The Hopkinson bar experiment was done at room temperature and due to the heat treatment of the specimen prior to the test it was in fully twinned martensitic state. The SMA model was applied in detwinning conditions (Section 1.3) with the material constants given in Table III, see also Figure 20. The adaptive FEM scheme was chosen because of its accuracy and ability to predict precisely the positions of both the elastic and transformation shocks. The results are presented in Figure 23.

As expected from the numerical examples studied in Section 3.1 the strain wave splits into an elastic and a transformation front. The transformation front timing and magnitude at all strain gauges is in good agreement with the experiment. The small oscillations observed at the first two gauges are due to surface effects caused by the impacting projectile. Such effects cannot possibly be modeled within a 1-D formulation.

There is, however, a noticeable disagreement in the timing of the elastic fronts.

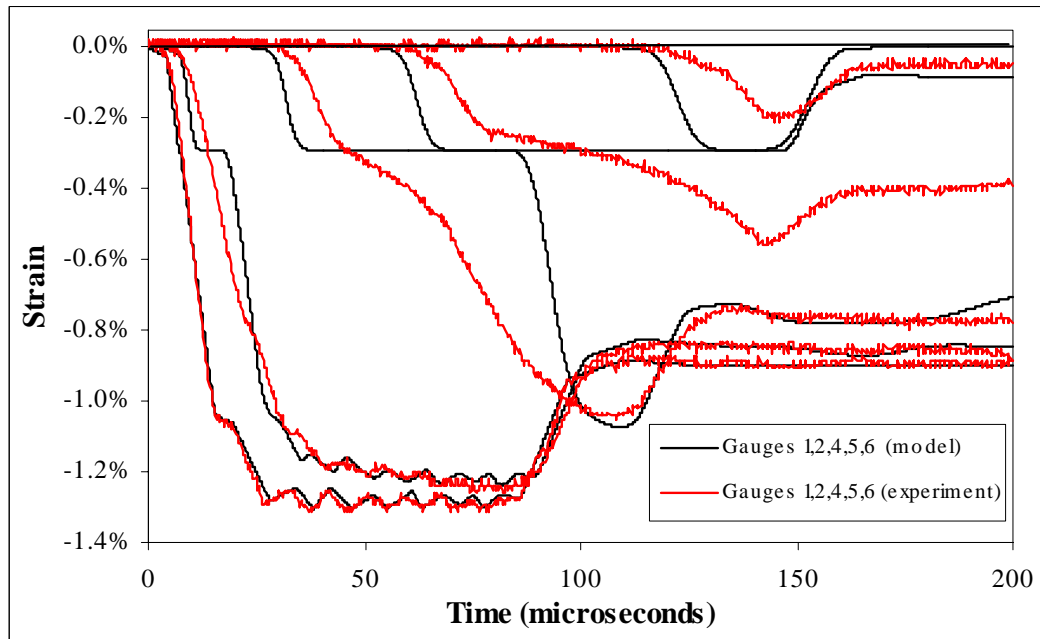


Fig. 23. An adaptive FE analysis of experimental data using SMA constitutive model with linear hardening. The first strain gauge is used to define the boundary condition.

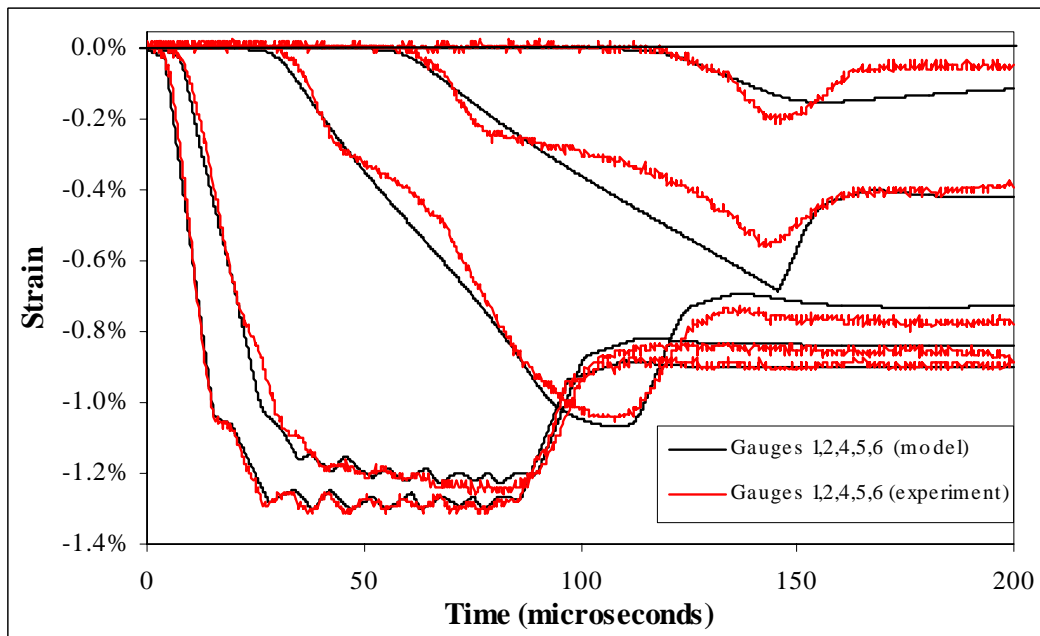


Fig. 24. An adaptive FE analysis of experimental data using deformation plasticity model. The first strain gauge is used to define the boundary condition.

The reason for this is the deviations from linear behavior for small strains. The polynomial model always predicts a linear response until the beginning of the detwinning deformation. However an inspection of Figure 20 shows a smoother nonlinear stress-strain relationship for small strain values.

To verify the hypothesis that the disagreement is due to the initial elastic response of the model an independent numerical simulation of the dynamic experiment was performed. A phenomenological deformation plasticity model was used instead of the constitutive model of Section 1.3. The loading is assumed to have the form of a sixth degree polynomial that curve fits the loading part of the quasi-static hysteresis in Figure 20. The unloading was assumed linear, the slope being the modulus of martensite, $42GPa$, as measured by the quasi-static experiments. Due to the fact that the deformation is mostly detwinning of martensite there is no significant release of latent heat, so the quasi-static hysteresis is very close to the actual material behavior in the dynamic case (Figure 22).

The results of the simulation of the dynamic problem are shown in Figure 24. This time the wave profiles are matched much more closely and the small disagreements can be attributed to measurement errors and effects of lateral inertia not included in the simulation. It should be noted that unlike a constitutive model based on physical principles such an approach will only work for a particular SMA specimen and particular operating temperature. However using a curve fit for the loading part of the hysteresis is sufficient to check whether disagreements between experiments and simulations are indeed due to the constitutive model.

5. Conclusions

The problem of dynamic loading of one-dimensional polycrystalline SMA rods has been explored numerically and recent experimental data was analyzed. FEM simulations were performed for SMAs experiencing both pseudoelastic phase transformation as well as detwinning deformations. Computational solutions were shown to coincide with known analytical results. Nonlinear shock formation and velocities were captured correctly by the FEM simulations. The standard semi-discrete FEM approach for hyperbolic problems was complemented by an adaptive mesh refinement technique. The utilization of the Zienkiewicz-Zhu error indicator lead to an order of magnitude decrease of the computational time. Energy dissipation calculations for both detwinning of martensite and stress-induced phase transformation showed that the strain energy can be reduced by 80-90% which suggests that SMAs can be used effectively as shock-absorption devices.

Further, through analysis of Hopkinson bar data, it is shown that dynamic and quasi-static material response are in excellent agreement, thus validating the assumption of rate independent constitutive response. Through careful calibration of the constitutive model for SMAs the peak strain levels of the Hopkinson bar experiment were accurately predicted. The main drawback of this model is its initial linear response in the case of detwinning and the existence of kinks in the hysteresis curve. Accurate predictions of the entire experimental data were obtained by using a polynomial curve fit of the quasi-static hysteresis of the material. Both the wave timings, shape and peaks were modeled within experimental error.

Theoretical work can also be extended to more realistic 2-D and 3-D geometries. Complicated SMA components and structures can be simulated to better understand the nonlinear wave propagation phenomena as well as the practical aspects of their

energy dissipation capabilities. More refined models which incorporate both detwinning and pseudoelastic deformations simultaneously and also predict accurately the smooth hysteresis of the detwinning deformation will be extremely helpful in further studies of wave propagations. One such model is presented in the next chapter.

CHAPTER III

A THREE PHASE, 3-D CONSTITUTIVE MODEL FOR POLYCRYSTALLINE
SMAS

In this chapter, a thermodynamics based model with three internal variables is formulated for the simultaneous modelling of pseudoelasticity and detwinning of self-accommodated martensite associated with the SME. The model is formulated in three dimensions and is made consistent with a uniaxial phase diagram in stress-temperature space. The chapter begins with new experimental results which demonstrate that twinned and detwinned martensite transform to austenite at different temperatures (Section 1). The phase diagram used in this work is motivated by the new experimental evidence of different transformation temperatures for twinned and detwinned martensite presented in Section 1. It is constructed in Section 2 based on these observations, as well as a careful reexamination of published experimental data on detwinning of twinned martensite and the conversion of twinned martensite to austenite. The 3-D constitutive model is presented in Section 3. A discussion of how to identify the material parameters used in the model from experimental measurements is given in Section 4. Two thermomechanical loading paths which indicate the capabilities of this model are presented in Section 5.

First let us summarize the notation used in the introduction. The three phases will be denoted by A , M^t and M^d for austenite, twinned martensite and detwinned martensite, respectively. The five possible phase transformations will be denoted by $A \rightarrow M^t$, $A \rightarrow M^d$, $M^t \rightarrow A$, $M^d \rightarrow A$ and $M^t \rightarrow M^d$ for austenite to twinned martensite, austenite to detwinned martensite, twinned martensite to austenite, detwinned martensite to austenite and twinned to detwinned martensite, respectively. The detwinning of twinned martensite $M^t \rightarrow M^d$ does not involve phase transfor-

mation and is, in fact, an inelastic deformation process, involving reorientation of martensitic variants. For the sake of simplicity, the collective term transformations will be applied to it whenever the distinction is not important. Finally, the critical start and finish transformation temperatures at zero stress level for the $A \rightarrow M^t$ transformation will be denoted by M_s and M_f , for the $M^t \rightarrow A$ by A_s^t and A_f^t , and for the $M^d \rightarrow A$ by A_s^d and A_f^d . The clarification that these temperatures are for a zero stress level will be omitted, and only the term transformation temperatures will be used.

1. Experimental results on the transformation temperatures of twinned and detwinned martensite to austenite.

In a recent paper, Sakamoto (2002) questioned the tacit assumption made by many researchers that, at zero stress, the transformation temperatures for $M^t \rightarrow A$ and $M^d \rightarrow A$ coincide. He introduced the concept of shape change stress which is an elastic back stress generated in the matrix when twinned martensitic variants are formed. In stress induced martensite this elastic stress field is absent, and a detailed analysis of the magnitude of this elastic back stress with respect to specimen and martensitic plate sizes leads to the conclusion of different transformation temperatures for twinned and detwinned martensite. In this section, mechanical testing combined with calorimetric measurements are used to confirm this idea.

1.1. Setup and experimental procedure

A 2.16mm *NiTi* wire was used in the experiment. Two specimen were annealed at 800 °C for 30min, slowly cooled to 0 °C and then brought to room temperature (22 °C). A DSC test was performed in order to establish the transformation tem-

peratures and characterize the material state after the annealing. It was found that the transformation temperatures were $M_s = 47\text{ }^\circ\text{C}$, $M_f = 3\text{ }^\circ\text{C}$, $A_s^t = 35\text{ }^\circ\text{C}$ and $A_f^t = 76\text{ }^\circ\text{C}$. Since the austenitic start temperature was well above room temperature it was concluded that after the heat treatment the wire was entirely in the M^t state.

Next, the two specimen were mechanically loaded at room temperature in a standard MTS test frame. Due to the initial state of the specimens (M^t) the self-accommodated martensite underwent the detwinning ($M^t \rightarrow M^d$) deformation. Upon completion of the mechanical loading, development of large inelastic strain of about 7% was observed, implying a detwinned material state (M^d). The specimens were then mechanically unloaded, and the inelastic strain was not recovered, indicating that they were still in the M^d state. In order to quantify the amount of inelastic strain due to detwinning of M^t and the amount due to plastic deformations the first specimen was heated. During the process about 5% of the inelastic strain was recovered, indicating it was due to detwinning and the remaining 2% is due to plastic deformations.

The second specimen, immediately after unloading and hence entirely in the M^d state, was further cut in order to obtain a sample from the region subjected to the mechanical loading (that is, in between the grips of the test frame). This sample was subjected to further DSC testing, described below. Care was taken to prepare the DSC sample so that the material state (M^d), achieved at the end of the unloading step, was not altered in the process.

1.2. Successive DSC results

A total of five thermal loading steps were executed by the Perkin-Elmer DSC apparatus. The resulting DSC test is shown in Figure 25. The specimen was first

heated from room temperature to $200\text{ }^{\circ}\text{C}$. The first signs of the forward, $M^d \rightarrow A$ transformation, were observed at $A_s = 82\text{ }^{\circ}\text{C}$, the peak of the transformation was at approximately $96\text{ }^{\circ}\text{C}$ and the transformation ended at approximately $A_f = 108\text{ }^{\circ}\text{C}$. At this point the sample was in the austenitic phase. The sample was then cooled from $200\text{ }^{\circ}\text{C}$ to $-60\text{ }^{\circ}\text{C}$. During the cooling a single peak was observed at approximately $28\text{ }^{\circ}\text{C}$, corresponding to the $A \rightarrow M^t$ transformation. Note that, due to the nature of a DSC test, the sample always remains stress free. The beginning of the reverse transformation indicated $M_s = 47\text{ }^{\circ}\text{C}$ and $M_f = 3\text{ }^{\circ}\text{C}$, which is consistent with the first DSC test performed before the wire was subjected to mechanical loading. A repeatability in the $A \rightarrow M^t$ temperatures was therefore observed.

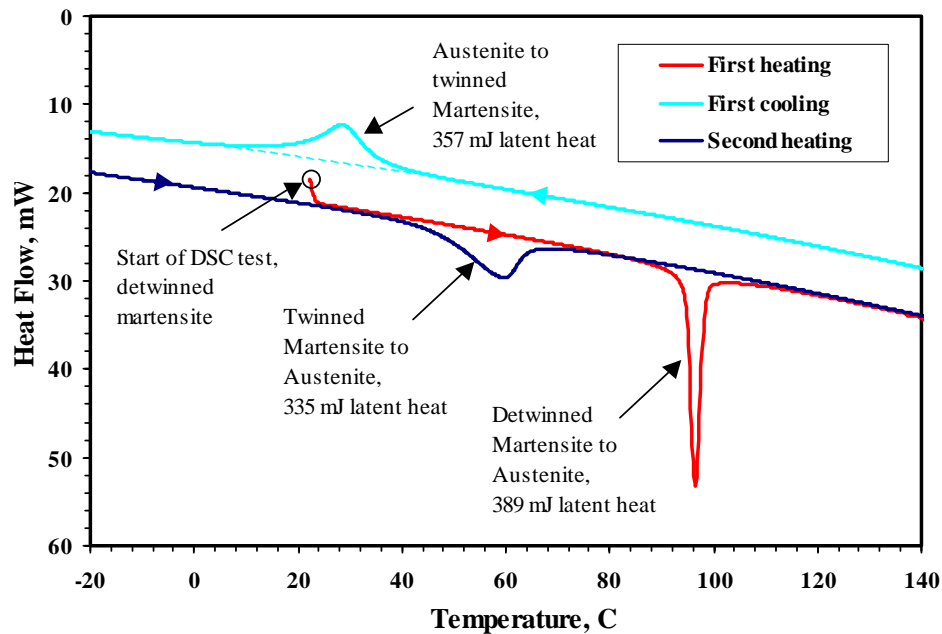


Fig. 25. DSC results after a single mechanical loading of an untrained NiTi wire. The initial state of the wire is M^d and is first heated, revealing the transformation temperatures for $M^d \rightarrow A$. This is followed by cooling step during which the wire undergoes $A \rightarrow M^t$ transformation, followed by a second heating which reveals the transformation temperatures for $M^t \rightarrow A$.

The third thermal loading step was again heating from $-60\text{ }^{\circ}\text{C}$ to $200\text{ }^{\circ}\text{C}$. The transformation temperatures were markedly different from the first heating step: $A_s^t = 35\text{ }^{\circ}\text{C}$, $A_f^t = 76\text{ }^{\circ}\text{C}$ with the peak at $59\text{ }^{\circ}\text{C}$. At the beginning of this step the sample was entirely in the M^t state, therefore the transformation temperatures correspond to the $M^t \rightarrow A$ transformation. Two more loading steps, not shown on Figure 25, were performed. These included an additional cooling and a heating cycle. Due to the stress free state of the SMA, the transformations involved were $A \rightarrow M^t$ and $M^t \rightarrow A$, respectively. Results were close to those from the second (cooling) and third (heating) cycles, respectively, indicating repeatability of the $A \leftrightarrow M^t$ transformation temperatures.

The entire testing procedure was applied to materials with different loading history with similar results. The DSC shown on Figure 26 is for the same type of test (one mechanical loading/unloading at room temperature followed by DSC test), but for a wire trained for stable pseudoelastic response. The training consisted of 20 mechanical loading/unloading cycles at elevated temperature ($80\text{ }^{\circ}\text{C}$) which resulted in a stable pseudoelastic hysteresis loop. The training introduced plastic strains into the material which resulted in partial two-way SME, as indicated by one heating-cooling cycle at zero stress level.

The detwinning test was again performed at room temperature and, upon unloading, some partial reversal of the detwinned martensite into austenite was observed. That is, at the completion of the mechanical test, the material was not in the pure M^d state, but in mixture of M^d and A . This implies that the first heating curve of the successive DSC measurements (Figure 26) may not give information about the beginning of the $M^d \rightarrow A$ transformation (and hence, A_s^d), because there already is some amount of austenite. However the region where the bulk of the transformation occurs and the transformation finish will not be affected, hence the data for the latent

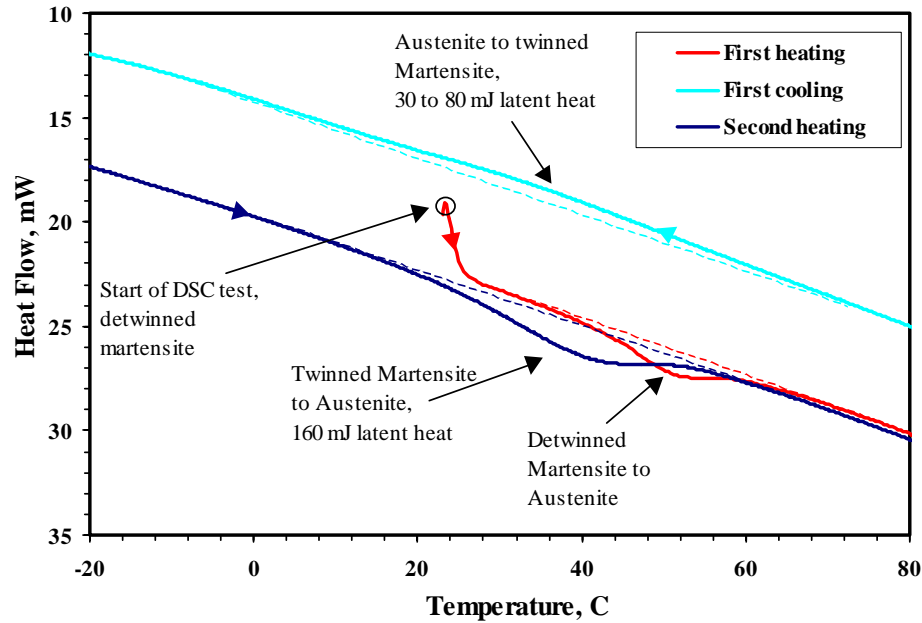
heat peak ($54\text{ }^{\circ}\text{C}$) and the A_f^d temperature ($62\text{ }^{\circ}\text{C}$) is still reliable.

Due to the partial two-way SME, during the cooling step, some amount of M^d has formed. Therefore, during the second heating (third DSC step), the DSC curve will indicate the transformation of a mixture $M^t, M^d \rightarrow A$. Since it was already demonstrated that the twinned martensite transforms at lower temperature than detwinned, the presence of detwinned martensite will tend to broaden the transformation region to the right. As a result, both the actual peak for the $M^t \rightarrow A$ transformation and the A_f^t temperature will be lower, or at most equal to what is observed in the second DSC curve: $40\text{ }^{\circ}\text{C}$ for the peak and $71\text{ }^{\circ}\text{C}$ for A_f^t , respectively. Therefore, by comparing the peaks and finish temperatures for the first heating (first DSC step) and the second heating (third DSC step), the difference in the $M^t \rightarrow A$ and $M^d \rightarrow A$ region is again significant, suggesting separate critical temperatures for the two processes.

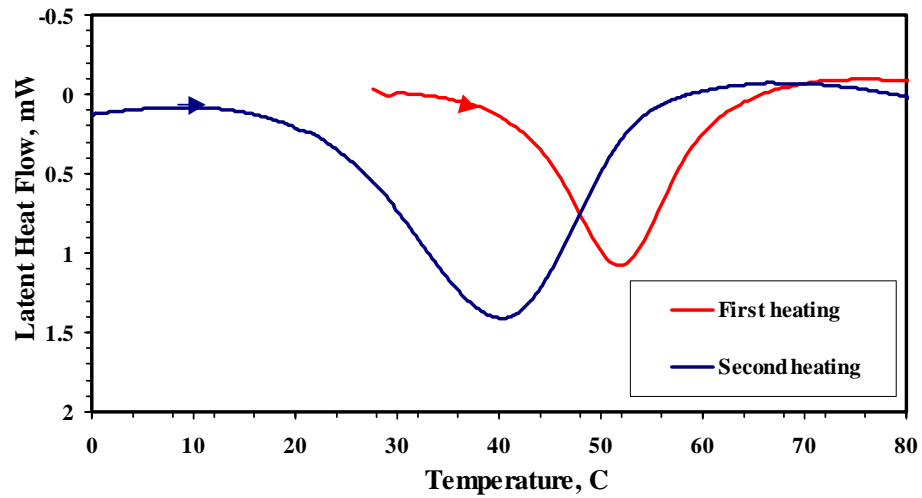
Note that due to the training, the critical transformation temperatures have shifted, compared to the untrained specimen. The peak for the forward ($A \rightarrow M^t$) transformation during the first cooling step (second DSC step) is barely visible, therefore a precise estimate of M_s , M_f and the latent heat during the $A \rightarrow M^t$ transformation is difficult to make.

The simplest conclusion from these experiments is that the $M^t \rightarrow A$ and $M^d \rightarrow A$ transformation temperatures at zero stress are, generally, different. A qualitative explanation for this results can be done as follows: the twinned martensite, requires some energy input to transform back to austenite. The detwinned martensite, also requires this energy output, but in addition it also needs more energy in order to reverse the inelastic strains which are present. Thus, the reverse phase transformation occurs at higher temperatures, compared to twinned martensite.

The theoretical study by Sakamoto (2002) arrives at the same conclusion with the help of microstructural arguments and by analyzing an elastic back stress which



(a) DSC results for wire trained for pseudoelasticity



(b) Detailed view of the reverse transformation region. The thermoelastic heat flow component is subtracted, so only the latent heat flow is shown

Fig. 26. DSC results after a single mechanical loading of a trained NiTi wire. Same test as the one shown on Figure 25, but for a trained NiTi wire. The transformation temperatures have shifted due to the pseudoelastic training, and the peak for the reverse transformation is barely visible. The regions where the reverse transformations $M^t \rightarrow A$ and $M^d \rightarrow A$ occur are still noticeably different.

is different for twinned and detwinned martensite. His study also suggests noticeable size effects. It will therefore be of particular interest to correlate the observed difference of the $M^t \rightarrow A$ and $M^d \rightarrow A$ transformation temperatures with the characteristic size of the SMA (in this case the diameter of the wire). From the modelling point of view, the experimental results, alone, motivate the assumption of different transformation temperatures A_s^t , A_s^d , A_f^t and A_f^d .

2. Modified SMA phase diagram

The phase transformations from austenite to martensite as well as the detwinning of self-accommodated martensite occur due to thermomechanical loading. In the previous section one particular type mechanical path (isothermal) and one purely thermal loading path (heating/cooling at zero applied stress) were used to measure the critical temperatures for phase transformation M_s , M_f , A_s^t , A_s^d , A_f^t and A_f^d . A convenient way of describing more general thermomechanical loading paths leading to the different transformations is a phase diagram in stress-temperature space. A 1-D phase diagram that includes the detwinning process of M^t has been proposed, for example, by Brinson (1993). Such a phase diagram includes the stable domains of A , M^t and M^d in stress-temperature space as well as transformation strips in which the various transformations take place. The transformation strips consist of start and finish lines for each individual transformation. In this section the phase diagram will be presented in uniaxial stress setting. The appropriate modifications, such as effective stress and transformation flow rules, required to generalize it for use in the 3-D model proposed in this work will be discussed in Sections 4. A 3-D numerical implementation will be presented in Chapter IV.

The phase diagram used in this work is given in Figure 27 and can be viewed as

a modification of the one first proposed by Brinson (1993). The regions where the three phases exist in a pure state are shaded and labeled A , M^t and M^d , respectively. The three regions are separated by transformation strips which are labeled according to the transformations ($A \rightarrow M^t$, $A \rightarrow M^d$, $M^t \rightarrow A$, $M^d \rightarrow A$, $M^t \rightarrow M^d$) which takes place. Note that some of these strips overlap and in an overlap region multiple transformations are possible. In the non-shaded region of the phase diagram various mixtures can exist.

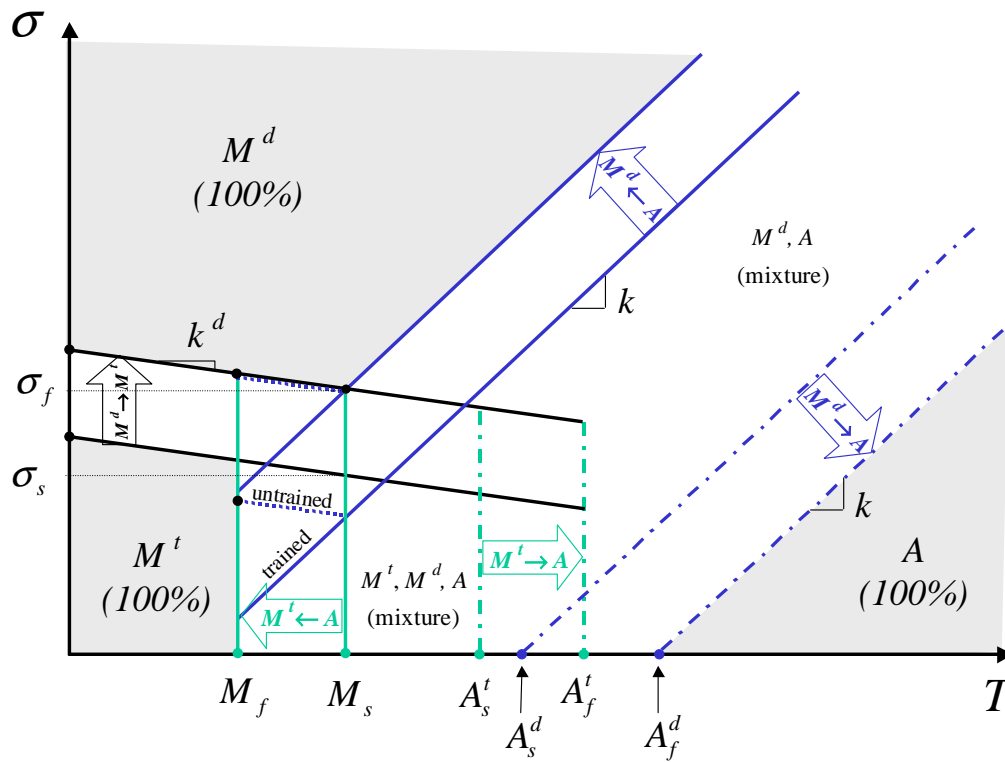


Fig. 27. The SMA phase diagram used in this work. All the three pure phase regions (A , M^d and M^t) are enclosed by transformation strips. The diagram is completely defined by the respective transformation temperatures M_s , M_f , A_s^t , A_f^t , A_s^d , A_f^d , the critical stresses for detwinning σ_s and σ_f and initial slope k .

The critical temperatures for the start and finish of the $A \rightarrow M^t$ transformation are denoted by M_s and M_f . Based on the experimental results of Section 1, the

critical start and finish temperatures at zero stress for the $M^t \rightarrow A$ transformation are denoted by A_s^t and A_f^t . They are assumed different from the corresponding critical temperatures at zero stress for the $M^d \rightarrow A$ transformation which are denoted by A_s^d and A_f^d . The start and finish lines for the forward and reverse transformations $A \leftrightarrow M^t$ are vertical and pass through the critical temperatures M_s , M_f , A_s^t , A_f^t , respectively. The start and finish lines for the reverse strip $M^d \rightarrow A$ pass through the critical temperatures A_s^d and A_f^d and exhibit a temperature dependence, defined by the positive slope k .

The critical uniaxial start and finish stresses at the specific temperature M_s required for detwinning of twinned martensite ($M^t \rightarrow M^d$) are denoted by σ_s and σ_f , respectively. The transformation strip $M^t \rightarrow M^d$ exhibits a mild temperature dependence characterized by a negative slope k^d . The start and finish lines for the forward $A \rightarrow M^d$ transformation exhibits the same temperature dependence as the reverse transformation $M^d \rightarrow A$ and the finish line for $A \rightarrow M^d$ passes through the triple point (M_s, σ_f) .

The modifications of this phase diagram compared to, for example, the one by Brinson (1993) are several. First, and most importantly, based on the experimental results of Section 1, the critical start and finish temperatures at zero stress for the $M^t \rightarrow A$ are assumed different from the corresponding critical temperatures at zero stress for the $M^d \rightarrow A$ transformation.

Secondly, the $M^t \rightarrow M^d$ strip is a single, well-defined strip for the entire temperature range $T < A_f$. The original work of Brinson (1993) assumes that detwinning strip $M^t \rightarrow M^d$ has the same form as the one assumed here for temperatures $T < M_s$ but coincides with that the strip for stress induced martensite $A \rightarrow M^d$ at temperatures $T > M_s$. This question is critically examined in Section 2.2. With the help of the experimental study of Cross et al. (1969), it is shown that a single transformation

strip extending to temperatures as high as A_s^t and possibly to A_f^t , as done in this work, is a more natural assumption.

In the SMA modelling literature which employs phase diagrams (cf., e.g., Bekker and Brinson, 1997; Brinson, 1993; Juhasz et al., 2002; Lagoudas and Shu, 1999; Leclercq and LExcellent, 1996) there is also an ambiguity in the definition of the $A \rightarrow M^d$ low stresses and temperatures ($T < M_s$ and $\sigma < \sigma_s$). Some authors have proposed that it extends to zero-stress level (Bekker and Brinson, 1997), while others (Lagoudas and Shu, 1999) suggest, that in the region $T < M_s$ the dependence on temperature disappears and there is a critical stress below which $A \rightarrow M^d$ does not occur. There are two possible ways of completing it, depending mainly on the training history of the material. In this work, for trained materials, it will be assumed that the $A \rightarrow M^d$ transformation strip extends all the way to zero stress level. For untrained SMA materials the respective lines in Figure 27 are labeled accordingly. Also, the start and finish lines (below M_s) for the untrained case are also marked with a dashed line.

A related question for untrained SMA materials is the existence of a triple point for the $A \rightarrow M^d$, $A \rightarrow M^t$, $M^t \rightarrow M^d$. In this work, it is first assumed to exist and second that it is located at the intersection of the finish lines for $A \rightarrow M^d$, $M^t \rightarrow M^d$ with the start line for $A \rightarrow M^t$. One implication is that the critical start stress for detwinning σ_s at $T = M_s$ is not necessarily the same as the minimal stress below which $A \rightarrow M^d$ does not occur. This question is discussed in Section 2.3.

There is also an ambiguity on the shape of the reverse $M^t \rightarrow A$ strip. In the original work of Brinson (1993) it is assumed to coincide with the $M^d \rightarrow A$ strip while other authors Lagoudas and Shu (1999); Leclercq and LExcellent (1996) have used a vertical $M^t \rightarrow A$ strip, which is independent of stress. An argument can be made (Section 2.1) that the later is more natural case.

The remainder of this section presents a detailed description of the proposed extensions and modifications of the phase diagram of Figure 27 in comparison with the earlier literature on the subject.

2.1. Austenite to martensite ($A \leftrightarrow M^t$, $A \leftrightarrow M^d$).

An early observation in quasi-static isothermal loading tests was that the transformation surfaces for $A \leftrightarrow M^d$ exhibit a strong temperature dependence (Cross et al., 1969; Jackson et al., 1972; Otsuka and Wayman, 1999; Wayman, 1983). These and many other experimental results show that the critical transformation stress required for initiation and completion of both the $A \rightarrow M^d$ and $M^d \rightarrow A$ forward and reverse transformations increase, more or less linearly, with increase in temperature. The reason for this dependence on temperature is the development of transformation strain during the transformation and the associated work expended by the SMA. The theoretical derivation of the precise functional dependence of the critical transformation stress for detwinning is based on a Clausius-Clapeyron relation (Wollants et al., 1979). After some simplifying assumptions such as equal stiffness and thermal expansion coefficient of austenite and martensite, a linear dependence on temperature is obtained (Otsuka and Wayman, 1999; Wollants et al., 1979). This has been observed consistently by many experimentalists ever since the work of Cross et al. (1969). Virtually any constitutive model for pseudoelastic SMA response, including the current work, takes this into account.

Unlike the $A \leftrightarrow M^d$ transitions, the phase transformation from A to M^t does not involve generation of macroscopic strains. At zero stress level, the $A \rightarrow M^t$ phase transformation begins when a critical temperature M_s is reached and is completed when a second, and lower, critical temperature M_f is reached. Due to the lack of transformation strain, a Clausius-Clapeyron argument suggests that there is no

dependence of the critical temperatures M_s and M_f on stress. As a consequence one can expect that the transformation strip $A \rightarrow M^t$ is nearly vertical when plotted in the stress-temperature space (Figure 27). This fact has been used in most models that take into account the separate development of twinned and detwinned martensite (cf., e.g., Bekker and Brinson, 1997; Brinson, 1993; Juhasz et al., 2002; Lagoudas and Shu, 1999; Leclercq and Lexcellent, 1996). There is however disagreement on what the shape of the reverse transformation strip $M^t \rightarrow A$ should be. Bekker and Brinson (1997); Brinson (1993) assume the same stress-temperature dependence as for the $M^d \rightarrow A$ transformation. Others (Juhasz et al., 2002; Lagoudas and Shu, 1999; Leclercq and Lexcellent, 1996) take the $M^t \rightarrow A$ strip to be stress independent

There are surprisingly few experiments reported in the literature, which aim at determining the shape of the $A \leftrightarrow M^t$ strips. It should be noted that, due to lack of macroscopically observable mechanical quantities, such as inelastic strains, it is very difficult to experimentally detect the formation of twinned martensite under applied stress. Differential scanning calorimeter measurements (DSC), which are usually employed for revealing the transformation temperatures at zero stress level cannot be directly used under applied stress. The two direct methods of measuring the progress of martensitic transformation under applied load that have been used by researchers are electrical resistivity measurements (cf., e.g., Kotil et al., 2003; Šittner et al., 2000) and in-situ neutron diffraction measurements (cf., e.g., Šittner et al., 2003). In both cases sophisticated testing procedures in precisely controlled thermal environment in a MTS-type testing frame are required. The focus of these and other direct measurement studies however was not the stress dependence of the critical temperatures for the $A \leftrightarrow M^t$ transformation.

An alternative indirect method, used specifically for determining the $M^t, M^d \rightarrow A$ transformation temperatures at different pre-strain levels during heating and cool-

ing cycles has recently been performed by Tsoi et al. (2003). The experiment is done by first loading an SMA wire and embedding it in a epoxy matrix. After the epoxy has cured, it keeps the SMA deformed without the need for external apparatus and composite can further be cut into small enough specimen, suitable for DSC measurements. The tests included pre-strain levels low enough that only $M^t \rightarrow A$ transformation can be expected during heating. While the DSC results are difficult to interpret conclusively, it can be inferred that the $M^t \rightarrow A$ temperatures do not depend on applied stress.

Due to the lack of inelastic strains associated with the $M^t \rightarrow A$ transformation and the experimental indications of Tsoi et al. (2003), in this work it will be assumed that both $M^t \rightarrow A$ and $A \rightarrow M^t$ are stress independent. In Section 5.1 a different indirect experimental method, based on the different stiffness of the pure martensitic and austenitic phases will be proposed.

2.2. Detwinning of self accommodated martensite ($M^t \rightarrow M^d$)

The three pure phases regions (A , M^t and M^d) are separated by transformation strips that indicate which transformations occur ($A \rightarrow M^d$, $A \rightarrow M^t$, etc). In the original phase diagram of Brinson (1993) the transformation strip $M^t \rightarrow M^d$ is not defined at temperatures above $T > M_s$. If the initial conditions are such that M^t is not present and once it is produced, the temperature is never increased beyond M_s , this will not cause problems. This is the case with a major class of SME paths, where all the M^t is depleted via the $M^t \rightarrow M^d$ deformation before the temperature is increased above M_s . Since these types of SME loading paths are quite important in characterization and testing of SMAs, the possibility that M^t may be present at temperatures in the range $M_s < T < A_f^t$ (for example, by detwinning only part of the M^t) has generally been overlooked. Brinson (1993) have assumed for simplicity

that the transformation strip for $M^t \rightarrow M^d$ coincides with the $A \rightarrow M^d$ one in this temperature range. This assumption creates the inconvenience of having a concave transformation surface in stress-temperature space. It can also be argued that the detwinning of martensite is an inelastic deformation process and does not involve change in the crystal lattice. Therefore the temperature dependence of the detwinning surface should not change drastically as suggested, that is, from slightly decreasing yield stress as the temperature is raised in the range $T < M_s$ to rapidly increasing with increase of temperature for $M_s < T < A_f^t$.

More importantly though, the assumption that the $M^t \rightarrow M^d$ strip coincides with $A \rightarrow M^d$ for temperatures higher than M_s does not seem to be supported by experimental results. A careful review of the pioneering work of Cross et al. (1969) suggests that it extends to temperatures higher than M_s . The reader is referred specifically to Figure 16 on page 26 of (Cross et al., 1969), which reports two sets of experiments. In both cases the material is loaded mechanically, under isothermal conditions at several different temperatures. The difference is that prior to the mechanical loading, in the first set, the material is cooled from high temperature and once the prescribed temperature is reached, it is fixed and the SMA is mechanically loaded. In the second set, the material is heated from low temperature, and then loaded. The initial yield stress is recorded in both cases. A look at the transformation temperatures, reported by the authors, shows, that for the first set of experiments the initial material state is A , while for the second it is M^t . The latter implies that the initial yield stress measured in the second set corresponds to the beginning of the $M^t \rightarrow M^d$ deformation over the entire range $T \leq A_s^t$. The results in the range $A_s^t \leq T \leq A_f^t$ cannot be easily interpreted since in this range the material before loading is a mixture of A and M^t . The observed values for the critical stress for detwinning exhibit only very slight dependence on temperature, decreasing slowly

as temperature is increased. Since for the first experimental set the initial material state is A , then a transformation surface for $M^t \rightarrow M^d$ can be inferred from the yield stress results only in the range $T \leq M_f$. Observe that, the measured yield stress in the range $T \leq M_f$ for both sets of experiments is the same, which means that the experimentation results are consistent.

Based upon this analysis, it is assumed in this work, that the shape of the $M^t \rightarrow M^d$ has the same dependence on temperature, both for temperatures below and above M_s (Figure 27). Note, that the region of the phase diagram covered both by the $M^t \rightarrow M^d$ and $M^t \rightarrow A$ (to be discussed next) completely surround the region where pure M^t can exist, therefore there is no possibility that a loading path may lead to existence of M^t at high temperature or high-stress regions of the phase diagram.

2.3. Combined austenite to detwinned martensite at low stresses

As was explained in the previous section, it is difficult to determine experimentally when the transformation to twinned martensite is occurring. Therefore another outstanding question, for which there is little experimental information, is what is the shape of the $A \rightarrow M^d$ surface at low stress $\sigma < \sigma_s$ and temperatures. In this region of the phase diagram it can be expected that both $A \rightarrow M^d$ and $A \rightarrow M^t$ occur. Note that the $A \rightarrow M^d$ is measured experimentally by observing the critical transformation stress required for the $A \rightarrow M^d$ transformation.

Bo and Lagoudas (1999a); Miller (2000) have measured the development of transformation strain during isobaric heating and cooling of annealed NiTiCu wires at different, constant, stress levels. Such a test can be represented by a horizontal line on the phase diagram and allows to determine the location of the $A \rightarrow M^d$ (during cooling) and $M^d \rightarrow A$ (during heating) transformation surfaces. The results for

untrained specimen suggest that there $A \rightarrow M^d$ does not take place at stress levels below 40MPa . They therefore argue that there is a critical stress level, below which detwinned martensite cannot form. This has usually been incorporated into SMA models (Brinson, 1993; Lagoudas and Shu, 1999) by assuming the $M^d \rightarrow A$ surface is independent of temperature below $T < M_s$.

This SMA behavior at low temperatures and stresses however is heavily influenced by the material composition, manufacturing process (e.g. cold work), heat treatments, etc. If a wire is trained for pseudoelastic operation, then development of transformation strain is observed even at zero stress level, which implies that the $M^d \rightarrow A$ surface should extend to zero stress. In order to take into account both types of behavior, the model developed here will include both the capability to proceed with the $A \rightarrow M^d$ transformation at arbitrary stress level and the possibility of a critical stress below which production of M^d does not happen. In the first case, the $A \rightarrow M^d$ transformation strip would reach zero stress (dotted line in Figure 27), while in the second, it becomes horizontal at $T < M_s$.

3. Description of the constitutive theory

3.1. Kinematic assumptions

In order to simplify the presentation, the term "transformation" will be used to denote both the phase transformation from austenite to twinned and detwinned martensite as well as the detwinning deformation of self-accommodated martensite. We start with the volume fractions c_i , $i = 1, 2, 3$ of the self accommodated martensite M^t , stress-induced martensite M^d and austenite A . The volume fractions are subject to the obvious constraints:

$$c_1 + c_2 + c_3 = 1, \quad (3.1)$$

$$0 \leq c_i \leq 1, \quad \text{for } i = 1, 2, 3. \quad (3.2)$$

While the state of the material is represented completely by the three volume fractions c_i , it is also useful to know how this state was achieved. To do this, the volume fractions ξ_1 of M^t produced from A , the volume fraction ξ_2 of M^d produced from A , and the volume fraction ξ_3 of M^d , produced from M^t , are introduced:

$$c_1 = c_{10} + \xi_1 - \xi_3, \quad (3.3)$$

$$c_2 = c_{20} + \xi_2 + \xi_3, \quad (3.4)$$

$$c_3 = c_{30} - \xi_1 - \xi_2, \quad (3.5)$$

where c_{i0} , $i = 1, 2, 3$ are the initial volume fractions of the three phases, subject to the constraint

$$c_{10} + c_{20} + c_{30} = 1.$$

These two representations of the phase state of the material are schematically portrayed in Figure 28. The two *phase* transformation $A \leftrightarrow M^t$ and $A \leftrightarrow M^d$ can proceed both ways, hence, $\dot{\xi}_1$, $\dot{\xi}_2$ can take arbitrary real values. The detwinning deformation $M^t \rightarrow M^d$ however is irreversible, therefore $\dot{\xi}_3 \geq 0$. Note that the constraint (3.1) is automatically satisfied.

In this work, the internal variables that describe the phase state of the material are selected to be:

$$\xi_i, \quad i = 1, 2, 3, \quad (3.6)$$

connected to the volume fractions c_i , $i = 1, 2, 3$ by equations (3.3)-(3.5). This is done, apart from the convenience of having equation (3.1) always satisfied, for one additional reasons. There are no general principles which prohibit that three simultaneous transformations occur. For example, it may happen, that upon specific thermal cooling and mechanical loading in the neighborhood of the triple point (M_s, σ_s) in

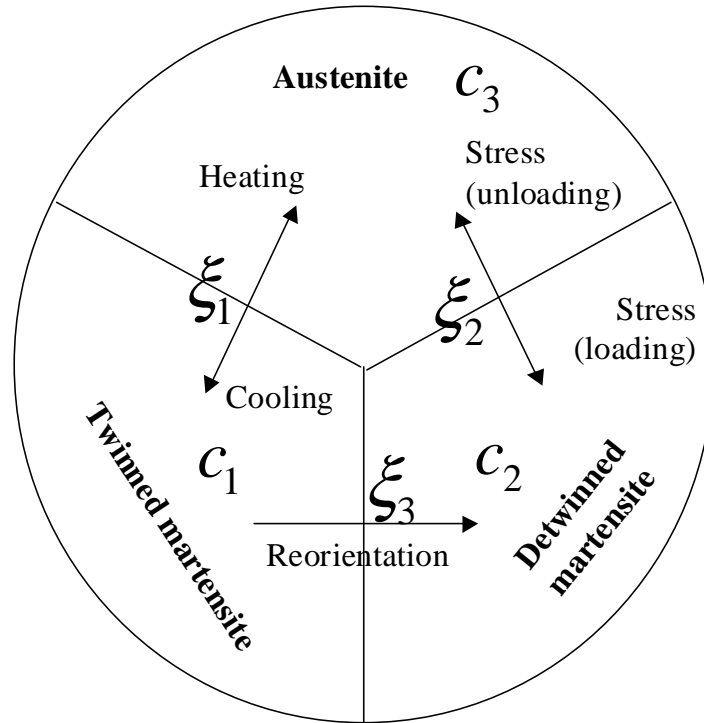


Fig. 28. Schematic of the three phases and the possible transitions between them.

the phase diagram, one has simultaneous $A \rightarrow M^t$, $A \rightarrow M^d$ and at the same time, some of the produced martensite is immediately detwinning due to the stress increase $M^t \rightarrow M^d$. While some authors (Juhász et al., 2002) argue that such situations should be forbidden, this in itself is an additional constitutive assumption.

The constitutive theory is formulated for small strains. Let $\boldsymbol{\varepsilon}$ be the total strain tensor, given by

$$\boldsymbol{\varepsilon} = \frac{1}{2} (\nabla \mathbf{u} + \nabla \mathbf{u}^T), \quad (3.7)$$

where \mathbf{u} is the displacement. Further, it is assumed that the strain can be decomposed additively into elastic $\boldsymbol{\varepsilon}^{el}$, thermal $\boldsymbol{\varepsilon}^{th}$ and inelastic strain $\boldsymbol{\varepsilon}^{in}$ components:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{el} + \boldsymbol{\varepsilon}^{th} + \boldsymbol{\varepsilon}^{in}.$$

The inelastic strain $\boldsymbol{\varepsilon}^{in}$ is produced during the forward and reverse stress induced phase transformation ($\dot{\xi}_2 \neq 0$) and during the detwinning deformation ($\dot{\xi}_3 > 0$). Consequently, it can be further decomposed into:

$$\boldsymbol{\varepsilon}^{in} = \boldsymbol{\varepsilon}^t + \boldsymbol{\varepsilon}^d, \quad (3.8)$$

where $\boldsymbol{\varepsilon}^t$ is the stress induced transformation strain (produced during $A \rightarrow M^d$ transformation) and $\boldsymbol{\varepsilon}^d$ is the inelastic strain generated during detwinning ($M^t \rightarrow M^d$).

Finally, it is assumed that the transformation and detwinning strains obey the following two transformation/detwinning flow rules:

$$\dot{\boldsymbol{\varepsilon}}^t = \boldsymbol{\Lambda}^t \dot{\xi}_2, \quad (3.9)$$

$$\dot{\boldsymbol{\varepsilon}}^d = \boldsymbol{\Lambda}^d \dot{\xi}_3, \quad (3.10)$$

In general, the flow tensors $\boldsymbol{\Lambda}^t$ and $\boldsymbol{\Lambda}^d$ are different (Lagoudas and Shu, 1999). The specific form of the transformation tensors will be discussed when the transformation surfaces are introduced in Section 3.4.

The last assumptions (3.9) and (3.10) allow the formulation of the constitutive theory in terms of ξ_i , $i = 1, 2, 3$ as the only internal variables. It is convenient to use vector notation $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)^T$ for the internal variables.

3.2. Energy balance and entropy inequality

The purpose of this section is to introduce the balance of energy and the second law of thermodynamics in a general setting. This will make clear what restrictions thermodynamics places on the SMA constitutive model.

Given an arbitrary part P of the body, the power \mathcal{W}_{int} expended internally in P

is given by

$$\mathcal{W}_{int} = \int_P \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} dV \quad (3.11)$$

where $\boldsymbol{\sigma}$ is the Cauchy stress. The external power \mathcal{W}_{ext} supplied to P is

$$\mathcal{W}_{ext} = \int_{\partial P} (\boldsymbol{\sigma} \mathbf{n}) \cdot \mathbf{v} dA + \int_P \mathbf{b} \cdot \mathbf{v} dV, \quad (3.12)$$

where \mathbf{v} is the velocity and \mathbf{n} is the outward normal to the surface the boundary ∂P .

Let $\mathcal{U}(P)$ denote the internal energy of P and r be a distributed heat source term.

The balance of energy states (Malvern, 1969):

$$\frac{d}{dt} \mathcal{U}(P) = - \int_{\partial P} \mathbf{q} \cdot \mathbf{n} dA + \int_P \rho r dV + \mathcal{W}_{ext}(P). \quad (3.13)$$

The density ρ of the SMA is assumed constant for all phases. Since, $\mathcal{W}_{int} = \mathcal{W}_{ext}$,

P is arbitrary, and with the help of Gauss-Ostrogradsky's theorem, the last equation

(3.13) can be written in local form:

$$\rho \dot{u} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} + \rho r - \nabla \cdot \mathbf{q}, \quad (3.14)$$

where u is the specific internal energy per unit mass.

Similarly, the second law of thermodynamics reads

$$\frac{d}{dt} \int_P \rho s dV \geq - \int_{\partial P} \frac{\mathbf{q} \cdot \mathbf{n}}{T} dA + \int_P \rho \frac{r}{T} dV, \quad (3.15)$$

where T is the temperature, and s is the specific entropy per unit mass. After applying

Gauss-Ostrogradsky's theorem and using the fact that P is arbitrary, the local form

of the entropy inequality is obtained:

$$\rho \dot{s} \geq - \frac{\nabla \cdot \mathbf{q}}{T} + \frac{\mathbf{q} \cdot \nabla T}{T^2} + \rho \frac{r}{T}. \quad (3.16)$$

The constitutive theory for the SMA is formulated using a Gibbs free energy G ,

which is related to the entropy s and internal energy u by a Legendre transformation (cf., e.g., Malvern, 1969, page 262):

$$G = u - sT - \frac{1}{\rho} \boldsymbol{\varepsilon} : \boldsymbol{\sigma}. \quad (3.17)$$

In order to express the entropy inequality (3.16) in terms of the Gibbs energy, the last equation is differentiated with respect to time

$$\dot{G} = \dot{u} - \dot{s}T - s\dot{T} - \frac{1}{\rho} \dot{\boldsymbol{\sigma}} : \boldsymbol{\varepsilon} - \frac{1}{\rho} \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}, \quad (3.18)$$

and combined with (3.14), (3.16) to obtain an equivalent form of the entropy inequality:

$$\rho \dot{G} + \dot{\boldsymbol{\sigma}} : \boldsymbol{\varepsilon} + \rho s \dot{T} + \frac{\mathbf{q} \cdot \nabla T}{T} \leq 0. \quad (3.19)$$

The restrictions this last inequality places on the constitutive theory are established in the next section.

3.3. Constitutive assumptions for the SMA material

A general constitutive equations of the form

$$G = G(\boldsymbol{\sigma}, T, \nabla T, \boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) \quad (3.20)$$

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}(\boldsymbol{\sigma}, T, \nabla T, \boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) \quad (3.21)$$

$$s = s(\boldsymbol{\sigma}, T, \nabla T, \boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) \quad (3.22)$$

$$\mathbf{q} = \mathbf{q}(\boldsymbol{\sigma}, T, \nabla T, \boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) \quad (3.23)$$

is considered. The independent state variables are stress $\boldsymbol{\sigma}$, temperature T , temperature gradient ∇T ¹, the three production volume fractions $\boldsymbol{\xi}$ given by (3.6) and their

¹Recall that the temperature gradient ∇T is not necessarily proportional to the heat flux \mathbf{q} .

rates of change $\dot{\boldsymbol{\xi}}$.

A basic assumption of continuum mechanics is that every process related through the constitutive equations (3.20)-(3.23) must satisfy the entropy inequality (3.19). Assuming sufficient smoothness of all involved quantities, the total time derivative of G can be written as:

$$\dot{G} = \dot{\boldsymbol{\sigma}} \cdot \frac{\partial G}{\partial \boldsymbol{\sigma}} + \dot{T} \frac{\partial G}{\partial T} + (\nabla T) \cdot \frac{\partial G}{\partial \nabla T} + \dot{\boldsymbol{\xi}} \cdot \frac{\partial G}{\partial \boldsymbol{\xi}} + \ddot{\boldsymbol{\xi}} \cdot \frac{\partial G}{\partial \dot{\boldsymbol{\xi}}} \quad (3.24)$$

and substituting it, into (3.19), the inequality becomes:

$$\begin{aligned} \left(\boldsymbol{\varepsilon} + \rho \frac{\partial G}{\partial \boldsymbol{\sigma}} \right) : \dot{\boldsymbol{\sigma}} + \rho \left(s + \frac{\partial G}{\partial T} \right) \dot{T} + \rho \frac{\partial G}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} \\ + \rho \frac{\partial G}{\partial \dot{\boldsymbol{\xi}}} \cdot \ddot{\boldsymbol{\xi}} + \rho \frac{\partial G}{\partial \nabla T} \cdot (\nabla T) + \frac{\mathbf{q} \cdot \nabla T}{T} \leq 0 \end{aligned} \quad (3.25)$$

The last inequality holds for all fields $\boldsymbol{\sigma}(\mathbf{p}, t)$, $T(\mathbf{p}, t)$ $\boldsymbol{\xi}(\mathbf{x}, t)$. Observe, that $\dot{\boldsymbol{\sigma}}$, \dot{T} , $\dot{\boldsymbol{\xi}}$ and (∇T) enter the above inequality linearly. That is, the functions defined in (3.20)-(3.23) do not depend $\dot{\boldsymbol{\sigma}}$, \dot{T} , $\dot{\boldsymbol{\xi}}$ and (∇T) ; and consequently, all other quantities in (3.25) also do not depend on them. Therefore,

$$\frac{\partial G}{\partial \dot{\boldsymbol{\xi}}} = 0, \quad \frac{\partial G}{\partial \nabla T} = 0, \quad (3.26)$$

and thus, provided G is a smooth function of $\dot{\boldsymbol{\xi}}$ and ∇T , it does not depend on them:

$$G = G(\boldsymbol{\sigma}, T, \boldsymbol{\xi}). \quad (3.27)$$

Also, the following two constitutive relations are established:

$$\boldsymbol{\varepsilon} = -\rho \frac{\partial G}{\partial \boldsymbol{\sigma}}, \quad (3.28)$$

$$s = -\frac{\partial G}{\partial T}, \quad (3.29)$$

and the reduced entropy inequality holds:

$$\rho \frac{\partial G}{\partial \boldsymbol{\xi}} \dot{\boldsymbol{\xi}} + \frac{\nabla \cdot \mathbf{q}}{T} \leq 0 \quad (3.30)$$

Further, it will be assumed that the SMA material is strongly dissipative, that is the two terms in the last inequality (3.30) holds separately:

$$\rho \frac{\partial G}{\partial \boldsymbol{\xi}} \dot{\boldsymbol{\xi}} \leq 0, \quad (3.31)$$

$$\frac{\mathbf{q} \cdot \nabla T}{T} \leq 0. \quad (3.32)$$

In light of equation (3.27), observe, that the last assumption is equivalent to assuming that \mathbf{q} is independent of $\dot{\boldsymbol{\xi}}$:

$$\mathbf{q} = \mathbf{q}(\boldsymbol{\sigma}, T, \nabla T, \boldsymbol{\xi}). \quad (3.33)$$

With this thermodynamic restrictions in mind, the constitutive theory is completed in the next two sections.

3.3.1. Gibbs free energy for a polycrystalline SMA

The following form of the Gibbs energy, based on the works of Bo and Lagoudas (1999a); Lagoudas and Shu (1999) is assumed:

$$G = (c_1 + c_2)G^M(\boldsymbol{\sigma}, T) + c_3G^A(\boldsymbol{\sigma}, T) + G^{\text{mix}}(\boldsymbol{\sigma}, T, \xi_1, \xi_2, \xi_3), \quad (3.34)$$

where G^M is the thermoelastic free energy of both martensitic phases (twinned and detwinned), G^A is the thermoelastic component of the free energy of austenite, and G^{mix} is the free energy of mixing, which is responsible for the transformation behavior

of the SMA. The two thermoelastic components are given by:

$$G^A(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) = -\frac{1}{2\rho}\boldsymbol{\sigma} : \boldsymbol{S}^A : \boldsymbol{\sigma} - \frac{1}{\rho}\boldsymbol{\alpha}^A : \boldsymbol{\sigma}(T - T_0) - \frac{1}{\rho}\boldsymbol{\sigma} : \boldsymbol{\varepsilon}^{in} \\ + c^A \left[(T - T_0) - T \ln \left(\frac{T}{T_0} \right) \right] - s_0^A T + u_0^A \quad (3.35)$$

$$G^M(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) = -\frac{1}{2\rho}\boldsymbol{\sigma} : \boldsymbol{S}^M : \boldsymbol{\sigma} - \frac{1}{\rho}\boldsymbol{\alpha}^M : \boldsymbol{\sigma}(T - T_0) - \frac{1}{\rho}\boldsymbol{\sigma} : \boldsymbol{\varepsilon}^{in} \\ + c^M \left[(T - T_0) - T \ln \left(\frac{T}{T_0} \right) \right] - s_0^M T + u_0^M \quad (3.36)$$

\boldsymbol{S}^i , $\boldsymbol{\alpha}^i$, c^i , s_0^i and u_0^i are the compliance tensor, thermal expansion coefficient tensor, specific heat, specific entropy and the specific internal energy at the reference state of the individual phases with the superscript $i = A$ for austenitic and $i = M$ for martensite, respectively. It is assumed that the material properties of the two martensitic phases are the same. Note that this assumption, and correspondingly, the selection of the same energy for M^t and M^d is guided by the fact that from a metallurgical point of view, the two phases are indistinguishable. However, as was discussed in the beginning of this section, it is the macroscopic mechanical behavior of twinned and detwinned martensite that is different, which is reflected in the kinematic considerations of Section 3.1.

The two free energies (3.35) and (3.36) depend on $\boldsymbol{\xi}$ due to the presence of the inelastic strain $\boldsymbol{\varepsilon}^{in}$, which is a path-dependent function of $\boldsymbol{\xi}$:

$$\boldsymbol{\varepsilon}^{in} = \int_0^t \left(\boldsymbol{\Lambda}^t \dot{\xi}_2 + \boldsymbol{\Lambda}^d \dot{\xi}_3 \right) d\tau = \int_0^{\xi_2} \boldsymbol{\Lambda}^t d\eta + \int_0^{\xi_3} \boldsymbol{\Lambda}^d d\eta$$

Upon substituting equations (3.35) and (3.36) into (3.34) and using the constraint

(3.1), the following expression is obtained for the free energy:

$$\begin{aligned}
G(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) = & -\frac{1}{2\rho} \boldsymbol{\sigma} : \boldsymbol{\mathcal{S}}(c_1 + c_2) : \boldsymbol{\sigma} - \frac{1}{\rho} \boldsymbol{\sigma} : [\boldsymbol{\alpha}(c_1 + c_2)(T - T_0) + \boldsymbol{\varepsilon}^{in}] \\
& + c(c_1 + c_2) \left[(T - T_0) - T \ln \left(\frac{T}{T_0} \right) \right] \\
& - s_0(c_1 + c_2)T + u_0(c_1 + c_2) + G^{\text{mix}}, \tag{3.37}
\end{aligned}$$

where $\boldsymbol{\mathcal{S}}(c_1 + c_2)$, $\boldsymbol{\alpha}(c_1 + c_2)$, $c(c_1 + c_2)$, $s_0(c_1 + c_2)$ and $u_0(c_1 + c_2)$ are the effective compliance tensor, thermal expansion coefficient tensor, specific heat, specific entropy and the specific internal energy at the reference state, respectively. The above effective material properties are calculated in terms of the total martensitic volume fraction $c_1 + c_2$ using the rule of mixtures as

$$\boldsymbol{\mathcal{S}}(c_1 + c_2) = \boldsymbol{\mathcal{S}}^A + (c_1 + c_2)(\boldsymbol{\mathcal{S}}^M - \boldsymbol{\mathcal{S}}^A) = \boldsymbol{\mathcal{S}}^A + (c_1 + c_2)\Delta\boldsymbol{\mathcal{S}}, \tag{3.38a}$$

$$\boldsymbol{\alpha}(c_1 + c_2) = \boldsymbol{\alpha}^A + (c_1 + c_2)(\boldsymbol{\alpha}^M - \boldsymbol{\alpha}^A) = \boldsymbol{\alpha}^A + (c_1 + c_2)\Delta\boldsymbol{\alpha}, \tag{3.38b}$$

$$c(c_1 + c_2) = c^A + (c_1 + c_2)(c^M - c^A) = c^A + (c_1 + c_2)\Delta c, \tag{3.38c}$$

$$s_0(c_1 + c_2) = s_0^A + (c_1 + c_2)(s_0^M - s_0^A) = s_0^A + (c_1 + c_2)\Delta s_0, \tag{3.38d}$$

$$u_0(c_1 + c_2) = u_0^A + (c_1 + c_2)(u_0^M - u_0^A) = u_0^A + (c_1 + c_2)\Delta u_0. \tag{3.38e}$$

A detailed discussion the functional form (3.37) for the free energy and the resulting rule of mixtures (3.38), based on micromechanical averaging over a representative volume element of the polycrystalline SMA can be found in (Bo and Lagoudas, 1999a).

Before introducing the energy of mixing, consider the real functions

$$f_i \left(\boldsymbol{\xi}; \text{sgn} \left(\dot{\xi}_i \right) \right), \quad i = 1, 2, \tag{3.39}$$

are smooth functions of $\boldsymbol{\xi}$ and depend on the sign of the rate of ξ_i :

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ \text{undefined} & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

For fixed $\boldsymbol{\xi}$, f_i has one well defined constant value for all $\dot{\xi}_i$, such that $\dot{\xi}_i > 0$, and another, possibly different, value when $\dot{\xi}_i < 0$. Now, the value of f_i when $\dot{\xi}_i = 0$ is undefined, however the following integral

$$\int_0^t f_i(\boldsymbol{\xi}; \text{sgn}(\dot{\xi}_i)) \dot{\xi}_i(\tau) d\tau \quad (3.40)$$

is a well defined function of t . Note that formally,

$$\int_0^t f_i(\boldsymbol{\xi}; \text{sgn}(\dot{\xi}_i)) \dot{\xi}_i(\tau) d\tau = \int_0^{\xi_i} f_1(\dots, \eta, \dots; \text{sgn}(\dot{\xi}_i)) d\eta$$

and therefore the derivative

$$\frac{\partial}{\partial \xi_i} \left[\int_0^t f_i(\boldsymbol{\xi}; \text{sgn}(\dot{\xi}_i)) \dot{\xi}_i(\tau) d\tau \right] = f_i(\boldsymbol{\xi}; \text{sgn}(\dot{\xi}_i))$$

has well defined values for $\dot{\xi}_i \neq 0$. On the other hand, the integral is a piecewise constant function of $\dot{\xi}_i$, possibly discontinuous at $\dot{\xi}_i = 0$. Its derivative with respect to $\dot{\xi}_i$ is, however, identically zero for all values of $\dot{\xi}_i$. Consequently, if a term of the form (3.40) is included in the Gibbs free energy, it will be consistent with the thermodynamic constraint (3.27).

The reason for this construction is to have a smooth in $\boldsymbol{\xi}$, (path-dependent) free energy whose partial derivatives with respect to $\boldsymbol{\xi}$ can have two different values, depending on whether $\dot{\xi}_i$, $i = 1, 2$ is positive or negative. This is important, because many SMAs exhibit different hardening behavior during loading and unloading, and the mixing term in the free energy is responsible precisely for this the transformation

hardening (see the next section). Therefore, from this point onwards the dependence on $\dot{\xi}_i$ of the functions f_i will be suppressed and the energy of mixing is assumed to have the form:

$$\begin{aligned} G^{\text{mix}} &= \frac{1}{\rho} \int_0^t \left(f_1(\boldsymbol{\xi}) \dot{\xi}_1(\tau) + f_2(\boldsymbol{\xi}) \dot{\xi}_2(\tau) + f_3(\boldsymbol{\xi}) \dot{\xi}_3(\tau) \right) d\tau \\ &= \frac{1}{\rho} \int_0^{\xi_1} f_1(\eta, \xi_2, \xi_3) d\eta + \frac{1}{\rho} \int_0^{\xi_2} f_2(\xi_1, \eta, \xi_3) d\eta + \frac{1}{\rho} \int_0^{\xi_3} f_3(\xi_1, \xi_2, \eta) d\eta \quad (3.41) \end{aligned}$$

The mixing term is clearly a piecewise continuous function of $\dot{\boldsymbol{\xi}}$ which may have jumps when $\dot{\xi}_i = 0$, $i = 1, 2$. Since the restrictions obtained by the second law were derived for smooth state variables, it is necessary to check (3.19) in the case when some $\dot{\xi}_i = 0$. Note that $\dot{\boldsymbol{\xi}}$ is the only variable, for which G is not smooth and observe that

$$\dot{G}^{\text{mix}} = \sum_{i=1}^3 \left(f_i \left(\boldsymbol{\xi}; \text{sgn} \left(\dot{\xi}_i \right) \right) \dot{\xi}_i(t) \right).$$

Further, the rate of the smooth part of G can be calculated again as in (3.24), and taking into account the assumed functional form (3.37) of G , the second law (3.19) becomes:

$$\left(\varepsilon + \rho \frac{\partial G}{\partial \boldsymbol{\sigma}} \right) : \dot{\boldsymbol{\sigma}} + \rho \left(s + \frac{\partial G}{\partial T} \right) \dot{T} + \rho \frac{\partial G}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} + \frac{\mathbf{q} \cdot \nabla T}{T} \leq 0$$

In view of the constraints (3.28)–(3.30), the last inequality holds and therefore the second law is shown to be satisfied for all possible values of the state variables.

The above assumption on the form of the Gibbs free energy mixing term and equation (3.34) with the help of (3.28) and (3.29) imply the following constitutive relations:

$$\boldsymbol{\sigma} = \boldsymbol{\mathcal{S}}(\boldsymbol{\xi})^{-1} : \left(\varepsilon - \boldsymbol{\alpha}(\boldsymbol{\xi})(T - T_0) - \varepsilon^{in} \right), \quad (3.42)$$

$$s = \frac{1}{\rho} \boldsymbol{\alpha}(\boldsymbol{\xi})T + c(\boldsymbol{\xi}) \ln(T/T_0) + s_0(\boldsymbol{\xi}), \quad (3.43)$$

hold. The thermodynamic forces, conjugate to ξ_i will be denoted by π_i , $i = 1, 2, 3$, and where appropriate, the vector notation $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3)^T$ will be used. They are given by:

$$\pi_1 = -\rho \frac{\partial G}{\partial \xi_1} = \tilde{\pi}(\boldsymbol{\sigma}, T) - f_1(\boldsymbol{\xi}), \quad \text{whenever } \dot{\xi}_1 \neq 0, \quad (3.44)$$

$$\pi_2 = -\rho \frac{\partial G}{\partial \xi_2} = \boldsymbol{\sigma} : \boldsymbol{\Lambda}^t + \tilde{\pi}(\boldsymbol{\sigma}, T) - f_2(\boldsymbol{\xi}), \quad \text{whenever } \dot{\xi}_2 \neq 0, \quad (3.45)$$

$$\pi_3 = -\rho \frac{\partial G}{\partial \xi_3} = \boldsymbol{\sigma} : \boldsymbol{\Lambda}^d - f_3(\boldsymbol{\xi}), \quad \text{whenever } \dot{\xi}_3 > 0. \quad (3.46)$$

where $\tilde{\pi}$ is:

$$\begin{aligned} \tilde{\pi}(\boldsymbol{\sigma}, T) = & \frac{1}{2} \boldsymbol{\sigma} : \Delta \boldsymbol{\mathcal{S}} : \boldsymbol{\sigma} + \Delta \boldsymbol{\alpha} : \boldsymbol{\sigma} (T - T_0) \\ & - \rho \Delta c \left[(T - T_0) - T \ln \left(\frac{T}{T_0} \right) \right] + \rho \Delta s_0 T - \rho \Delta u_0. \end{aligned} \quad (3.47)$$

In this work, f_i , $i = 1, 2, 3$ are assumed to be polynomial functions:

The hardening function f_1 for the $A \leftrightarrow M^t$ transformation is assumed to depend on c_1 , and may be different for the forward and reverse transformation:

$$f_1 = \begin{cases} \Delta_1^+ f_1^+(c_1) & \text{for } \dot{\xi}_1 > 0 \\ \Delta_1^- f_1^-(c_1) & \text{for } \dot{\xi}_1 < 0 \end{cases}. \quad (3.48)$$

Here $f_1^\pm(c_1)$ are two arbitrary monotonously increasing functions in the interval $[0, 1]$ for the forward and reverse transformations $A \rightarrow M^t$ and $M^t \rightarrow A$ respectively which can be determined from experimental measurements. The two material constants Δ_1^\pm serve as a scaling factors for $f_1^\pm(c_1)$ respectively, so that

$$f_1^\pm(0) = 0, \quad f_1^\pm(1) = 1. \quad (3.49)$$

The hardening function f_2^\pm , f_3 for the stress induced martensitic transformation $A \leftrightarrow M^d$ and the reorientation of twinned martensite $M^t \rightarrow M^d$ respectively are

assumed to depend on the volume fraction of twinned martensite c_2 :

$$f_2 = \begin{cases} \Delta_2^+ f_2^+(c_2) & \text{for } \dot{\xi}_2 > 0 \\ \Delta_2^- f_2^-(c_2) & \text{for } \dot{\xi}_2 < 0 \end{cases}, \quad f_3 = \Delta_3 f_3(c_2) \quad \text{for } \dot{\xi}_3 > 0. \quad (3.50)$$

Similarly to equation 3.48, the material constants Δ_2^\pm and Δ_3 are scaling factors for the monotonous functions f_2^\pm and f_3 , respectively, and:

$$f_2^\pm(0) = 0, \quad f_2^\pm(1) = 1, \quad (3.51)$$

$$f_3(0) = 0, \quad f_3(1) = 1. \quad (3.52)$$

Several things should be noted about this selection of hardening functions. The choice of c_2 as the independent variable for f_2^\pm and f_3 has generally been accepted in the literature. The choice of c_1 as the unknown variable for f_1^\pm , while often used in the literature (Brinson, 1993; Juhasz et al., 2002; Leclercq and LExcellent, 1996) is not the only possible option. The total amount of austenite c_3 may be an equally suitable choice for certain classes of SMA materials.

The specific form of the functions f_i (e.g. polynomials, trigonometric functions, exponents, etc.) is material dependent and should be treated as part of the material specifications. The experimental tests, required to curve-fit a specific hardening function will be presented in Section 4.

Finally, the hardening functions depend indirectly on ξ through the volume fractions c_i (equations (3.3)-(3.5)). The volume fractions c_i have fixed bounds (cf. equation (3.2)). Hence, a hardening function which depends explicitly on c_i will have the property that the transformation strips (see next section) will not change with cyclic thermomechanical loading. It should be kept in mind that the position of the transformation strips in the phase diagram do evolve with cyclic repetition of thermo-

mechanical loading paths, so if such effects are to be considered, it may be beneficial to specify an explicit dependency of f_i^\pm on ξ_1 , ξ_2 and ξ_3 . The evolution of SMA material response however was outside the scope of this work.

3.4. Transformation surfaces and flow rules

It is assumed that a closed elastic domain is associated with each possible transformation, bounded by a transformation surface. The five surfaces are:

$$\Phi_1^+(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) = 0, \quad \text{whenever the } A \rightarrow M^t \text{ transformation takes place,} \quad (3.53)$$

$$\Phi_1^-(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) = 0, \quad \text{whenever the } M^t \rightarrow A \text{ transformation takes place,} \quad (3.54)$$

$$\Phi_2^+(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) = 0, \quad \text{whenever the } A \rightarrow M^d \text{ transformation takes place,} \quad (3.55)$$

$$\Phi_2^-(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) = 0, \quad \text{whenever the } M^d \rightarrow A \text{ transformation takes place,} \quad (3.56)$$

$$\Phi_3(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) = 0, \quad \text{whenever the } M^t \rightarrow M^d \text{ deformation takes place,} \quad (3.57)$$

and the elastic domains in stress-temperature space, for given $\boldsymbol{\xi}$, with respect to ξ_i are defined implicitly by the inequalities:

$$\left\{ (\boldsymbol{\sigma}, T) \mid \Phi_i^+(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) \leq 0, \quad \dot{\xi}_i > 0 \right\}, \quad \text{for } i = 1, 2, \quad (3.58)$$

$$\left\{ (\boldsymbol{\sigma}, T) \mid \Phi_i^-(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) \leq 0, \quad \dot{\xi}_i < 0 \right\}, \quad \text{for } i = 1, 2, \quad (3.59)$$

$$\left\{ (\boldsymbol{\sigma}, T) \mid \Phi_3(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) \leq 0, \quad \dot{\xi}_3 > 0 \right\}. \quad (3.60)$$

The first two inequalities describe the elastic domains of the two *forward* transformations, the second two inequalities the elastic domains of the two *reverse* transformations. The last inequality describes the elastic domain for the $M^t \rightarrow M^d$ transformation. In contrast to conventional plasticity, the phase transformation terminates, whenever the constraints (3.2) are violated. Therefore, the elastic domain associated with given phase transformation is assumed to be the entire space, when the

transformation is complete or there is no more material to transform.

Following Lagoudas and Shu (1999); Qidwai and Lagoudas (2000b), the following form for of the transformation surfaces is suggested:

$$\Phi_1^+(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) = \pi_1 - Y_1^+, \quad (3.61)$$

$$\Phi_1^-(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) = -\pi_1 - Y_1^-, \quad (3.62)$$

$$\Phi_2^+(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) = \pi_2 - Y_2^+, \quad (3.63)$$

$$\Phi_2^-(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) = -\pi_2 - Y_2^-, \quad (3.64)$$

$$\Phi_3(\boldsymbol{\sigma}, T, \boldsymbol{\xi}) = \pi_3 - Y_3, \quad (3.65)$$

where and Y_1^\pm , Y_2^\pm , Y_3 are measures of internal dissipation of the respective transformations. In this work it is assumed that Y_i^\pm , $i = 1, 2, 3$ are constants, independent of $\boldsymbol{\sigma}$, T and $\boldsymbol{\xi}$. This, due to the inequalities (3.58)-(3.60), implies that the appropriate conjugate forces π_i remain constant during the transformation. It also implies that the entropy production due to a phase transformation is proportional to $\dot{\xi}_i$, with Y_i^\pm being the proportionality constant (cf. equation (3.31)). As will be seen later, Y_3 being constant implies that k^d is zero (cf. Figure 27), that is, the temperature dependence of the $M^t \rightarrow M^d$ surface will be ignored. This assumption was made for the sake of simplicity only and the appropriate modification to include the small temperature dependence of $M^t \rightarrow M^d$ are straightforward.

The functions f_i defined by (3.39) appear in the definition of the transformation function (3.61)-(3.65) through the constitutive relations (3.44)-(3.46). They are the only terms in the transformation functions dependent on the internal variables $\boldsymbol{\xi}$, hence they are responsible for the transformation hardening.

In order to complete the model, the transformation tensors in the flow rules (3.9)

and (3.10) should be specified. Let $\text{dev}(\boldsymbol{\sigma})$ be the deviatoric stress:

$$\text{dev}(\boldsymbol{\sigma}) = \boldsymbol{\sigma} - \frac{1}{3}\text{tr}(\boldsymbol{\sigma})\mathbf{I}$$

and $\|\cdot\|$ is the usual tensor norm:

$$\|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}},$$

The detwinning flow tensor is taken to of the form

$$\boldsymbol{\Lambda}^d = \sqrt{\frac{3}{2}}H \frac{\text{dev}(\boldsymbol{\sigma})}{\|\text{dev}(\boldsymbol{\sigma})\|}. \quad (3.66)$$

where H^d is the maximal uniaxial inelastic strain, assumed to be a material constant.

The flow rule used for the $A \leftrightarrow M^d$ transformation is, following Lagoudas et al. (1996); Qidwai and Lagoudas (2000a) taken to be

$$\boldsymbol{\Lambda}^t = \begin{cases} \sqrt{\frac{3}{2}}H^t \frac{\text{dev}(\boldsymbol{\sigma})}{\|\text{dev}(\boldsymbol{\sigma})\|} & \text{for } \dot{\xi}_2 > 0 \\ \sqrt{\frac{3}{2}}H^t \frac{\text{dev}(\boldsymbol{\varepsilon}^{in})}{\|\text{dev}(\boldsymbol{\varepsilon}^{in})\|} & \text{for } \dot{\xi}_2 < 0 \end{cases} \quad (3.67)$$

where H^t is a material constant having the meaning of maximal uniaxial transformation induced strain. The reverse transformation tensor of the last equation is discussed first.

The reason why two different transformation flow tensors are used for loading and unloading is the need to account for reorientation in multiaxial loading path. In general, if the direction of the stress state is changed, some martensitic variants will reorient in the new direction, thus changing the direction of the inelastic strain. A constitutive model with a single volume fraction for all detwinned variants of martensite cannot account for this process. If the same transformation tensor is used for forward and reverse transformations it may happen that residual inelastic strain is

present after unloading to austenite (e.g. $c_3 = 1$ and the stress becomes zero) from a non-proportional loading path. The unloading criterion used above ensures that when $c_3 = 1$, the inelastic strain becomes zero. It reduces to the same transformation tensor used by (Qidwai and Lagoudas, 2000a) when $\xi_3 = 0$.

The reader will recognize that the two flow rules, $\mathbf{\Lambda}^d$, and $\mathbf{\Lambda}^t$ when $\xi_2 > 0$ are normal to a J_2 based hyper-surface (which is not the actual transformation surface) from classical plasticity, that is,

$$J_2 = H^t \sqrt{\frac{3}{2}} \|\text{dev}(\boldsymbol{\sigma})\| \quad (3.68)$$

and

$$J_2 = H^d \sqrt{\frac{3}{2}} \|\text{dev}(\boldsymbol{\sigma})\|. \quad (3.69)$$

While the flow rules are not associative with respect to the actual surfaces (3.63) and (3.64), respectively they are associative in the space of generalized thermodynamic forces (Qidwai and Lagoudas, 2000b). In fact, the flow rules and transformation surfaces are closely connected. Consider for a moment the $A \rightarrow M^d$ transformation. A natural notion of maximal transformation dissipation can be introduced (Qidwai and Lagoudas, 2000b; Rajagopal and Srinivasa, 1998), that is, of all possible processes, the evolution of the transformation strain (3.9) is such that the work expended by the thermodynamic force π_2 attains a maximum. It can be shown that this notion of maximal transformation dissipation implies a flow rule normal to the transformation surface in the space of generalized thermodynamic forces. The reader is referred to Qidwai and Lagoudas (2000b) for further details.

The motivation for selecting different transformation surfaces on the other hand, is the observed tension-compression asymmetry of SMA materials as well as the development of a small volumetric strain during phase transformations (Lexcelent et al.,

2002; Qidwai and Lagoudas, 2000b). Together with the assumption of maximum transformation dissipation, the selection of a surface also implies a flow rule. Due to the large number of different SMA alloys the selection of an appropriate transformation surface can be a difficult task. Since the main goal of the current research is the formulation of a consistent model capable of accounting for phase transformation and detwinning, the specific choice of transformation functions was not addressed in detail.

As a last remark, suppose that transformation function and the flow rule are selected independently and the later is non-associative with respect to the selected surface. In that case, one has to check that the second law is satisfied, as one no longer has $\pi_i \dot{\xi}_i = Y_i \dot{\xi}_i > 0$ during the transformation. For example, if the same J_2 based surfaces is used for the pseudoelastic unloading, that is, Φ_2^- has the same form as Φ_2^+ , but the flow rule is retained, not only is it non-associative, but a loading path can be constructed which violates the second law of thermodynamics. Since this is against our religion, we cannot accept such a surface/flow rule.

4. Determination of material parameters

To summarize, the material parameters entering the model are:

$$\mathcal{S}^i, \boldsymbol{\alpha}^i, c^i, s_0^i, u_0^i, H^t, H^d, \quad (3.70)$$

$$f_1^\pm, f_2^\pm, f_3, Y_1^\pm, Y_2^\pm, Y_3, \Delta_1^\pm, \Delta_2^\pm, \Delta_3. \quad (3.71)$$

where the index i takes the values A, M for austenite and martensite, respectively. A polycrystalline SMA, unlike the single crystal SMAs, is an isotropic material. Therefore the compliances $\mathcal{S}^A, \mathcal{S}^M$ are determined if the Young's modulus E^A, E^M and Poisson's ratio ν^A, ν^M of the two phases are available. These can be determined from

standard uniaxial pseudoelastic test. The thermal expansion coefficient α^A , α^M for an isotropic material are scalars and are determined from an isobaric test and the specific heats c^A , c^M , the change in specific entropy $\rho\Delta s_0$ between the two phases and the change of specific internal energy Δu_0 can be determined from calorimetric measurements (Bo and Lagoudas, 1999a,b). The maximum uniaxial transformation strain H can be obtained from either an isothermal test or from an isobaric test (Bo and Lagoudas, 1999a).

The remaining parameters are related to the position of the transformation strips/surfaces in the uniaxial phase diagram in Figure 27. To this end, assume that the critical temperatures M_s , M_f , A_s^t , A_f^t , A_s^d , A_f^d as well as the critical stresses σ_s and σ_f are known. In the next section, a one-dimensional reduction of the model will be presented and the material parameters (3.71) will be expressed in terms of the critical transformation temperatures and stresses. Then, in Section 4.2 it will be shown that this one-dimensional reduction conforms with the phase diagram of Figure 27.

4.1. Uniaxial reduction of the model

Having determined the phase diagram, the remaining material parameters are selected so that the model conforms with the phase diagram. To do that it is necessary to write the transformation surfaces explicitly in the uniaxial stress case. That is, a stress state

$$\sigma_{11} = \sigma, \quad \sigma_{12} = \dots = \sigma_{33} = 0 \quad (3.72)$$

is assumed. Since uniaxial loading is always proportional, any combination of detwinning $M^t \rightarrow M^d$, forward $A \rightarrow M^d$ or reverse $M^d \rightarrow A$ by virtue of (3.66) and/or (3.67) will result in a transformation strain:

$$\varepsilon_{11}^{in} = H, \quad \varepsilon_{22}^{in} = \varepsilon_{33}^{in} = -\frac{1}{2}H, \quad \varepsilon_{12}^{in} = \varepsilon_{13}^{in} = \varepsilon_{23}^{in} = 0. \quad (3.73)$$

With this in mind, the inequalities (3.58)-(3.60) become:

$$\hat{\pi}(\sigma, T) - \Delta_1^+ f_1^+(c_1) \leq Y_1^+ \quad (3.74)$$

$$-\hat{\pi}(\sigma, T) + \Delta_1^- f_1^-(c_1) \leq Y_1^- \quad (3.75)$$

$$\sigma H + \hat{\pi}(\sigma, T) - \Delta_2^+ f_2^+(c_2) \leq Y_2^+ \quad (3.76)$$

$$-\sigma H - \hat{\pi}(\sigma, T) + \Delta_2^- f_2^-(c_2) \leq Y_2^- \quad (3.77)$$

$$\sigma H - \Delta_3 f_3(c_2) \leq Y_3 \quad (3.78)$$

where,

$$\begin{aligned} \hat{\pi}(\sigma, T) = & \Delta S \sigma^2 + \Delta \alpha \sigma (T - T_0) \\ & - \rho \Delta c \left[(T - T_0) - T \ln \left(\frac{T}{T_0} \right) \right] + \rho \Delta s_0 T - \rho \Delta u_0. \end{aligned} \quad (3.79)$$

Note that whenever one or more transformations are taking place (that is, $\dot{\xi}_i \neq 0$, $i = 1, 2, 3$) the respective inequalities (3.74)-(3.78) turn into equalities.

The unknown quantities in the above equations are Y_1^+ , Y_1^- , Y_2^+ , Y_2^- , Y_3 , Δ_1^+ , Δ_1^- , Δ_2^+ , Δ_2^- , Δ_3 as well as the functional form of f_i .

The phase diagram on the other hand is completely defined if the start and finish detwinning stresses σ_s and σ_f are known, the transformation temperatures M_s , M_f , A_s^t , A_f^t , A_s^d , A_f^d and the initial slope k of the $A \leftrightarrow M^d$ transformation surfaces. All these parameters can be determined after sufficient number of mechanical tests are performed.

The model parameters are then established as follows. Consider a loading path in which a purely twinned SMA ($c_1 = 1$, $c_2 = c_3 = 0$) is loaded at temperature below M_f . As the detwinning deformation progresses, $\dot{\xi}_3 > 0$, and the inequality (3.78) becomes an equality:

$$\sigma H - \Delta_3 f_3(c_2) = Y_3, \quad (3.80)$$

Therefore, at the beginning of the detwinning deformation one has $\sigma = \sigma_s$, $c_2 = 0$, and the last equation, together with (3.52) implies:

$$Y_3 = \sigma_s H.$$

Similarly, upon completion of the deformation, one has $\sigma = \sigma_f$, $f_3(1) = 1$ and $\Phi_3 = 0$, hence:

$$\Delta_3 = H(\sigma_f - \sigma_s).$$

The function f_3 itself is curve-fitted from a stress-strain relationship obtained in a standard isothermal loading test at some fixed temperature below M_f .

The material parameters Y_1^\pm , Δ_1^\pm for the $A \leftrightarrow M^t$ are determined with the help of a zero stress cooling/heating cycle. During cooling, the forward transformation surface (3.74) turns into equality:

$$\hat{\pi}(\sigma, T) - \Delta_1^+ f_1^+(c_1) = Y_1^+, \quad (3.81)$$

which, in conjunction with (3.49) yields:

$$Y_1^+ = \hat{\pi}(0, M_s) = -\rho \Delta u_0 + \rho \left(M_s \Delta s_0 - \Delta c \left[(M_s - T_0) - M_s \ln \left(\frac{M_s}{T_0} \right) \right] \right), \quad (3.82)$$

$$\begin{aligned} \Delta_1^+ &= \hat{\pi}(0, M_f) - Y_1^+ = \hat{\pi}(0, M_f) - \hat{\pi}(0, M_s) \\ &= \rho \left((M_f - M_s) \Delta s_0 - \Delta c \left[M_f - M_s + M_s \ln \left(\frac{M_s}{T_0} \right) - M_f \ln \left(\frac{M_f}{T_0} \right) \right] \right). \end{aligned} \quad (3.83)$$

Similarly, during the heating (3.75) becomes:

$$-\hat{\pi}(\sigma, T) + \Delta_1^- f_1^-(c_1) = Y_1^-, \quad (3.84)$$

hence Y_1^- and Δ_1^- can be determined:

$$\begin{aligned} Y_1^- &= -\hat{\pi}(0, A_f^t) \\ &= \rho\Delta u_0 - \rho \left(A_f^t \Delta s_0 - \Delta c \left[(A_f^t - T_0) - A_f^t \ln \left(\frac{A_f^t}{T_0} \right) \right] \right), \end{aligned} \quad (3.85)$$

$$\begin{aligned} \Delta_1^- &= \hat{\pi}(0, A_s^t) + Y_1^- = \hat{\pi}(0, A_s^t) - \hat{\pi}(0, A_f^t) \\ &= \rho \left((A_s^t - A_f^t) \Delta s_0 - \Delta c \left[A_s^t - A_f^t + A_f^t \ln \left(\frac{A_f^t}{T_0} \right) - A_s^t \ln \left(\frac{A_s^t}{T_0} \right) \right] \right). \end{aligned} \quad (3.86)$$

Determining the parameters for the stress-induced martensitic transformation is done by considering two loading paths. First, assume a fully detwinned state at some temperature below A_s^d and at zero stress (this can be obtained by loading isothermally at $T \leq M_f$ until all the material has detwinned and then unloading until zero stress is reached) and heat, while maintaining the material stress free. Then $\dot{\xi}_2 < 0$ and (3.77) becomes an equality:

$$-\sigma H - \hat{\pi}(\sigma, T) + \Delta_2^- f_2^-(c_2) = Y_2^-. \quad (3.87)$$

Noting that $\sigma = 0$ throughout the loading path, and with the help of (3.51), Y_2^- and Δ_2^- are found to be:

$$\begin{aligned} Y_2^- &= -\hat{\pi}(0, A_f^d) \\ &= \rho\Delta u_0 - \rho \left(A_f^d \Delta s_0 - \Delta c \left[(A_f^d - T_0) - A_f^d \ln \left(\frac{A_f^d}{T_0} \right) \right] \right), \end{aligned} \quad (3.88)$$

$$\begin{aligned} \Delta_2^- &= \hat{\pi}(0, A_s^d) + Y_2^- = \hat{\pi}(0, A_s^d) - \hat{\pi}(0, A_f^d) \\ &= \rho \left((A_s^d - A_f^d) \Delta s_0 - \Delta c \left[A_s^d - A_f^d + A_f^d \ln \left(\frac{A_f^d}{T_0} \right) - A_s^d \ln \left(\frac{A_s^d}{T_0} \right) \right] \right). \end{aligned} \quad (3.89)$$

Finally, in order to determine Y_2^+ and Δ_2^+ , load the material in austenite to some stress level, for example, σ_f and then cool the material. Let the critical temperatures for the $A \rightarrow M^d$ transformation at this stress level be $T_s(\sigma_f)$ for the start and $T_f(\sigma_f)$

for the finish. Then the constraint (3.76) becomes:

$$\sigma H + \hat{\pi}(\sigma, T) - \Delta_2^+ f_2^+(c_2) = Y_2^+ \quad (3.90)$$

which results in

$$Y_2^+ = \sigma_f H + \hat{\pi}(\sigma_f, T_s(\sigma_f)), \quad (3.91)$$

$$\Delta_2^+ = \sigma_f H + \hat{\pi}(\sigma_f, T_f(\sigma_f)) - Y_2^+. \quad (3.92)$$

Note that it is necessary to load to a stress equal or higher than σ_f , in order to avoid development of twinned martensite.

With this last equation, all the material parameters, except for the functional form of f_i^\pm are expressed from physically observable quantities. Through the rest of this work, it will be assumed that the hardening functions are linear:

$$f_1^\pm(c_1) = c_1, \quad f_2^\pm(c_2) = c_2, \quad f_3(c_2) = c_2.$$

In principle however, the model allows for arbitrary monotonous functions which can be curve-fitted from experiments. The curve-fit for f_2^\pm can be done from a single uniaxial loading (Lagoudas et al., 1996, cf., e.g.,). A curve-fit for $f_1^\pm(c_1)$ can be obtained by using a DSC measurement in conjunction with the balance of energy. These issues, however, will not be discussed further.

4.2. The uniaxial transformation strips and the phase diagram

The one dimensional reduction of the model resulted in the inequalities (3.74)-(3.78) for the elastic domains of the respective transformations. It is clear from equation (3.80) that the transformation strip in stress-temperature space for the $M^t \rightarrow M^d$

deformation is the horizontal strip

$$\sigma_s \leq \sigma \leq \sigma_f,$$

which is consistent with the assumptions of Section 2, see also the remarks after equation (3.65).

Next, assume for a moment that the elastic moduli of the two phases, the thermal moduli and the specific heats of the two phases are equal:

$$\mathcal{S}^A = \mathcal{S}^M, \quad \alpha^A = \alpha^M, \quad c^a = c^M.$$

In this case, equation (3.79) reduces to

$$\hat{\pi} = \rho \Delta s_0 T - \rho \Delta u_0.$$

Then, equation (3.81) implies that the transformation strip for the $A \rightarrow M^t$ is defined by

$$M_f \leq T \leq M_s$$

and from equation (3.84), the transformation strip for the $M^t \rightarrow A$ is the vertical region

$$A_s^t \leq T \leq A_f^t.$$

It can also be seen from equations (3.90) and (3.87) that, for any given c_2 , the transformation line for both $A \rightarrow M^d$ and $M^d \rightarrow A$ transformation is linear and has slope

$$k = -\frac{\rho \Delta s_0}{H}.$$

Therefore, the $A \rightarrow M^d$ and $M^d \rightarrow A$ strips have the shape shown in Figure 27, and the slope k is given by the above formula. This last formula is frequently used (for example, by Qidwai and Lagoudas, 2000a) to determine the difference in specific

entropies Δs_0 .

Now, when the moduli for the two phases are different, the transformation lines for the $A \leftrightarrow M^t$ and $A \leftrightarrow M^d$ depart from the above linear relationships. However, the terms $\Delta S\sigma^2$, $\Delta\alpha\sigma$ and $\rho\Delta c\left[(T - T_0) - T \ln\left(\frac{T}{T_0}\right)\right]$ which will now appear in (3.79) are all an order of magnitude smaller than the leading term $\rho\Delta s_0 T$. The departure from a linear shape is therefore visible for high stress (several hundred *MPa*) for $A \leftrightarrow M^t$ transformation and for both higher stresses and away from the equilibrium temperature T_0 for the $A \leftrightarrow M^d$ transformation. An example of the effect of different material moduli on the transformation surfaces is shown in Figure 29. Note that the $M^t \rightarrow M^d$ strip is not shown to improve the clarity of the figure. It is easy to show, that in the general case of different elastic and thermal moduli, the meaning of the slope k becomes now the tangent to the transformation surface at zero stress.

The derivation of the material parameters in the previous section assumed that the critical temperatures M_s , M_f , A_s^t , A_f^t , A_s^d , A_f^d , and the critical stresses σ_s , σ_f are known. We will conclude this section by discussing how these can be measured.

The transformation temperatures M_s , M_f , A_s^t , A_f^t can be easily determined from a DSC test such as the one shown in Figure 25. Knowing the critical temperatures at zero stress is sufficient to determine the $A \rightarrow M^t$ and $M^t \rightarrow A$ transformation strips. The A_s^d , A_f^d temperatures on the other hand can be found by first loading a specimen in detwinning conditions until the specimen has entirely detwinned. It is then mechanically unloaded in a way which preserves the material state and then a DSC test is performed as described in Section 1. To do the DSC test it is necessary to perform the mechanical loading and the subsequent preparation of a DSC sample from the loaded specimen at temperatures below A_s^d , which may not always be possible. A more direct approach relies on several isothermal test above A_f^t , which will allow to construct both the $A \rightarrow M^d$ and $M^d \rightarrow A$ strips and therefore, also the A_s^d , A_f^d

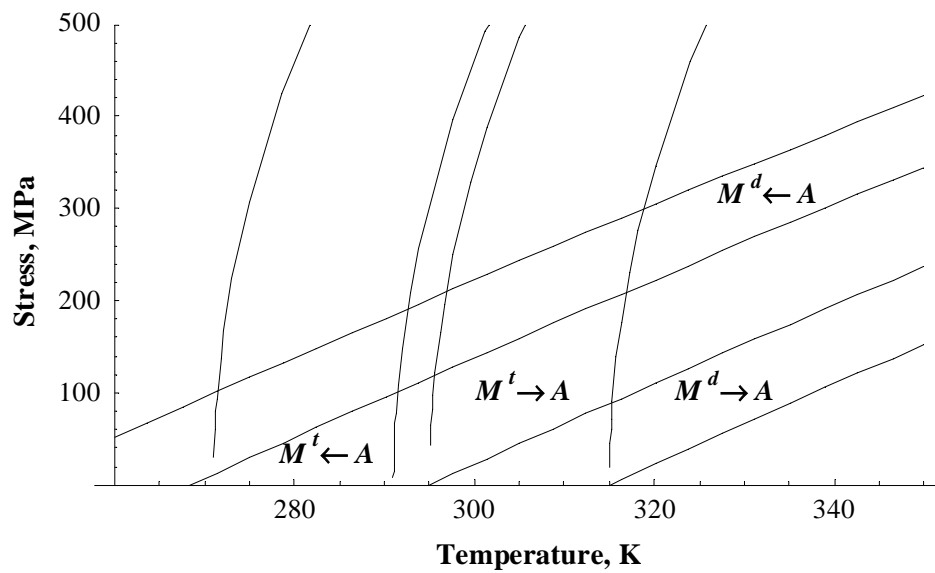


Fig. 29. Transformation strips for the $A \leftrightarrow M^t$ and $A \leftrightarrow M^d$ as predicted by the SMA model. Different material moduli are used (cf. Table IV). The deviation from straight line for the $A \leftrightarrow M^t$ transformation becomes significant for stresses above 200MPa . Considering that the σ_f rarely exceeds such values, the effects will be negligible. The deviation for the $A \leftrightarrow M^d$ strips is not visible until much higher temperatures and stresses, not shown on the figure.

temperatures. Isothermal tests at temperatures below M_f can be used to determine σ_s and σ_f and hence the $M^t \rightarrow M^d$ strip.

4.3. Limitations

It should also be noted that the current model does not assume any triple point as often done in the literature Brinson (1993). As a result, the $A \rightarrow M^d$ strip can be translated according to the experimental measurements. Certain restrictions, which result from the assumed functional dependence of f_1 , f_2 and f_3 however are still valid. To the best of the authors knowledge, two of these exist and will be mentioned briefly here. Both of them occur for certain specific material parameters and it may happen that such classes of SMA materials do not exist in practice.

First, the transformation strips $M^t \rightarrow A$, $M^d \rightarrow A$ and $M^t \rightarrow M^d$ must have a zero intersection. It is easy to show, that if they do, the three inequalities (3.75), (3.77) and (3.78) cannot be satisfied simultaneously. In other words, a simultaneous transformation $M^t \rightarrow A$, $M^d \rightarrow A$ and $M^t \rightarrow M^d$ is not possible. In light of the experimental results of Section 1, it seems unlikely that such a situation can occur. Also, it is physically difficult to explain why some twinned martensite will transform to austenite through an intermediate detwinned phase, while the rest of the twinned martensite will transform directly to austenite. This limitation can be removed by assuming a different functional dependence of f_1 , for example, on c_3 .

The second limitation of the theory is associated with a bifurcation in the stress for some material parameters. Suppose that the finish line for the $A \rightarrow M^d$ transformation passes above the point (M_s, σ_f) in stress temperature space. Then, one can find thermomechanical which starts in austenite, which, under specific cooling at some stress slightly higher than σ_f which will produce twinned martensite above the critical stress σ_f . As this happens, if the constraint () is to be satisfied, a finite

amount of detwinning must occur which will result in a discontinuous drop of the stress. Whether such materials, for which the finish $A \rightarrow M^d$ line passes above the point (M_s, σ_f) , exist is an open question. Again, for such class of materials a different functional dependence of f_1 may provide a solution to the problem.

5. Uniaxial examples

The numerical examples in this section were selected so that complicated loading path in stress-temperature phase space could be tested. Two uniaxial examples (Section 5.1) of a constrained SMA rod are considered. One-dimensional setting allows to carefully select thermomechanical loading path in the phase space (Figure 27) and the relevant equations can be solved with symbolic software.

In the first of these examples (Section 5.1.1), an SMA rod is cooled from the austenitic phase to low temperature while the strain is kept constrained. This problem allows to demonstrate the cut-off of the $A \rightarrow M^d$ transformation in untrained SMA materials and the predominant development of M^t from A at low stress levels. Secondly, an SMA rod is heated, again while constrained (Section 5.1.2). During the heating, simultaneous $M^t \rightarrow A$ and $M^t \rightarrow M^d$ transformations take place, thus offering a nontrivial test for the model. The basic material parameters used in the simulations are given in Table IV and represent a generic SMA properties Qidwai and Lagoudas (2000a). The only exception is the loading path of Section 5.1.2, where the critical stresses for detwinning are half the values given in the table. This selection is made in order to illustrate the class of SMA materials from which $M^t \rightarrow A$ and $M^t \rightarrow M^d$ transformations can happen simultaneously.

Table IV. Material parameters used in the three phase, 3-D SMA model.

Material constant	Value	Material constant	Value
E^A	$70 \times 10^9 \text{ Pa}$	M_f	$275 \text{ }^\circ K$
E^M	$30 \times 10^9 \text{ Pa}$	M_s	$291 \text{ }^\circ K$
α^A	$22 \times 10^{-6}/K$	A_s^t, A_s^d	$295 \text{ }^\circ K$
α^M	$10 \times 10^{-6}/K$	A_f^t, A_f^d	$315 \text{ }^\circ K$
H	0.05	σ_s	100 MPa
k	$4.5 \times 10^6 \text{ Pa}/(m^3 K)$	σ_f	200 MPa

5.1. Constrained SMA rod

Consider a rod in uniaxial stress state (3.72), (3.73), which is loaded isothermally and then its two ends are fixed. The stress in the rod is connected to the strain by

$$\sigma = E(c_1 + c_2) (\varepsilon - \alpha(c_1 + c_2)(T - T_0) - \varepsilon^{in}) \quad (3.93)$$

Assume that the maximal detwinning and transformation strains are the same, e.g. $H^t = H^d = H$. In the uniaxial case, the inelastic strain is proportional to the volume fraction of detwinned martensite c_3 :

$$\varepsilon^{in} = Hc_2. \quad (3.94)$$

The relevant transformation surfaces were solved using symbolic software (Mathematica).

5.1.1. Constrained cooling of a rod

Suppose that a rod is initially loaded in the austenitic phase up to a stress σ_0 , which is below the critical stress σ^{Ms} (cf. Figures 3 and 4), required to initiate the forward, $A \rightarrow M^d$ phase transformation. Without loss of generality, let this be a tensile stress.

Then the inelastic strain is identically zero:

$$\varepsilon^{in} = 0.$$

From equation (3.93), the rod has developed uniform elastic strain

$$\varepsilon_0 = \sigma_0/E^A.$$

At this point of the loading path, the strain is then fixed and the SMA is gradually cooled. The state of the material is plotted in stress-temperature space in Figure 30.

At first, a thermoelastic contraction of the rod increases slightly the stress. When the $A \rightarrow M^d$ transformation surface is reached, transformation strain begins to develop. Since, the maximal possible value of the transformation strain H is an order of magnitude larger than the elastic strain ε_0 , very little transformation is required to drastically reduce the stress. In this example the $A \rightarrow M^d$ surface terminates at some finite value of stress σ_s (which, as discussed before is material dependent). Slightly before this point the $A \rightarrow M^t$ transformation surface is also reached and the material undergoes combined transformation.

As the stress decreases below the critical stress σ_s , only the $A \rightarrow M^t$ transformation proceeds. In the process, no further transformation strain is produced, however the stiffness changes. The stiffness of the martensite E^M is less than the stiffness of austenite E^A , so the effective stiffness decreases (cf. equation (3.38)). On the other hand, the total strain is fixed. Therefore, neglecting the thermal strains, and noting that very small amount of $A \rightarrow M^d$ has occurred, from equation (3.93) it follows that the stress in the rod will decrease by a factor of E^A/E^M . This is clearly visible in Figure 30. Upon completion of the transformation, the a material again exhibits thermoelastic contraction, which causes small increase in stress.

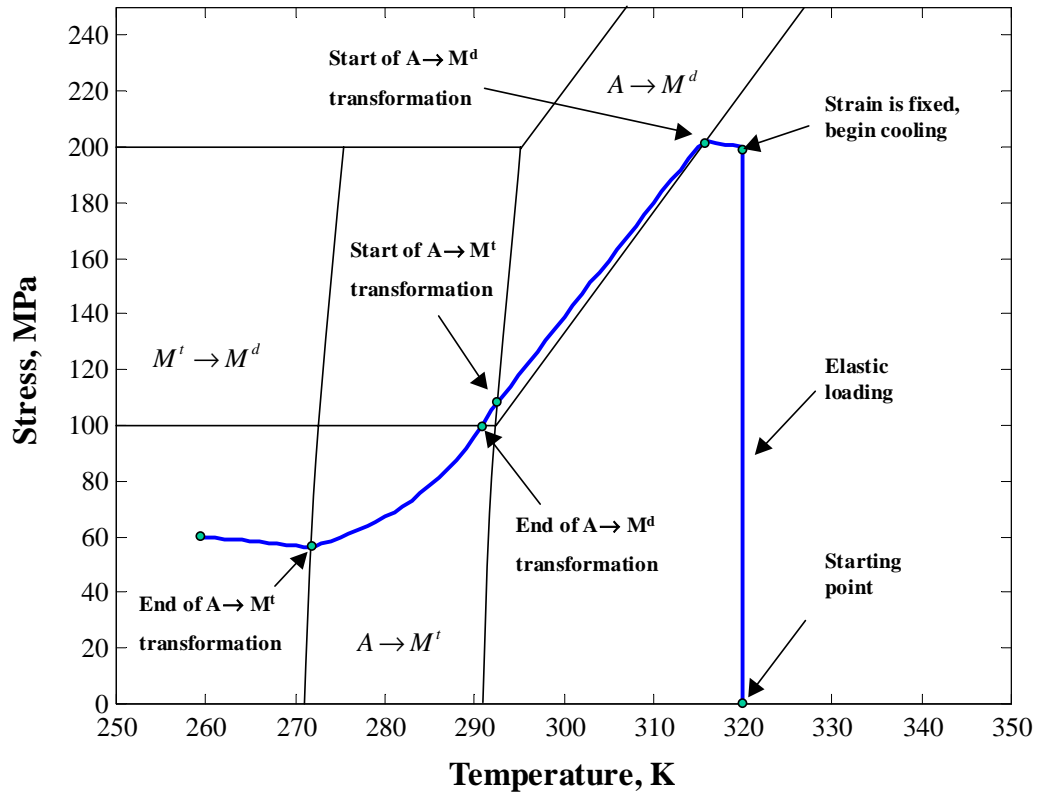


Fig. 30. A constrained cooling path in stress-temperature space. The rod is loaded in tension at the austenitic phase to a stress lower than required for phase transformation. The strain is then fixed and the rod is cooled. The rapid drop of the stress during the phase transformation is caused by the development of inelastic strains. Since the total achievable inelastic strain is an order of magnitude larger than the initial elastic strain, very little $A \rightarrow M^d$ transformation occurs. For clarity, only the $A \rightarrow M^d$, $A \rightarrow M^t$, $M^t \rightarrow M^d$ and transformation strips are shown.

5.1.2. Constrained cooling/heating of a rod

A more complicated example is given in Figure 31, where an SMA rod is constrained at zero stress while in austenite. It is then cooled, leading to initial development of thermal stresses, followed by the transformation of all austenite to twinned martensite. The specimen is then loaded in tension to a stress level σ within the $\sigma_s < \sigma < \sigma_f$ range, which causes part of the twinned martensite to detwin. Then the strain is again constrained and the specimen is then heated.

The interesting part of this loading path is the heating part. In the absence of any transformations the stress has a tendency to decrease, due to thermal expansion which relaxes the tensile stress. The $M^t \rightarrow A$ causes the effective stiffness of the material to increase. Similarly to the previous example, this in turn increases the stress. This can have two possible consequences. First, the raise in the stress may become sufficient to activate the $M^t \rightarrow M^d$ surface which will result in a simultaneous $M^t \rightarrow M^d, M^t \rightarrow A$ transformations. Secondly, as the temperature increases, the $M^d \rightarrow A$ transformation surface will also activate. This may happen simultaneously with the $M^t \rightarrow A$ or after the later is completed. As discussed in Section 4, the $M^t \rightarrow M^d$ and $M^d \rightarrow A$ strips are assumed not to intersect, so this combination is not possible. In the particular example shown on Figure 31, the $M^t \rightarrow A$ and $M^t \rightarrow M^d$ transitions happen sequentially over a short temperature range. Eventually, the M^t is depleted and only the $M^d \rightarrow A$ transformation proceeds, with the effect of producing large recovery stress. The $M^d \rightarrow A$ transformation reduces c_2 and therefore, the transformation strain (cf. equation (3.94)). Given that the total strain is fixed, the later results in increase of the elastic strain. Consequently, large recovery stress are observed (Figure 31).

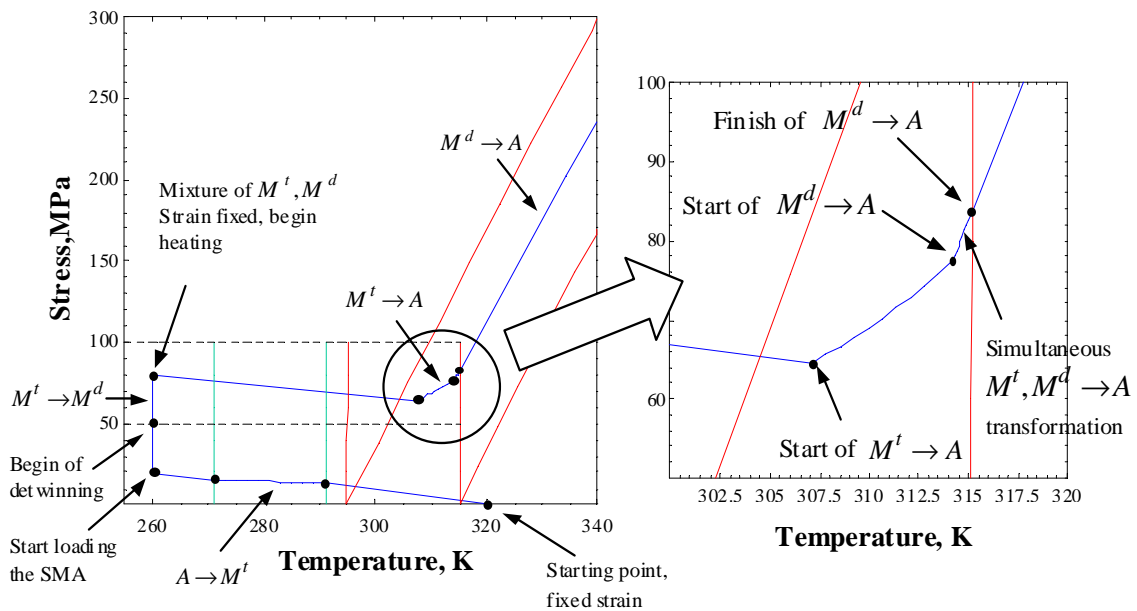


Fig. 31. Simulated loading path involving simultaneous reverse transformation. An SMA rod is cooled from austenite to twinned martensite, while the strain is fixed. It is then uniaxially loaded to obtain a mixture of twinned and detwinned martensite, the strain is fixed again and the rod is subsequently heated. Due to the mixture of the two phases, the transformation begins inside the transformation strips. For clarity, the $M^t \rightarrow M^d$ strip is omitted.

CHAPTER IV

NUMERICAL IMPLEMENTATION OF SMA CONSTITUTIVE MODEL

In this section the numerical implementation of thermomechanical constitutive equations is presented. The structure of the governing equations of an SMA constitutive model are very similar to the structure of the equations describing classical rate-independent plasticity. Therefore it is natural to utilize return-mapping algorithms developed for rate-independent plasticity (cf., e.g., Ortiz and Popov, 1985; Ortiz and Simo, 1986; Simo and Hughes, 1998) and apply them to SMAs.

Return mapping algorithms have been studied extensively over the years in the context of elasto-plasticity for the integration of constitutive relations (Simo and Hughes, 1998). They are also called elastic predictor-plastic corrector algorithms where a purely (thermo) elastic trial loading is followed by a plastic corrector phase (return mapping). The corrector part is only applied if the stress state after the predictor step violates the plastic yield condition (transformation surface for the case of SMAs). The corrector enforces continuity iteratively in a manner consistent with the prescribed flow rule. Return mapping algorithms may differ on the basis of the kind of discretization employed to numerically integrate the evolution differential equations for the flow rule and the iterative procedure adopted to solve the resultant set of non-linear algebraic equations in the corrector part. Some commonly used return mapping algorithms proposed in the literature are the radial return (backward Euler based) algorithm (Wilkins, 1964) and the mean normal (mid-point rule based) algorithm (Rice and Tracey, 1973). A detailed review of the historical developments of the return mapping algorithms can be found in (Simo and Hughes, 1998).

Ortiz and Popov (1985) have observed that most of the return mapping algorithms employ integration rules that are particular cases of the trapezoidal and mid-

point rules, suitably generalized to facilitate satisfaction of the plastic consistency condition. The stability of these integration algorithms for inelasticity is studied by Argyris et al. (1979); Simo and Govindjee (1991). The general conclusion is that backward integration strategies provide good stability of the incremental solution. Additional accuracy can be obtained by midpoint integration technique or by higher order approximation of the inelastic flow rule, for example by Runge-Kutta methods.

There are several differences between standard plastic materials and SMAs. SMAs have multiple transformation surfaces, e.g. forward and reverse stress-induced phase transformation as well as detwinning, if the model includes it. The surfaces are defined in both stress and temperature space. The reverse transformation surface is typically non-convex, which presents certain uniqueness issues discussed in Qidwai and Lagoudas (2000a). A typical flow rule such as the one for the forward transformation in equation (3.67) is not associative in stress space but becomes associative in the space of generalized thermodynamic forces. In addition, the material properties - compliance, thermal expansion coefficient, specific entropy and internal energy involved in the constitutive relationship change during the transformation (cf. equations (3.38)) which results in additional complications in the numerical implementation. Also, unlike plasticity, the phase transformation ends after which the loading proceeds elastically. This imposes an additional constraint (see inequalities (3.2)) which needs to be addressed as well.

Extensive research on return mapping algorithms for SMAs has only recently been performed by Qidwai and Lagoudas (2000a) who have implemented return-mapping algorithms for the family of SMAs models (Boyd and Lagoudas, 1994a, 1996a; Brinson, 1993; Liang and Rogers, 1990, 1992; Tanaka, 1986; Tanaka et al., 1986, 1995). Two algorithms are developed - the closest point projection algorithm Ortiz and Pinsky (1981) and convex cutting plane algorithm (Ortiz and Simo, 1986;

Simo and Ortiz, 1985). The difference between the two algorithms is in the corrector part. The application of the closest point projection algorithm results in a set of non-linear algebraic equations solved using Newtons iteration method. The closest point algorithm is unconditionally stable provided the yield surface is convex, and it is first-order accurate (Ortiz and Pinsky, 1981). On the other hand, the convex cutting plane method uses a Newton method only for the transformation surface and is based an explicit integration of the transformation flow rule. As a result it is computationally less demanding, however it is not unconditionally stable. While the work of Qidwai and Lagoudas (2000a) is focused on a single volume fraction of martensite, Govindjee and Miehe (2001) have focused on the numerical implementation by return mapping algorithms of models with multiple martensitic variants and, correspondingly, internal variables. In order to satisfy a polytope constraint an active set search strategy is proposed which determines the active variants and the corresponding thermoelastic prediction. The transformation correction is performed by the backward Euler method. The works considered above are all set up within the framework of infinitesimal strains. Auricchio (2001); Auricchio et al. (1997) have focused on adapting SMA constitutive models to finite strains and the have proposed a series of return-mapping algorithms based on the backward Euler integration scheme for the transformation correction.

1. Closest point projection algorithm for multiple internal variables

The major steps of the numerical implementation of the SMA constitutive model are presented next. The implementation can be viewed as an extension to the implementation of the earlier class of models described in Boyd and Lagoudas (1996a) for which the reader is referred to Qidwai and Lagoudas (2000a).

Equation (3.28) together with the definition of the free energy (3.34) imply that the total strain $\boldsymbol{\varepsilon}$ is given by

$$\boldsymbol{\varepsilon} = \boldsymbol{\mathcal{S}}(\boldsymbol{\xi}) : \boldsymbol{\sigma} + \boldsymbol{\alpha}(\boldsymbol{\xi})(T - T_0) + \boldsymbol{\varepsilon}^{in}. \quad (4.1)$$

The evolution equations (3.9) and (3.10) and the decomposition (3.8) imply that the total inelastic strain $\boldsymbol{\varepsilon}^{in}$ can be written as

$$\dot{\boldsymbol{\varepsilon}}^{in} = \boldsymbol{\Lambda}^t \dot{\boldsymbol{\xi}}_2 + \boldsymbol{\Lambda}^d \dot{\boldsymbol{\xi}}_3, \quad (4.2)$$

where $\boldsymbol{\Lambda}^t$ and $\boldsymbol{\Lambda}^d$ are defined by equations (3.67) and (3.66) respectively. It is also convenient to write the consistency conditions imposed by the transformation surfaces (3.58)–(3.65) for the evolution of the internal variables $\boldsymbol{\xi}$ in the following compact form:

$$\dot{\xi}_1 \geq 0, \quad \Phi_1^+ \leq 0, \quad \Phi_1^+ \dot{\xi}_1 = 0, \quad (4.3a)$$

$$\dot{\xi}_1 \leq 0, \quad \Phi_1^- \leq 0, \quad \Phi_1^- \dot{\xi}_1 = 0, \quad (4.3b)$$

$$\dot{\xi}_2 \geq 0, \quad \Phi_2^+ \leq 0, \quad \Phi_2^+ \dot{\xi}_2 = 0, \quad (4.3c)$$

$$\dot{\xi}_2 \leq 0, \quad \Phi_2^- \leq 0, \quad \Phi_2^- \dot{\xi}_2 = 0, \quad (4.3d)$$

$$\dot{\xi}_3 \geq 0, \quad \Phi_3 \leq 0, \quad \Phi_3 \dot{\xi}_3 = 0. \quad (4.3e)$$

Thus the final system of nonlinear differential-algebraic equations consists of equations (4.1)–(4.2) along with the constraints (4.3).

1.1. The loading step

As mentioned above, the implementation of the SMA model is focused on a single material point. Further, the history of all field and internal variables at this material point is known. In particular, the values of $\boldsymbol{\varepsilon}_n$, T_n , $\boldsymbol{\sigma}_n$, $\boldsymbol{\varepsilon}_n^{in}$, and $\boldsymbol{\xi}_n$ are known. The

subscript n is used to denote a history/time parameter¹. The new values of $\boldsymbol{\varepsilon}_{n+1}$ and T_{n+1} for the strain and temperature respectively are also given². Since the steps are discontinuous events, it is assumed that the continuous loading path which the material follows between step n and $n + 1$ is characterized by

$$\boldsymbol{\varepsilon}_{n+\alpha} = (1 - \alpha)\boldsymbol{\varepsilon}_n + \alpha\boldsymbol{\varepsilon}_{n+1}, \quad \text{for } \forall \alpha \in [0, 1], \quad (4.4)$$

and

$$T_{n+\alpha} = (1 - \alpha)T_n + \alpha T_{n+1}, \quad \text{for } \forall \alpha \in [0, 1]. \quad (4.5)$$

The Closest Point Projection Return Mapping Algorithm is a numerical method which computes the values for $\boldsymbol{\sigma}_{n+1}$, $\boldsymbol{\varepsilon}_{n+1}^{in}$ and $\boldsymbol{\xi}_{n+1}$ by solving equations (4.1)–(4.2) along with the constraints (4.3).

1.2. Closest point projection return mapping algorithm for SMA constitutive model

To solve this system, the evolution equation (4.2) for the transformation and detwinning strains are discretized as follows:

$$\boldsymbol{\varepsilon}_{n+1}^{in} = \boldsymbol{\varepsilon}_n^{in} + (\xi_{2n+1} - \xi_{2n})\boldsymbol{\Lambda}^t(\boldsymbol{\sigma}_{n+1}) + (\xi_{3n+1} - \xi_{3n})\boldsymbol{\Lambda}^d(\boldsymbol{\sigma}_{n+1}), \quad (4.6)$$

This type of backward Euler discretization is referred to as *the closest point projection* return mapping algorithm (Qidwai and Lagoudas, 2000a). The stress-strain

¹For a quasi-static problem, this would be the values at the n -th loading step, while in a dynamic problem this would be the values of the field and internal variables at some discrete instance of time t_n .

²Alternatively, the increments $\Delta\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n$ and $\Delta T_{n+1} = T_{n+1} - T_n$ may be given, which, of course, is equivalent to knowing $\boldsymbol{\varepsilon}_{n+1}$ and T_{n+1} .

relation (4.1) is equivalent to:

$$\boldsymbol{\sigma}_{n+1} = \mathcal{S}(\boldsymbol{\xi}_{n+1})^{-1} : (\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_{n+1}^{in} - \boldsymbol{\alpha}(\boldsymbol{\xi}_{n+1})(T_{n+1} - T_0)), \quad (4.7)$$

which is the discrete version of the generalized Hooke's law (3.42). In order to solve the discrete system (4.6), (4.7) subject to the constraints (4.3), first substitute $\boldsymbol{\varepsilon}_{n+1}^{in}$ from equation (4.6) into (4.7), multiply both sides by $\mathcal{S}(\boldsymbol{\xi}_{n+1})$ and then rearrange the terms to obtain:

$$\begin{aligned} & \mathcal{S}(\boldsymbol{\xi}_{n+1})\boldsymbol{\sigma}_{n+1} - \boldsymbol{\varepsilon}_{n+1} + \boldsymbol{\alpha}(\boldsymbol{\xi}_{n+1})(T - T_0) \\ & + \boldsymbol{\varepsilon}_n^{in} + (\xi_{2n+1} - \xi_{2n})\boldsymbol{\Lambda}^t(\boldsymbol{\sigma}_{n+1}) + (\xi_{3n+1} - \xi_{3n})\boldsymbol{\Lambda}^d(\boldsymbol{\sigma}_{n+1}) = \mathbf{0}. \end{aligned} \quad (4.8)$$

Note that in the above equation, all members with subscript n as well as $\boldsymbol{\varepsilon}_{n+1}$ and T_{n+1} have known values. It is convenient to introduce the residual \mathbf{F} :

$$\begin{aligned} \mathbf{F}(\boldsymbol{\sigma}, \boldsymbol{\xi}) = & \mathcal{S}(\boldsymbol{\xi})\boldsymbol{\sigma} - \boldsymbol{\varepsilon}_{n+1} + \boldsymbol{\alpha}(\boldsymbol{\xi})(T_{n+1} - T_0) \\ & + \boldsymbol{\varepsilon}_n^{in} + (\xi_2 - \xi_{2n})\boldsymbol{\Lambda}^t(\boldsymbol{\sigma}) + (\xi_3 - \xi_{3n})\boldsymbol{\Lambda}^d(\boldsymbol{\sigma}). \end{aligned} \quad (4.9)$$

Observe, that the system (4.6), (4.7) is now equivalent to

$$\mathbf{F}(\boldsymbol{\sigma}_{n+1}, \boldsymbol{\xi}_{n+1}) = \mathbf{0}. \quad (4.10)$$

The Closest Point Projection method, like most return mapping algorithms, first performs a linear thermoelastic loading using equation (4.7), called **thermoelastic prediction**. It then determines, using (4.3), if phase transformation occurs or not. If it does not, then the solution is accepted. If it does, it determines which one and performs a **transformation correction**. Without loss of generality, suppose that during the loading step the forward stress-induced phase transformation occurs and the rest of the phase transitions don't. This implies $\xi_{2n+1} - \xi_{2n} > 0$ and (4.3) reduces

to

$$\Phi(\boldsymbol{\sigma}_{n+1}, T_{n+1}, \boldsymbol{\xi}_{n+1}) = 0. \quad (4.11)$$

The Closest Point Projection method then does nothing else but to solve (4.10) and (4.11) by Newton's method in order to obtain a consistent material state. These two steps are explained in details below.

Both the predictor and corrector steps can be viewed as part of an iterative process which solves the nonlinear algebraic system of equations (4.10), subject to the constraints (4.3), by constructing a converging sequence

$$\boldsymbol{\sigma}_{n+1}^{(k)} \xrightarrow[k \rightarrow \infty]{} \boldsymbol{\sigma}_{n+1}, \quad \boldsymbol{\varepsilon}_{n+1}^{in(k)} \xrightarrow[k \rightarrow \infty]{} \boldsymbol{\varepsilon}_{n+1}^{in}, \quad \boldsymbol{\xi}_{n+1}^{(k)} \xrightarrow[k \rightarrow \infty]{} \boldsymbol{\xi}_{n+1}. \quad (4.12)$$

How this converging sequence is constructed is revealed next.

1.2.1. Thermoelastic prediction

As the first step, a **thermoelastic prediction** is performed during which, the internal variables do not change:

$$\boldsymbol{\varepsilon}_{n+1}^{in(0)} = \boldsymbol{\varepsilon}_n^{in}, \quad (4.13)$$

$$\boldsymbol{\xi}_{n+1}^{(0)} = \boldsymbol{\xi}_n, \quad (4.14)$$

$$\boldsymbol{\sigma}_{n+1}^{(0)} = \mathcal{S}(\boldsymbol{\xi}_n)^{-1} : [\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n^{in} - \boldsymbol{\alpha}(\boldsymbol{\xi}_n)(T_{n+1} - T_0)]. \quad (4.15)$$

It should be noted, that this first step corresponds to purely thermoelastic loading without any transformation ($\dot{\boldsymbol{\xi}} = 0$), hence its name thermoelastic prediction. The corresponding values of the five transformation functions are then evaluated:

$$\Phi_\alpha^{(0)} = \Phi_\alpha(\boldsymbol{\sigma}_{n+1}^{(0)}, T_{n+1}, \boldsymbol{\xi}_{n+1}^{(0)}). \quad (4.16)$$

The subscript α is understood in the sense $\Phi_\alpha \in \{\Phi_1^+, \Phi_1^-, \Phi_2^+, \Phi_2^-, \Phi_3\}$. If the value of all transformation functions satisfy $\Phi_\alpha^{(0)} \leq 0$ then all equations and constraints are satisfied and the iteration is terminated for $k = 0$.

1.2.2. Transformation correction

The predictor step assumed that $\xi_n = \xi_{n+1}$, hence (4.3) are satisfied iff all $\Phi_\alpha \leq 0$. Therefore, if at least one of the transformation function $\Phi_\alpha > 0$ then the corresponding consistency condition is violated. Such surfaces will be referred to as *inconsistent*. The existence of inconsistent surfaces implies that during the loading step, phase transformation takes place and a **transformation correction** is needed. During this step, the stress and the internal variables are modified in accordance with the transformation flow rules so that the consistency conditions are satisfied.

The consistency condition(s) which correspond to the phase transformation(s) taking place will be called *active*. The same term will be used for the respective transformation surfaces. Following the assumption of the previous chapter, no more than two surfaces can be active. The consistency conditions that the elastic predictor violates are not necessarily the active ones, nor are they necessarily the only ones active. An example can be found in the loading path of Figure 31 during heating when both $M^t \rightarrow A$ and $M^d \rightarrow A$ occur. In this temperature region, the last state n is such that $\Phi_3(\sigma_n, T_n, \xi_n) = 0$. Remember that in this particular loading path the strain is fixed and only the temperature is increased. It is easy to check (see (3.65) and (3.78)) that $\Phi_3(\sigma_n^{(0)}, T_{n+1}, \xi_{n+1}^{(0)}) = 0$ that is the thermoelastic predictor will not violate (4.3e), however, the final state does, because the stress increases due to the increase in stiffness caused by the $M^t \rightarrow A$ transformation. Therefore the inconsistent and active surfaces are not necessarily the same.

Assume, for a moment, that it is known which transformation(s) are active during

the load step. Suppose first, that only one transformation is active, say Φ_α . This implies that the corresponding volume fraction, denoted also by ξ_α , has nonzero rate³. That is, $\dot{\xi}_\alpha \neq 0$, and the corresponding consistency conditions (4.3) is satisfied, iff,

$$\Phi_\alpha(\boldsymbol{\sigma}_{n+1}, T_{n+1}, \boldsymbol{\xi}_{n+1}) = 0. \quad (4.17)$$

Therefore, during the transformation correction, one has to solve (4.10) along with the last equation. This is done by Newton's method: *For the given k -th iterate of $\boldsymbol{\sigma}_{n+1}^{(k)}$, $\boldsymbol{\varepsilon}_{n+1}^{in(k)}$ and $\boldsymbol{\xi}_{n+1}^{(k)}$, find the $k+1$ iterates by linearizing \mathbf{F} and Φ_α around $(\boldsymbol{\sigma}_{n+1}^{(k)}, \boldsymbol{\xi}_{n+1}^{(k)})$ and requiring that:*

$$\mathbf{F}^{(k)} + \frac{\partial \mathbf{F}^{(k)}}{\partial \boldsymbol{\sigma}} : \Delta \boldsymbol{\sigma}^{(k)} + \frac{\partial \mathbf{F}^{(k)}}{\partial \xi_\alpha} \Delta \xi_\alpha^{(k)} = 0, \quad (4.18)$$

$$\Phi_\alpha^{(k)} + \frac{\partial \Phi_\alpha^{(k)}}{\partial \boldsymbol{\sigma}} : \Delta \boldsymbol{\sigma}^{(k)} + \frac{\partial \Phi_\alpha^{(k)}}{\partial \xi_\alpha} \cdot \Delta \xi_\alpha^{(k)} = 0. \quad (4.19)$$

The shortcut notation for $\mathbf{F}^{(k)} = \mathbf{F}(\boldsymbol{\sigma}_{n+1}^{(k)}, \boldsymbol{\xi}_{n+1}^{(k)})$, $\Phi_\alpha^{(k)} = \Phi_\alpha(\boldsymbol{\sigma}_{n+1}^{(k)}, T_{n+1}, \xi_{n+1}^{(k)})$ and all their derivatives is used. When the increments $\Delta \boldsymbol{\sigma}^{(k)}$ and $\Delta \xi_\alpha^{(k)}$ are determined from the above system of linear equations, the stress and the internal variable are updated according to

$$\boldsymbol{\sigma}_{n+1}^{(k+1)} = \boldsymbol{\sigma}_{n+1}^{(k)} + \Delta \boldsymbol{\sigma}^{(k)}, \quad \xi_{\alpha n+1}^{(k+1)} = \xi_{\alpha n+1}^{(k)} + \Delta \xi_\alpha^{(k)},$$

and $\boldsymbol{\varepsilon}_{n+1}^{in(k)}$ is updated according to equation (4.6).

If two of the transformations are active, say Φ_α and Φ_β , then during the correction, equation (4.10) along with

$$\Phi_\alpha(\boldsymbol{\sigma}_{n+1}, T_{n+1}, \boldsymbol{\xi}_{n+1}) = 0, \quad (4.20)$$

³If $\Phi_\alpha \in \{\Phi_1^+, \Phi_1^-\}$ then the internal variable responsible is $\xi_\alpha = \xi_1$, if $\Phi_\alpha \in \{\Phi_2^+, \Phi_2^-\}$ then $\xi_\alpha = \xi_2$ and if $\Phi_\alpha = \Phi_3$ then $\xi_\alpha = \xi_3$

$$\Phi_\beta(\boldsymbol{\sigma}_{n+1}, T_{n+1}, \boldsymbol{\xi}_{n+1}) = 0, \quad (4.21)$$

is being solved, again by a Newton's method: *For the given k -th iterate of $\boldsymbol{\sigma}_{n+1}^{(k)}$, $\boldsymbol{\varepsilon}_{n+1}^{in(k)}$ and $\boldsymbol{\xi}_{n+1}^{(k)}$ find the $k+1$ iterates by linearizing \mathbf{F} , Φ_α and Φ_β around $(\boldsymbol{\sigma}_{n+1}^{(k)}, \boldsymbol{\xi}_{n+1}^{(k)})$ and requiring that:*

$$\mathbf{F}^{(k)} + \frac{\partial \mathbf{F}^{(k)}}{\partial \boldsymbol{\sigma}} : \Delta \boldsymbol{\sigma}^{(k)} + \frac{\partial \mathbf{F}^{(k)}}{\partial \xi_\alpha} \Delta \xi_\alpha^{(k)} + \frac{\partial \mathbf{F}^{(k)}}{\partial \xi_\beta} \Delta \xi_\beta^{(k)} = 0, \quad (4.22)$$

$$\Phi_\alpha^{(k)} + \frac{\partial \Phi_\alpha^{(k)}}{\partial \boldsymbol{\sigma}} : \Delta \boldsymbol{\sigma}^{(k)} + \frac{\partial \Phi_\alpha^{(k)}}{\partial \xi_\alpha} \cdot \Delta \xi_\alpha^{(k)} + \frac{\partial \Phi_\alpha^{(k)}}{\partial \xi_\beta} \cdot \Delta \xi_\beta^{(k)} = 0, \quad (4.23)$$

$$\Phi_\beta^{(k)} + \frac{\partial \Phi_\beta^{(k)}}{\partial \boldsymbol{\sigma}} : \Delta \boldsymbol{\sigma}^{(k)} + \frac{\partial \Phi_\beta^{(k)}}{\partial \xi_\alpha} \cdot \Delta \xi_\alpha^{(k)} + \frac{\partial \Phi_\beta^{(k)}}{\partial \xi_\beta} \cdot \Delta \xi_\beta^{(k)} = 0. \quad (4.24)$$

When the increments $\Delta \boldsymbol{\sigma}^{(k)}$, $\Delta \xi_\alpha^{(k)}$ and $\Delta \xi_\beta^{(k)}$ are determined from the above system of linear equations, the stress and the internal variable are updated according to

$$\boldsymbol{\sigma}_{n+1}^{(k+1)} = \boldsymbol{\sigma}_{n+1}^{(k)} + \Delta \boldsymbol{\sigma}^{(k)}, \quad \xi_{\alpha n+1}^{(k+1)} = \xi_{\alpha n+1}^{(k)} + \Delta \xi_\alpha^{(k)}, \quad \xi_{\beta n+1}^{(k+1)} = \xi_{\beta n+1}^{(k)} + \Delta \xi_\beta^{(k)},$$

and $\boldsymbol{\varepsilon}_{n+1}^{in(k)}$ is updated according to equation (4.6).

This completes the outline the return mapping algorithm. The details of solving the linear system (4.18)-(4.19) or (4.22)-(4.24), including the functional form of the derivatives involved will not be discussed.

It is important to note that when Φ_2^\pm is the only active surface, the iteration (4.18),(4.19) reduces to the Closest Point Projection method of Qidwai and Lagoudas (2000a). In the later work, the algorithm is formulated by defining a residual for the flow rule (4.6), instead of (4.8). It is easy to show that the two lead to the same algorithm. The current approach has the advantage that by taking the residual of Hooke's law the algorithm generalizes for the twinning transformation $A \leftrightarrow M^t$ in which no transformation strain is generated.

1.2.3. Active surfaces and other implementation details

Returning to the question, which transformation surfaces are active during the correction, observe, that at any time during the loading step, they can be separated into two groups, $\mathcal{F} = \{\Phi_1^+, \Phi_2^+, \Phi_3\}$ and $\mathcal{B} = \{\Phi_1^-, \Phi_2^-, \Phi_3\}$. If two phase transformation are active, say Φ_α and Φ_β , then either $\Phi_\alpha, \Phi_\beta \in \mathcal{F}$ or $\Phi_\alpha, \Phi_\beta \in \mathcal{B}$. The inconsistent surfaces from the predictor state need not follow the same rule. However, observe that the elastic prediction is a continuous mapping of $\boldsymbol{\sigma}_{n+1}$ with respect to T_{n+1} and $\boldsymbol{\varepsilon}_{n+1}$ and the loading step $n + 1$ is part of a continuous loading path (4.4)-(4.5). Then, for some sufficiently small loading parameter λ , the thermoelastic predictor for the state $n + \lambda$ will satisfy the above criterion. Since the thermoelastic prediction is a computationally inexpensive process, if the current prediction cannot determine whether the loading step belongs to \mathcal{F} or \mathcal{B} , the simplest practical approach is to take $\lambda = \frac{1}{2}, \frac{1}{4}, \dots$ until this can be determined.

Once this is done, in order to find which transformation(s) are active, the natural way is to attempt a correction of the ones which are violated first. It may happen that after the correction, some other transformation surface becomes inconsistent, or the increment of the corresponding volume fraction is inconsistent (the consistency conditions also specify the sign of $\dot{\boldsymbol{\xi}}$, that is, if active surface is Φ_1^- , then $\xi_{1n} > \xi_{1n+1}$, etc.). In such case the brute force approach of attempting all possible single and double transformations from the active set is used.

The last important detail is how to terminate the transformations, that is how to impose the constraints (3.2). Again, without loss of generality, suppose the correction step was restoring consistency of Φ_1^- , and either

$$c_{3n+1} < 0,$$

or

$$c_{1n+1} > 1.$$

Suppose $c_{1n+1} > 1$. The volume fraction can be treated as a continuous, monotonous function of the loading parameter λ and, which is more, $c_{1n} < 1$, therefore the equation

$$c_{1n+\lambda} = 1$$

has a root λ in the interval $\lambda \in [0, 1]$. Given such precise information about the location of the root and that the explicit form of $c_{1n+\lambda}$ as a function of λ does not seem easy to determine, a modified secant's method is used to determine a the root of the above equation. Each evaluation of $c_{1n+\lambda}$ consists of performing the transformation correction method.

1.3. Algorithmic tangent stiffness (Jacobian)

So far the return-mapping algorithm described in this section calculates the state variables at the current time step $n + 1$ at a material point. Depending on the numerical method used to solve a boundary value problem, the calculation of the state variables at material points may not be sufficient. By far the most popular numerical method for solving boundary value problems in quasi-static elasticity problems is the Finite Element Method. When the material is nonlinear, as is the SMA, the discretization generated by the FEM results in a system of nonlinear algebraic equations. If the FEM method is *displacement* based and this system is solved via a gradient type of method (such as Newton's method), it becomes necessary to compute not only the state variables, but also the derivatives of stress with respect to strain at a material point. These derivatives are usually referred to as the *algorithmic tangent stiffness* (cf., e.g., Simo and Hughes, 1998) and will be denoted by \mathcal{L} . The reader is referred to

Appendix B for short summary of a displacement based FEM, the resulting system of algebraic equations and how this jacobian \mathcal{L} appears when Newton's method is used to solve this nonlinear algebraic system.

The algorithmic tangent stiffness and thermal moduli will now be defined. Recall (Section 1.1), that the loading step is defined by specifying the strain $\boldsymbol{\varepsilon}_{n+1}$ and temperature T_{n+1} . All the remaining state variables $\boldsymbol{\sigma}_{n+1}$, $\boldsymbol{\xi}_{n+1}$, $\dot{\boldsymbol{\xi}}_{n+1}$ and $\boldsymbol{\varepsilon}_{n+1}^{in}$ are determined using the system of equations and constraints (4.6), (4.7) and (4.3). Thus, they can be treated as implicit functions of $\boldsymbol{\varepsilon}_{n+1}$ and T_{n+1} , and in particular:

$$\boldsymbol{\sigma}_{n+1} = \tilde{\boldsymbol{\sigma}}_{n+1}(\boldsymbol{\varepsilon}_{n+1}, T_{n+1}). \quad (4.25)$$

The tangent stiffness and thermal tangent moduli are defined by:

$$\mathcal{L} := \frac{\partial \tilde{\boldsymbol{\sigma}}_{n+1}(\boldsymbol{\varepsilon}_{n+1}, T_{n+1})}{\partial \boldsymbol{\varepsilon}_{n+1}} \quad (4.26)$$

and

$$\mathcal{M} := \frac{\partial \tilde{\boldsymbol{\sigma}}_{n+1}(\boldsymbol{\varepsilon}_{n+1}, T_{n+1})}{\partial T_{n+1}}, \quad (4.27)$$

respectively. The tangent stiffness is a fourth-order tensor while the thermal tangent is a second-order tensor.

Our goal is to arrive at an analytical expression for \mathcal{L} , given our inelastic SMA constitutive theory and the selected return mapping algorithm. In general it depends on which transformation surfaces are active. The subscript $n + 1$ will be omitted for the rest of the derivation, and all state variable without subscript will, by default, be considered at time step $n + 1$. The thermal tangent moduli are listed for completeness only, since they are required when a coupled, thermo-mechanical problem is solved and such problems are not considered in this work. The derivation of \mathcal{M} follows the same procedure that is used for \mathcal{L} given below, so it is left to the reader.

To derive \mathcal{L} , first, it is useful to notice that if a volume fraction ξ_α is active during the loading step, then its derivative with respect to strain can be evaluated. Indeed, suppose first the only one transformation surface is active during the loading step and ξ_α is the active variable. This implies that

$$\Phi_\alpha(\boldsymbol{\sigma}, \xi_\alpha) = 0,$$

so ξ_α is an implicit function of $\boldsymbol{\sigma}$. The remaining variables which Φ_α is a function of do not change during the loading step, so they are suppressed. In view of equation (4.25) the last equation can be differentiated by $\boldsymbol{\varepsilon}$ to obtain:

$$\frac{\partial \Phi_\alpha}{\partial \boldsymbol{\sigma}} : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}} + \frac{\partial \Phi_\alpha}{\partial \xi_\alpha} \frac{\partial \xi_\alpha}{\partial \boldsymbol{\varepsilon}} = \mathbf{0}.$$

Therefore, if the 2-nd order tensor Ξ_α is defined as

$$\Xi_\alpha = - \frac{\partial \Phi_\alpha}{\partial \boldsymbol{\sigma}} \Big/ \frac{\partial \Phi_\alpha}{\partial \xi_\alpha}, \quad (4.28)$$

then

$$\frac{\partial \xi_\alpha}{\partial \boldsymbol{\varepsilon}} = \Xi_\alpha : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}$$

is the derivative of ξ_α with respect to $\boldsymbol{\varepsilon}$. Note that transformation surface and its derivatives have a well defined functional form, so this quantity can be evaluated directly for any values of $\boldsymbol{\sigma}$, T and $\boldsymbol{\xi}$.

If, on the other hand, two surfaces are active, say Φ_α and Φ_β , then the appropriate derivatives are calculated in the same way, but this time a 2×2 linear system has to be solved:

$$\begin{bmatrix} \Xi_\alpha \\ \Xi_\beta \end{bmatrix} := - \begin{bmatrix} \frac{\partial \Phi_\alpha}{\partial \xi_\alpha} & \frac{\partial \Phi_\alpha}{\partial \xi_\beta} \\ \frac{\partial \Phi_\beta}{\partial \xi_\alpha} & \frac{\partial \Phi_\beta}{\partial \xi_\beta} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \Phi_\alpha}{\partial \boldsymbol{\sigma}} \\ \frac{\partial \Phi_\beta}{\partial \boldsymbol{\sigma}} \end{bmatrix}, \quad (4.29)$$

and

$$\frac{\partial \xi_\alpha}{\partial \boldsymbol{\varepsilon}} = \boldsymbol{\Xi}_\alpha : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}, \quad (4.30)$$

$$\frac{\partial \xi_\beta}{\partial \boldsymbol{\varepsilon}} = \boldsymbol{\Xi}_\beta : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}. \quad (4.31)$$

Finally, let

$$\boldsymbol{\Xi}_\alpha = \mathbf{0}, \quad (4.32)$$

whenever Φ_α is not an active surface.

To summarize, $\boldsymbol{\Xi}_\alpha$ is given by equation (4.28) if only one transformation is active, by (4.29) if two are active and by (4.32) if Φ_α is not an active transformation surface.

With these definitions, the tangent stiffness \mathcal{L} is given by the formula:

$$\begin{aligned} \mathcal{L} = & [(\Delta \boldsymbol{\mathcal{S}} : \boldsymbol{\sigma} + \Delta \boldsymbol{\alpha}(T_{n+1} - T_0)) \oplus (\boldsymbol{\Xi}_1 + \boldsymbol{\Xi}_2) + \boldsymbol{\mathcal{S}}(\boldsymbol{\xi}) + \boldsymbol{\Lambda}^t \oplus \boldsymbol{\Xi}_2 \\ & + \boldsymbol{\Lambda}^d \oplus \boldsymbol{\Xi}_3 + (\xi_2 - \xi_{2n}) \frac{\partial \boldsymbol{\Lambda}^t}{\partial \boldsymbol{\sigma}} + (\xi_3 - \xi_{3n}) \frac{\partial \boldsymbol{\Lambda}^d}{\partial \boldsymbol{\sigma}}]^{-1} \end{aligned} \quad (4.33)$$

The formula will be proven only for the case when Φ_1 and Φ_2 are active. Checking all other possibilities of active transformation surfaces is left to the reader. Note that the above formula includes derivatives of the transformation flow tensors. Since both $\boldsymbol{\Lambda}^t$ and $\boldsymbol{\Lambda}^d$ are J_2 based, the reader is referred to the work of (Qidwai and Lagoudas, 2000a) for the functional form of these derivatives, as well as the derivation itself.

So, assume, that $\xi_1 - \xi_{1n} \neq 0$, $\xi_2 - \xi_{2n} \neq 0$ and $\xi_3 = \xi_{3n}$ and differentiate equation (4.8) with respect to $\boldsymbol{\varepsilon}$. Note, that for fixed T , all the quantities in this equation are either implicit functions of $\boldsymbol{\varepsilon}$ or enter as constants, e.g. independent of $\boldsymbol{\varepsilon}$. By applying

the product rule and, very carefully, the chain rule, the following result is obtained:

$$\begin{aligned}
\mathbf{0} &= \frac{\partial}{\partial \boldsymbol{\varepsilon}_{n+1}} (\mathcal{S}(\boldsymbol{\xi}) \boldsymbol{\sigma}) - \mathcal{I} + \frac{\partial}{\partial \boldsymbol{\varepsilon}_{n+1}} (\boldsymbol{\alpha}(\boldsymbol{\xi}) (T_{n+1} - T_0)) + \frac{\partial}{\partial \boldsymbol{\varepsilon}_{n+1}} ((\xi_2 - \xi_{2n}) \boldsymbol{\Lambda}^t(\boldsymbol{\sigma})) \\
&= \frac{\partial \mathcal{S}(\boldsymbol{\xi})}{\partial \boldsymbol{\varepsilon}_{n+1}} : \boldsymbol{\sigma} + : \mathcal{S}(\boldsymbol{\xi}) : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}_{n+1}} - \mathcal{I} + \frac{\partial \boldsymbol{\alpha}(\boldsymbol{\xi})}{\partial \boldsymbol{\varepsilon}_{n+1}} (T_{n+1} - T_0) \\
&\quad + \left(\frac{\partial \xi_2}{\partial \boldsymbol{\varepsilon}_{n+1}} \right) \oplus \boldsymbol{\Lambda}^t(\boldsymbol{\sigma}) + \left((\xi_2 - \xi_{2n}) \frac{\partial \boldsymbol{\Lambda}^t(\boldsymbol{\sigma})}{\partial \boldsymbol{\varepsilon}_{n+1}} \right) \\
&= \left(\frac{\partial \mathcal{S}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \cdot \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\varepsilon}_{n+1}} \right) : \boldsymbol{\sigma} + (T_{n+1} - T_0) \frac{\partial \boldsymbol{\alpha}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \cdot \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\varepsilon}_{n+1}} + \mathcal{S}(\boldsymbol{\xi}) : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}_{n+1}} - \mathcal{I} \\
&\quad + \left(\boldsymbol{\Xi}_2 : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}_{n+1}} \right) \oplus \boldsymbol{\Lambda}^t(\boldsymbol{\sigma}) + \left((\xi_2 - \xi_{2n}) \frac{\partial \boldsymbol{\Lambda}^t(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}_{n+1}} \right), \tag{4.34}
\end{aligned}$$

where \mathcal{I} is the fourth-order identity tensor⁴. Now, using equation (3.38), (4.30) and (4.31), observe that:

$$\begin{aligned}
&\left(\frac{\partial \mathcal{S}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \cdot \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\varepsilon}_{n+1}} \right) : \boldsymbol{\sigma} + (T_{n+1} - T_0) \frac{\partial \boldsymbol{\alpha}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \cdot \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\varepsilon}_{n+1}} = \\
&\quad (\Delta \mathcal{S} : \boldsymbol{\sigma} + (T_{n+1} - T_0) \Delta \boldsymbol{\alpha}) \oplus \left(\frac{\partial \xi_1}{\partial \boldsymbol{\varepsilon}_{n+1}} + \frac{\partial \xi_2}{\partial \boldsymbol{\varepsilon}_{n+1}} \right) = \\
&\quad [(\Delta \mathcal{S} : \boldsymbol{\sigma} + (T_{n+1} - T_0) \Delta \boldsymbol{\alpha}) \oplus (\boldsymbol{\Xi}_1 + \boldsymbol{\Xi}_2)] : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}. \tag{4.35}
\end{aligned}$$

Therefore, by combining with (4.34), rearranging the terms, and factoring $\partial \boldsymbol{\sigma} / \partial \boldsymbol{\varepsilon}$ the following, hopefully correct, identity is obtained:

$$\begin{aligned}
\mathcal{I} &= [(\Delta \mathcal{S} : \boldsymbol{\sigma} + (T_{n+1} - T_0) \Delta \boldsymbol{\alpha}) \oplus (\boldsymbol{\Xi}_1 + \boldsymbol{\Xi}_2) + \mathcal{S}(\boldsymbol{\xi}) \\
&\quad + \boldsymbol{\Lambda}^t \oplus \boldsymbol{\Xi}_2 + (\xi_2 - \xi_{2n}) \frac{\partial \boldsymbol{\Lambda}^t(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}}] : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}. \tag{4.36}
\end{aligned}$$

⁴That is, $\mathcal{I} : \mathbf{v} = \mathbf{v}$ for any second order tensor \mathbf{v} , or in indicial notation, $\mathcal{I} = I_{ijkl} = \delta_{ik} \delta_{jl}$.

After inverting the fourth order tensor⁵, the final result is obtained:

$$\begin{aligned} \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}} = & [(\Delta \boldsymbol{\mathcal{S}} : \boldsymbol{\sigma} + (T_{n+1} - T_0) \Delta \boldsymbol{\alpha}) \oplus (\boldsymbol{\Xi}_1 + \boldsymbol{\Xi}_2) \\ & + \boldsymbol{\mathcal{S}}(\boldsymbol{\xi}) + \boldsymbol{\Lambda}^t \oplus \boldsymbol{\Xi}_2 + (\xi_2 - \xi_{2n}) \frac{\partial \boldsymbol{\Lambda}^t(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}}]^{-1}. \end{aligned} \quad (4.37)$$

Remember that $\xi_3 = \xi_{3n}$, that is Φ_3 is not an active surface and hence by the definition (4.32), $\boldsymbol{\Xi}_3$, it is identically zero. Therefore, (4.33) reduces to (4.37) when Φ_1 and Φ_2 are the active surfaces.

As a final remark, observe that the discretized flow rule (4.6) enters in the expression for \mathcal{L} and as a result the later includes a dependency $\xi_{i_{n+1}} - \xi_{i_n}$, $i = 2, 3$, as well as the derivative of the transformation flow tensors. A different discretization of (4.2) would have resulted in a different flow rule, so \mathcal{L} is algorithmic specific, hence the name *algorithmic tangent*. As can be seen from Appendix B, this is the precise derivative required when solving the nonlinear FEM equations. For a further discussion, see Simo and Hughes (1998).

2. Numerical examples

The numerical examples considered in this section were all solved using a displacement based Finite Element Method (FEM). A short summary of the method for nonlinear

⁵The author of this thesis has, in the early stages of the research, been puzzled by the question "what is the inverse of a fourth-order tensor?". After all, second-order tensors are, well, "just matrices", so one knows what is an inverse. The definition of a fourth-order states that it is a linear mapping of the space of second order tensors, so therefore, the inverse is the inverse mapping. To visualize this, index notation helps. Say $\mathcal{A} = A_{ijkl}$ is the fourth order tensor and the tensor $\mathbf{v} = v_{ij}$ is mapped by contraction of the last two indices: $A_{ijkl}v_{kl}$. Well, map uniquely the nine pairs (i, j) to an index $I = 1, 2, \dots, 9$ (for example, $I = 3 * (i - 1) + j$) and (k, l) to an index $J = 1, 2, \dots, 9$ and observe that \mathcal{A} becomes the 9×9 matrix A_{IJ} , which we know how to invert.

problems⁶ and the integration of the closest point projection method of the previous section is given in Appendix B. The SMA material subroutine⁷ implementing the closest point projection algorithm was written in the C++ programming language and is provided in Appendix C. All discretizations used triangular/tetrahedral meshes and the finite element spaces used were the usual linear Lagrangian ones. The SMA material properties used again represent the generic SMA material given in Table IV.

2.1. Constrained heating of a perforated square

In this section we consider constrained cooling of a perforated square cross-section (Figure 32). The SMA material is initially in the self-accommodated phase, i.e. $c_1 = 1$ everywhere in the domain. The thermomechanical loading that the square is subjected to is shown on Figure 33. The SMA is first loaded mechanically at isothermal temperature $T = 260 \text{ }^\circ K$ as follows: the left side of the square is fixed against horizontal displacement, but is allowed to move in the vertical direction; the right side is pulled uniformly by the amount of $0.002m$ in the horizontal direction and the side is again allowed to move in the vertical direction; the remaining part of the boundary (including the hole) is stress free. The second loading step consists of keeping the horizontal component of the displacement fixed and raising the temperature to $T = 350 \text{ }^\circ K$.

⁶The FEM software used and all other related components (excluding mesh generation) such as numerical linear algebra subroutines and graphical post-processing tools were developed by the author. They were implemented in the *Object Pascal* programming language (except the material subroutine, which is written in C++) and compiled with *Borland[®] Delphi[™], version 2.01*. The *Triangle, version 1.4*, mesh generating software was used for meshing 2-D domains (Shewchuk, 2002) and *the Netgen, version 4.0*, was used for meshing 3-D domains. *Netgen* is developed by Joachim Schöberl and can be downloaded (as of March 2005) from <http://www.hpfem.jku.at/netgen/>.

⁷Historical note: **subroutine** is the keyword for a function in FORTRAN, an archaic programming language.

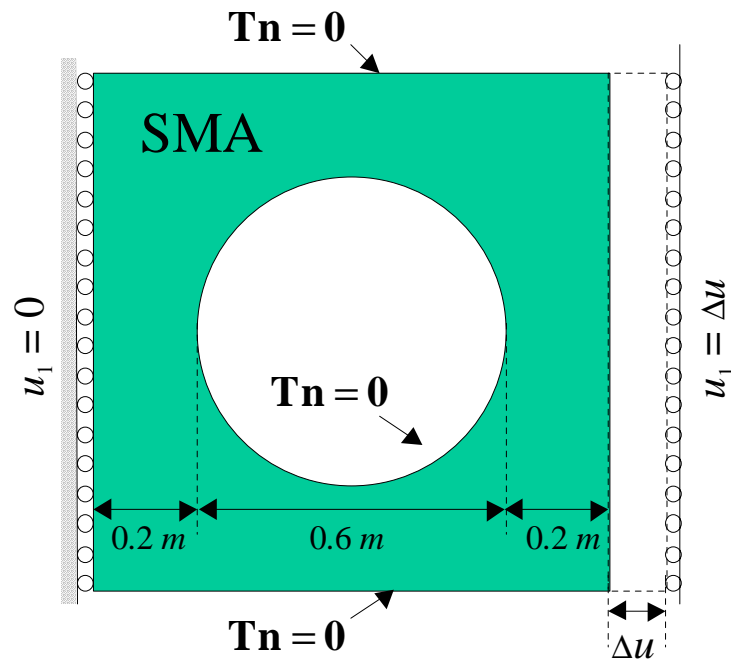


Fig. 32. Schematic of the mechanical boundary conditions applied for the perforated square model problem.

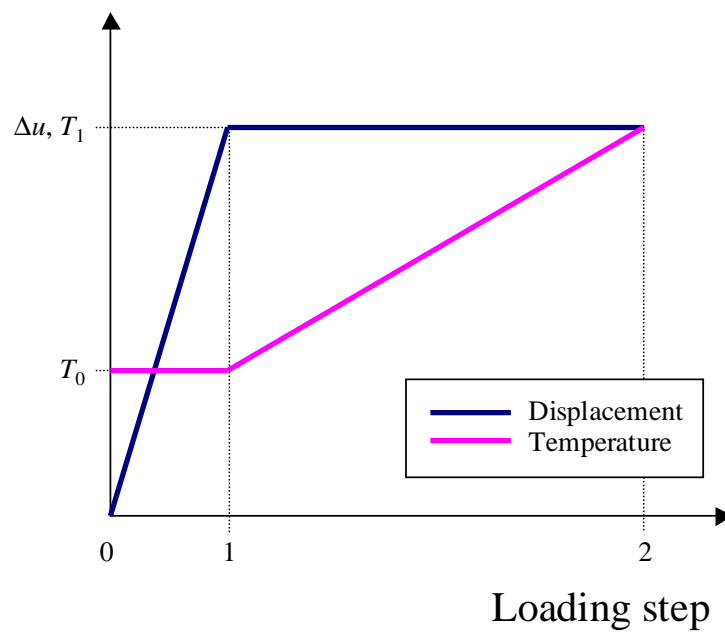


Fig. 33. Schematic of the loading path followed for the boundary conditions.

The first loading step was used to determine a suitable mesh size for the entire simulation. This was done by starting with a very coarse mesh (Figure 34) and consecutively refining it (Figures 35–38). Due to obvious symmetry consideration, only one quarter of the domain was used in the calculations. The stresses are shown as is, i.e. piecewise constant over each element. Since no error estimator was implemented in 2D, the solution is judged to be accurate enough when the pictures become smooth enough. That is, the relative difference between stresses in neighboring elements is small in the eye-ball norm. While the solution shown on Figure 38 (110793 elements, 111994 DOF) is clearly the best, in complicated nonlinear problems the cost of assembly and the memory requirements needed to save the material state at each integration point make it desirable to keep the number of elements at a minimum. In this respect, the second refinement (8964 elements, 9274 DOF) is acceptable enough and was used in the rest of the simulations.

During the first loading step, stress concentrations developed near the top and bottom edge of hole. The effective stress in these locations become sufficiently high for the detwinning of small amounts of self-accommodated martensite as shown on Figure 39.

When the second loading step begins, the material experiences initial linear thermoelastic expansion. Since the conditions are of plane strain, and the prescribed displacements are fixed, this results in a nonhomogeneous change in the stress state. As the critical temperature for the $M^t \rightarrow A$ transformation is reached, the self-accommodated martensite begins to transform to austenite. The stiffness of austenite is approximately 2.3 times that of martensite (see Table IV) and due to the boundary condition the stresses increase throughout the square. This causes further detwinning of martensite in some areas of the square, resulting in a simultaneous $M^t \rightarrow A, M^d$ transformation, mostly near the top and bottom parts of the hole (Figure 40). This is

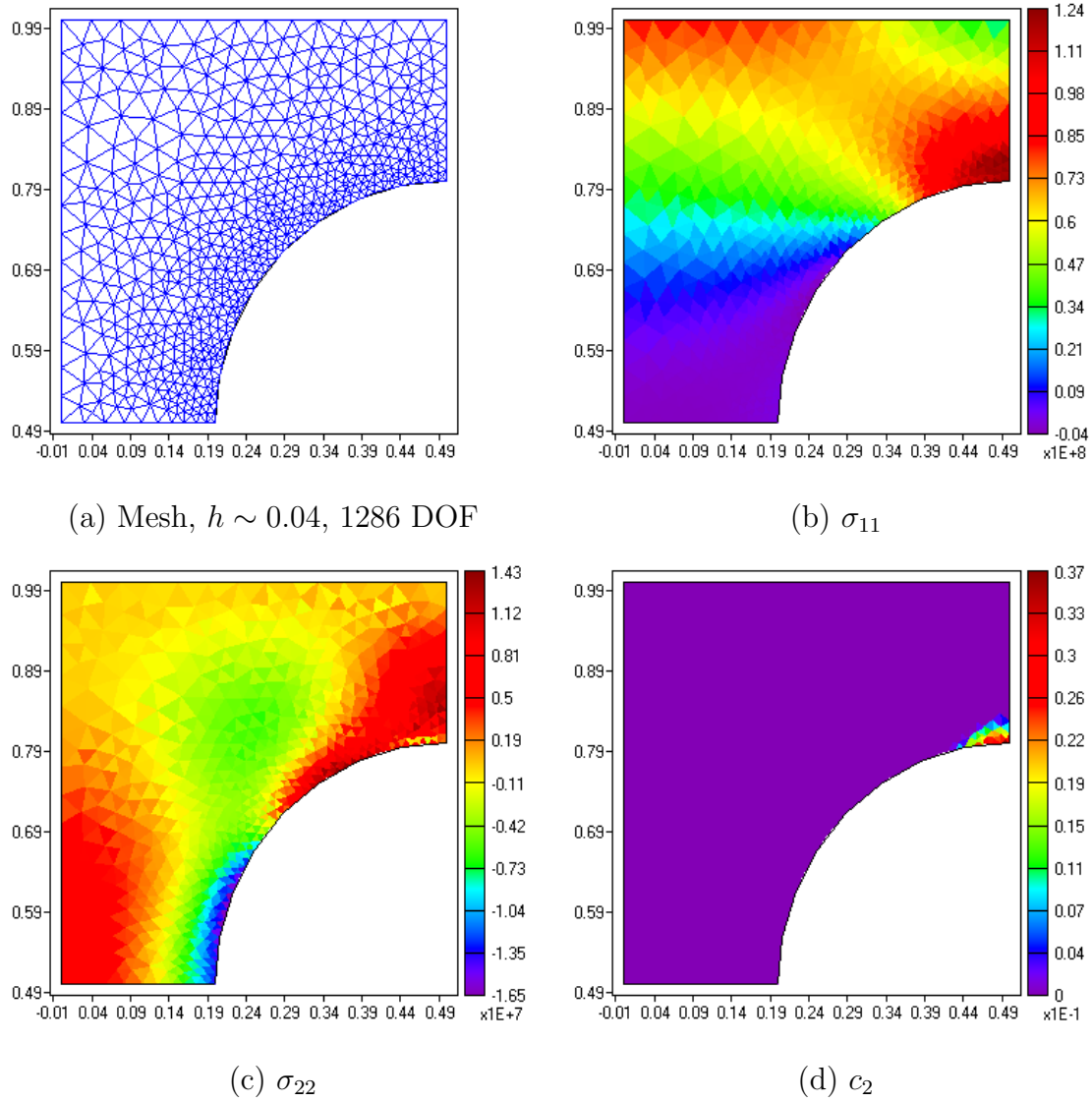


Fig. 34. Basic mesh used to perform the first loading step, $T = 260$.

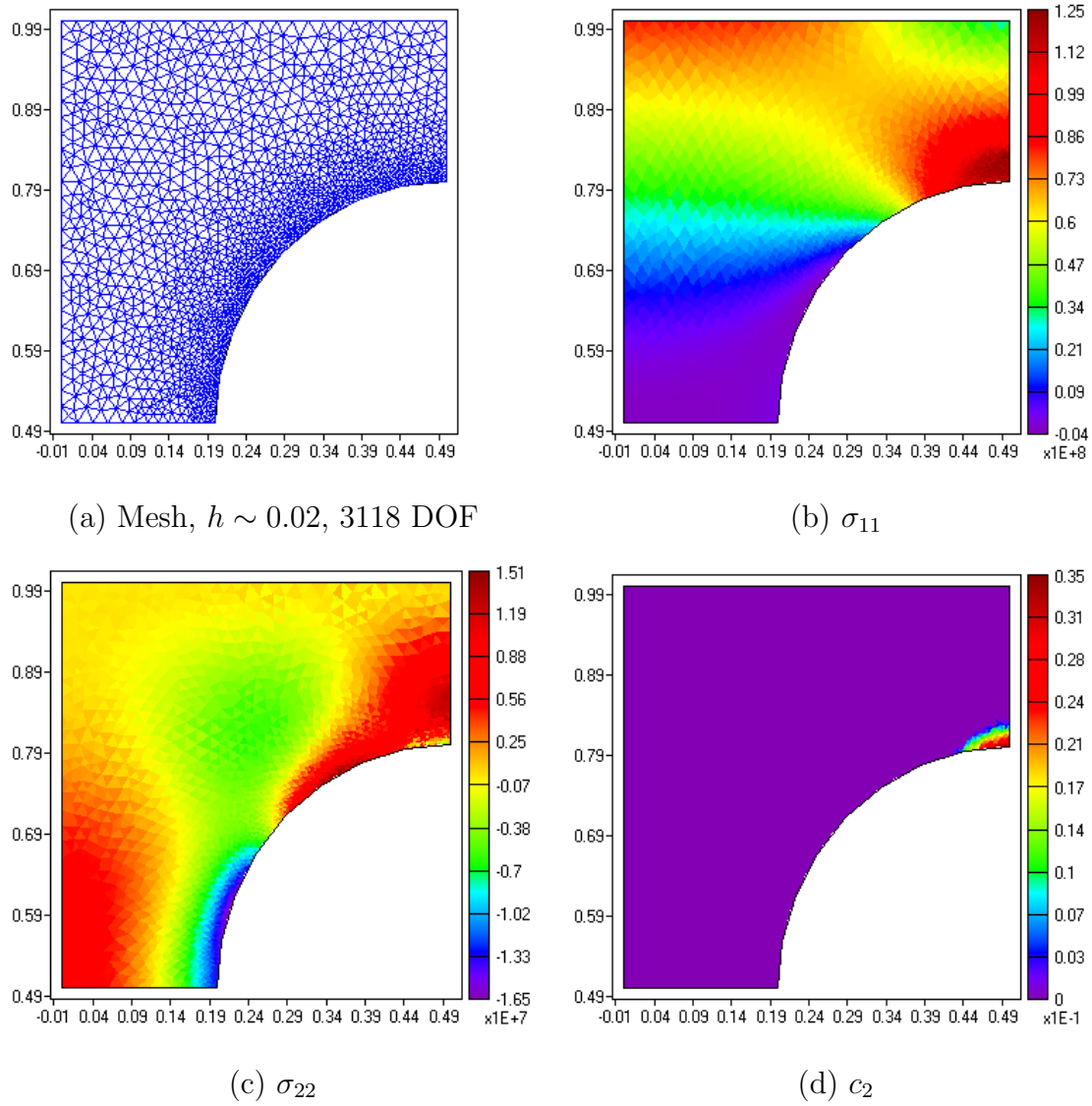


Fig. 35. First refinement, $T = 260$, end of first loading step.

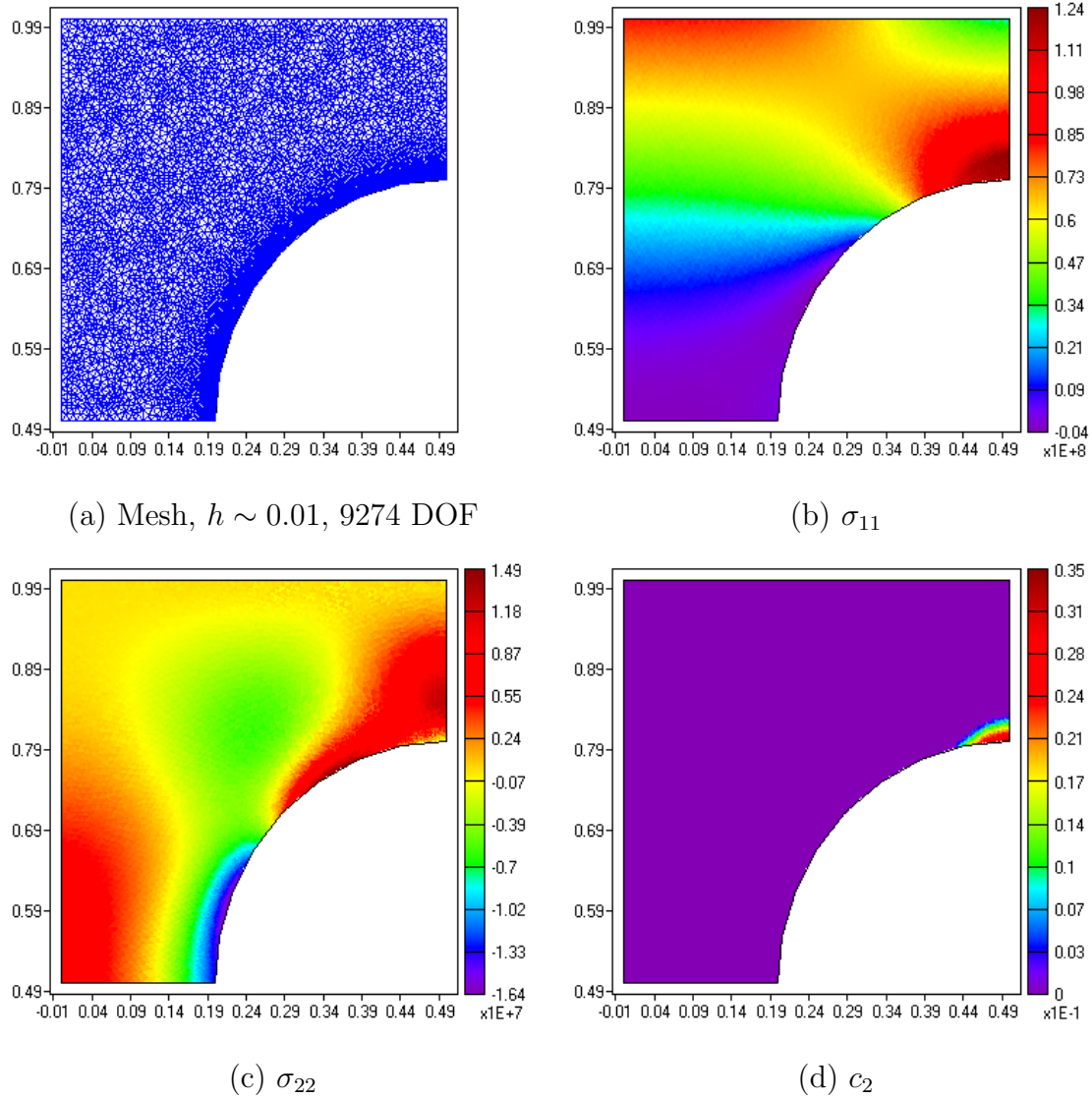
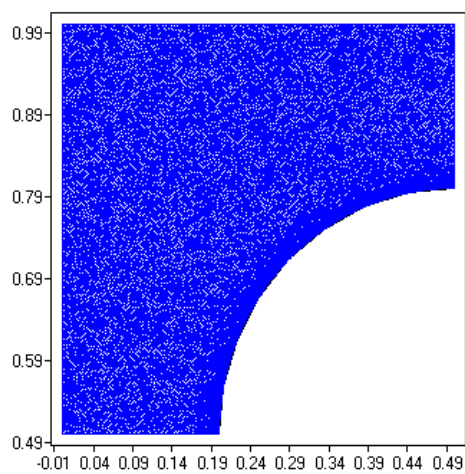
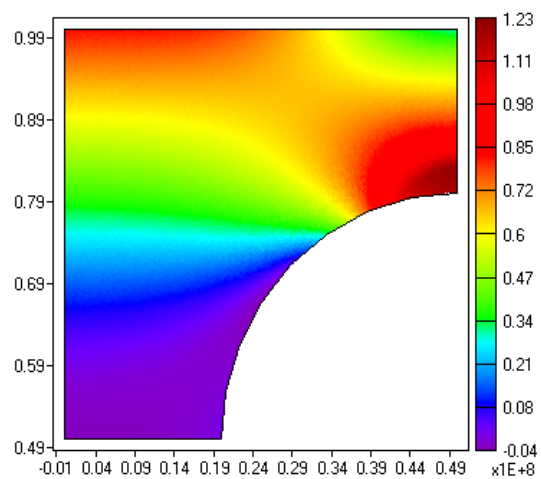
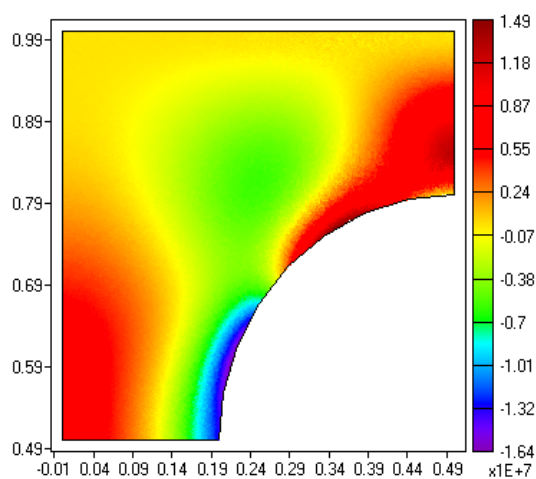
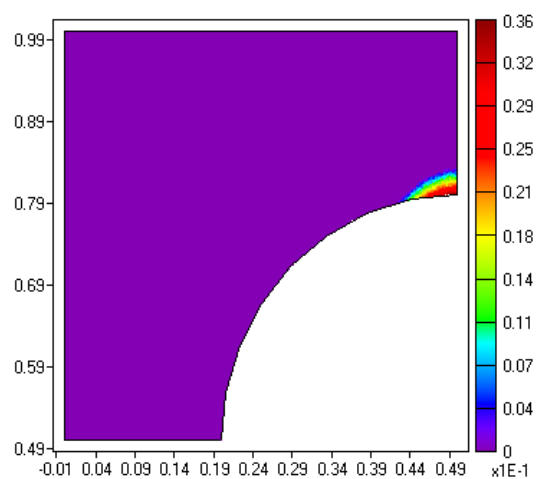


Fig. 36. Second refinement, $T = 260$, end of first loading step. This mesh was selected for the rest of the simulation.

(a) Mesh, $h \sim 0.005$, 31054 DOF(b) σ_{11} (c) σ_{22} (d) c_2 Fig. 37. Third refinement, $T = 260$, end of first loading step.

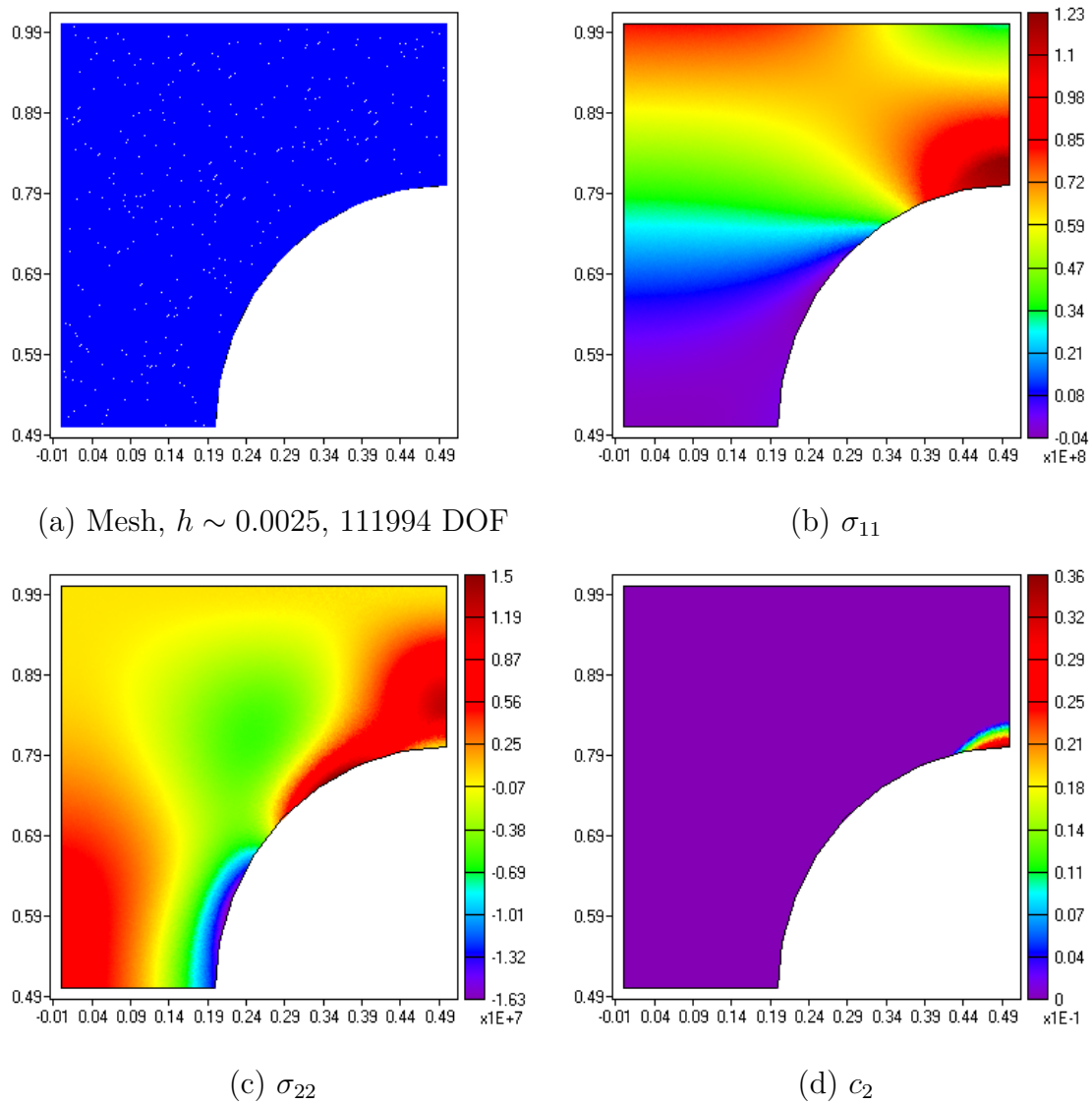


Fig. 38. Fourth refinement, $T = 260$, end of first loading step.

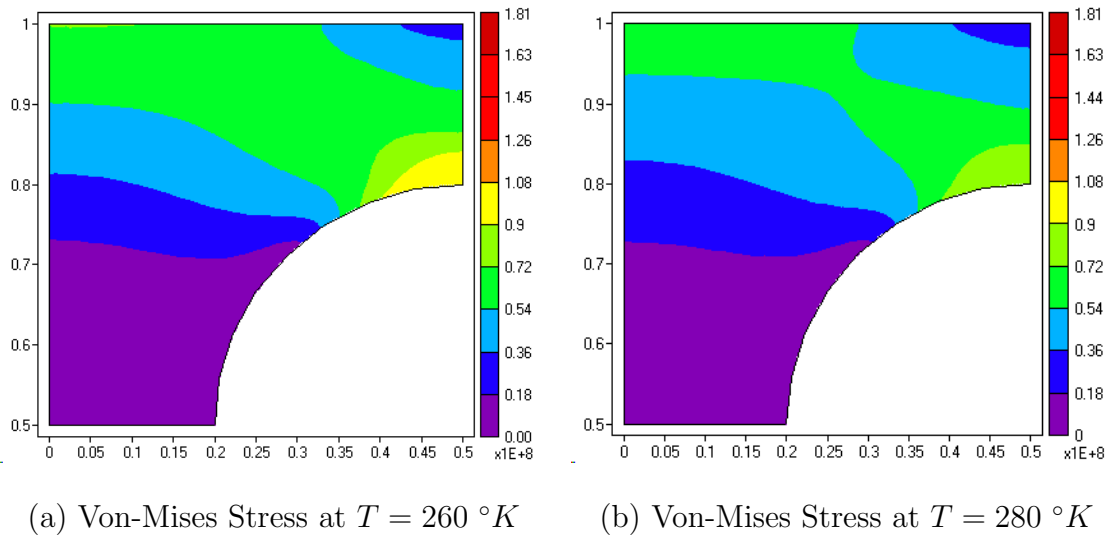


Fig. 39. Solution for square hole problem at $T = 260 \text{ }^\circ\text{K}$ and $T = 280 \text{ }^\circ\text{K}$. Shows the evolution of the effective Von-Mises stress at the beginning of the heating loading step. The stress decreases due to thermal expansion of the material.

manifested in further increase in c_2 in comparisons to the amount that was produced during the first loading step. The effective stress and c_2 after the completion of the $M^t \rightarrow A$ transformation are shown on Figure 41.

The areas, where detwinned martensite is present have generally higher effective stress (above σ_s), compared to the rest of the domain. Hence, the reverse transformation of detwinned martensite ($M^d \rightarrow A$) does not happen until much higher temperature, due to the fact the corresponding transformation surfaces exhibit stress dependence (see Figure 27). Around $T = 335 \text{ }^\circ\text{K}$, the $M^d \rightarrow A$ transformation begins in areas with lowest effective stress. During this phase, the inelastic strains decrease according to the transformation rule (3.9),(3.67). Again, due to the constraint on the displacements, imposed by the boundary conditions, the elastic portion of the stress generally increases which leads to a corresponding (non-uniform) increase in the stress during the reverse transformation (Figures 42-45).

2.2. A 3-D structural member

In order to test the numerical implementation in 3-D, a structural member is considered. The geometry, part of a periodic planar arrangement, is shown in Figure 47. This type of SMA structural members have various applications in active deformable surfaces such as self-expanding medical stents and grafts (Jung et al., 2004), impact absorbing meshes or filters with varying cross-section.

Two loading steps are considered, similar to the ones of the previous section. In the first, the SMA structural member is compressed in the x direction until the displacements at side B (see Figure 47) reach $0.2mm$. This step corresponds to the initial shrinkage of the stent. The second loading step consists of keeping the displacements fixed and raising the temperature by $100\text{ }^\circ K$. The initial condition is twinned martensite.

The two loading steps are of the same type as the previous example, so the same type of behavior is observed as in the perforated square block. Initially, (see Figures 48 and 49) small amount of M^d is produced from M^t , reaching a maximum of about 12%. Heating causes first an increase of stiffness due to the $M^t \rightarrow A$ transformation and, due to the boundary conditions, this results in an increased stress. The later causes some additional $M^d \rightarrow A$ transformation, which, as in the previous example, occurs simultaneously with the $M^t \rightarrow A$ one. The maximum amount of the detwinned volume fraction is achieved after the depletion of M^t and is about 22% (Figure 50). The reverse, $M^d \rightarrow A$ transformation occurs at about $337\text{ }^\circ K$ and leads to sharp rise in the stress (Figures 51). By $360\text{ }^\circ K$ (Figure 52) a substantial part of the the detwinned martensite has transformed back to austenite. The evolution of the maximal effective (Von-Mises) stress in the stent is shown on Figure 53.

This example demonstrates that the numerical implementation behaves in a ro-

bust way in both two and three dimensions and under multiple simultaneous/consecutive transformations.

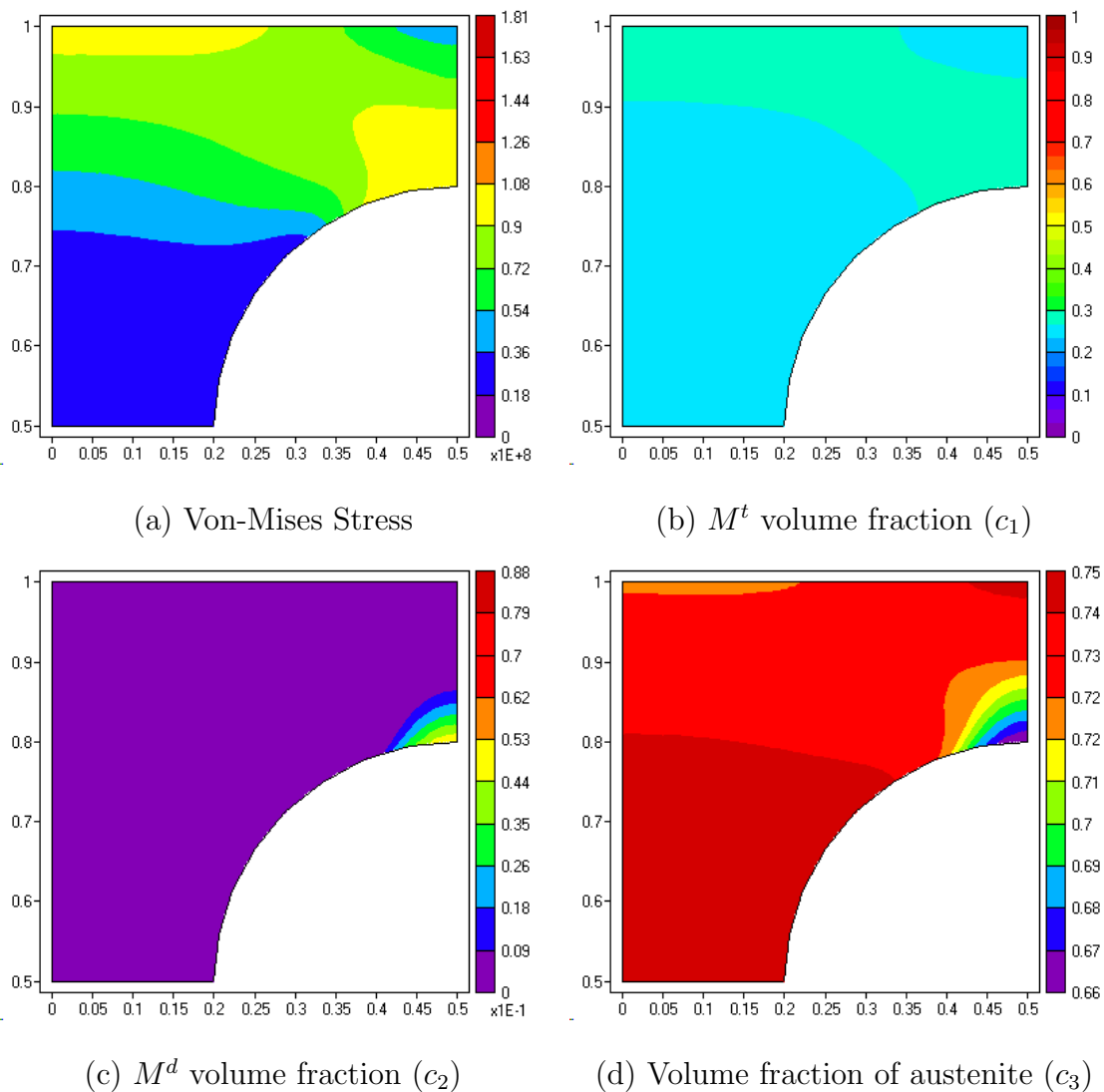


Fig. 40. Solution for square hole problem at $T = 310 \text{ }^\circ\text{K}$. As the material is slowly heated, the reverse transformation $M^t \rightarrow A$ occurs (c). Since the austenite is considerably stiffer than martensite, and the material is constrained, the stresses also increase. This results in a simultaneous $M^t \rightarrow A, M^d$ transformation, which is manifested in increase in the volume fraction of M^d (a,b).

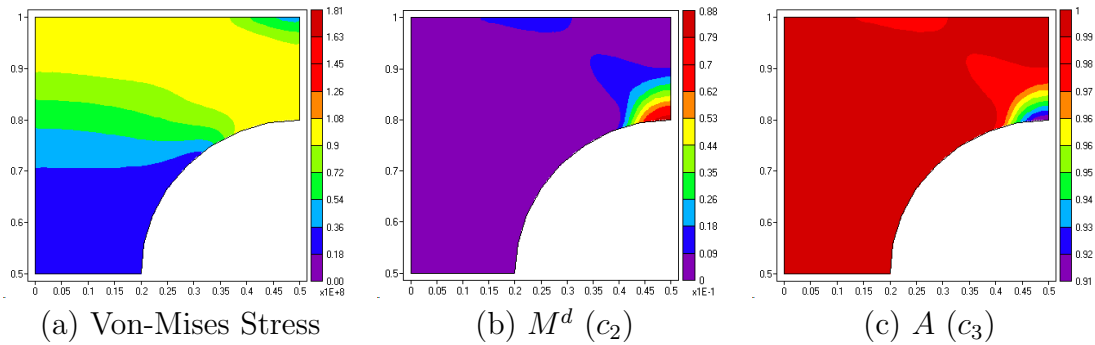


Fig. 41. Solution for square hole problem at $T = 320 \text{ }^\circ K$. At this temperature, M^t is already depleted. The maximum effective stress is approximately $104 MPa$. The volume fraction of M^d (b) has reached approximately 8.8% and the rest is in the A phase (c). Note, that Figures 41-46 are drawn to the same scale.)

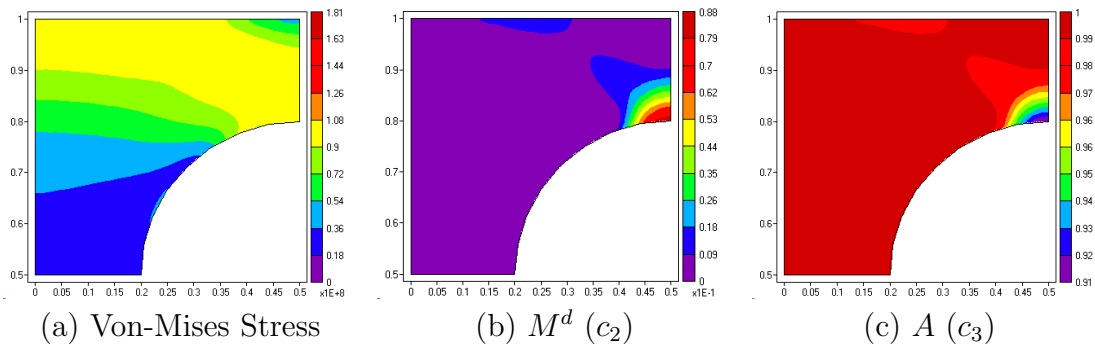


Fig. 42. Solution for square hole problem at $T = 330 \text{ }^\circ K$. At $T = 330 \text{ }^\circ K$ the heating has caused only thermal expansion of the material. Due to the plane strain conditions, this results in non-proportional change of σ_{11} and σ_{22} in comparisons to σ_{33} . Subsequently, the Von-Mises stress has changed and its maximum is now $97 MPa$.

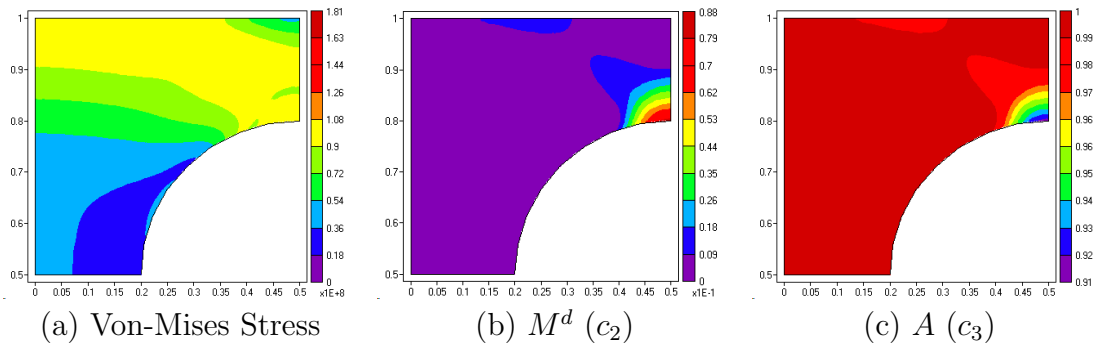


Fig. 43. Solution for square hole problem at $T = 335 \text{ }^\circ K$. Now the temperature has risen just enough to activate the reverse, $M^d \rightarrow A$, transformation. The maximum effective stress is approximately 96 MPa and c_2 has decreased to 8.2% .

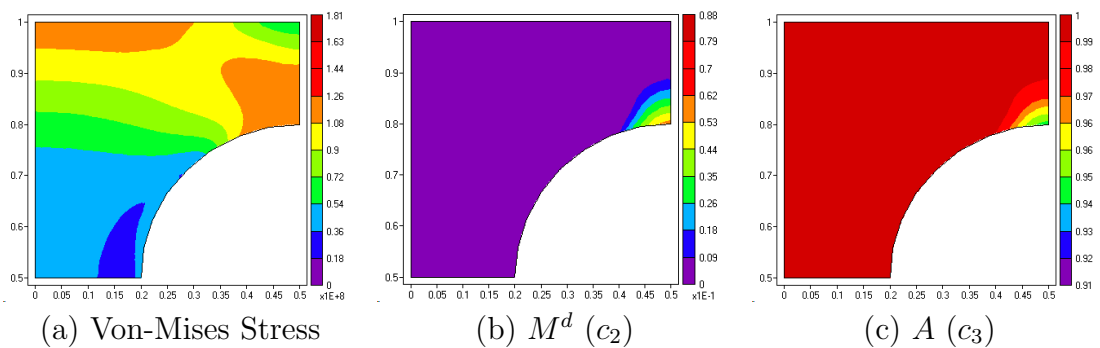


Fig. 44. Solution for square hole problem at $T = 340 \text{ }^\circ K$. Further heating causes further $M^d \rightarrow A$ transformation and, therefore, a decrease in c_2 (b). The maximum value for c_2 is now 5.6% . The inelastic strains (not shown) also decrease (see equations (3.9 and (3.67))) and, due to the constrained displacement the stresses (a) begin to increase compared to previous temperatures. The maximum value for the Von-Mises stress (a) is now 115 MPa .

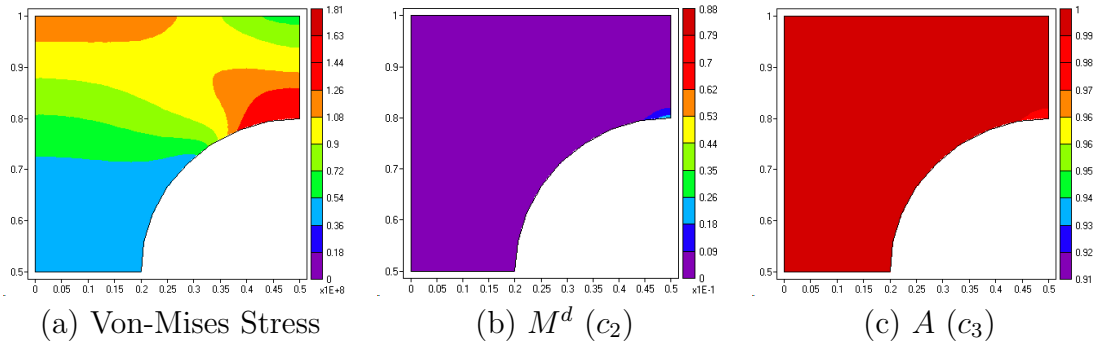


Fig. 45. Solution for square hole problem at $T = 350 \text{ }^\circ\text{K}$. By this temperature, the $M^d \rightarrow A$ is complete almost everywhere in the domain (c) with the maximum for c_2 being 0.7%. Note that the maximum value for the effective stress has increased to approximately 166MPa (a), thanks to the boundary conditions and decrease in the inelastic strains.

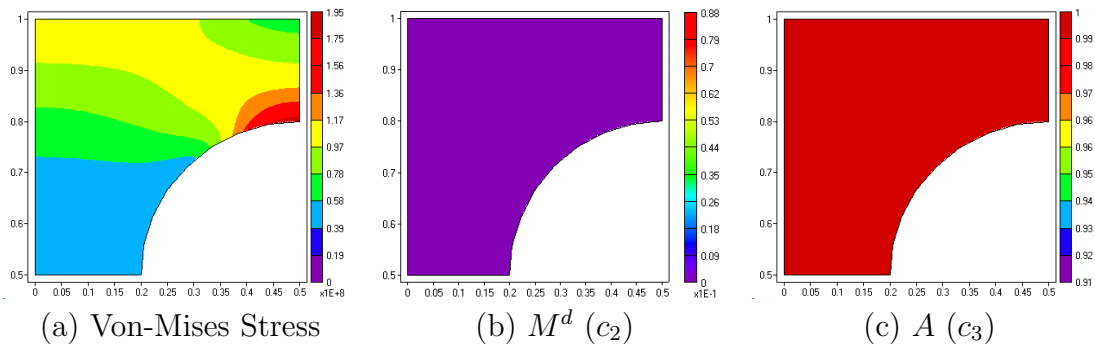


Fig. 46. Solution for square hole problem at $T = 360 \text{ }^\circ\text{K}$. The $M^d \rightarrow A$ is now complete everywhere and the maximum in the domain (c). The maximum value for the effective stress has increased to approximately 181MPa (a).

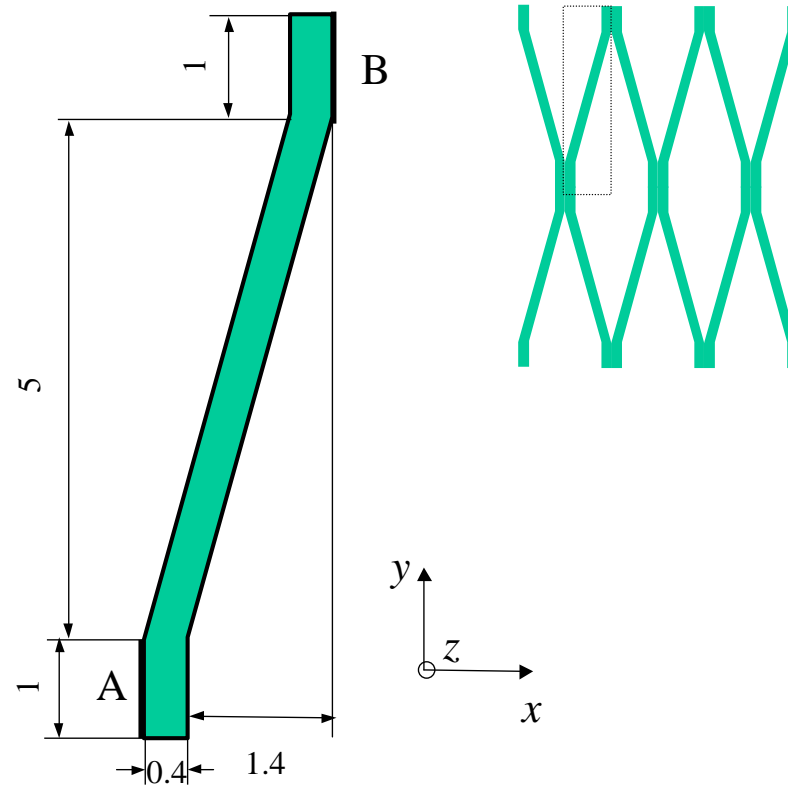


Fig. 47. Schematic of the 3-D SMA structural member geometry. The member is cut from a planar SMA sheet. The cutting pattern is shown on the top left part of the figure. The unit cell geometry used for the model problem is shown in 2D projection. All dimensions are in generic. The out of plane thickness is 0.4. All displacements on side *A* are fixed, and side *B* is displaced by 0.2 and allowed to slide in the *y* and *z* directions.

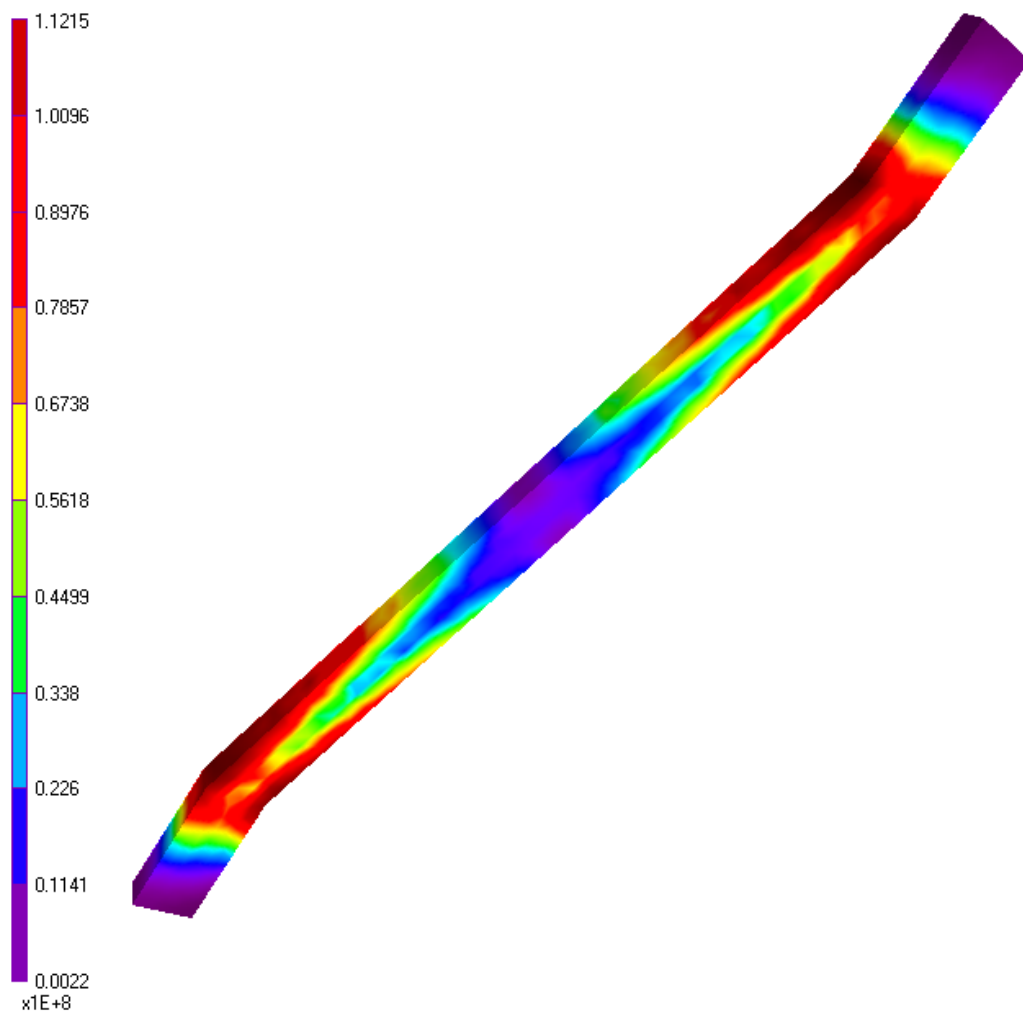


Fig. 48. Von-Mises stress in the 3-D SMA structural member at $T = 260 \text{ }^\circ\text{K}$.

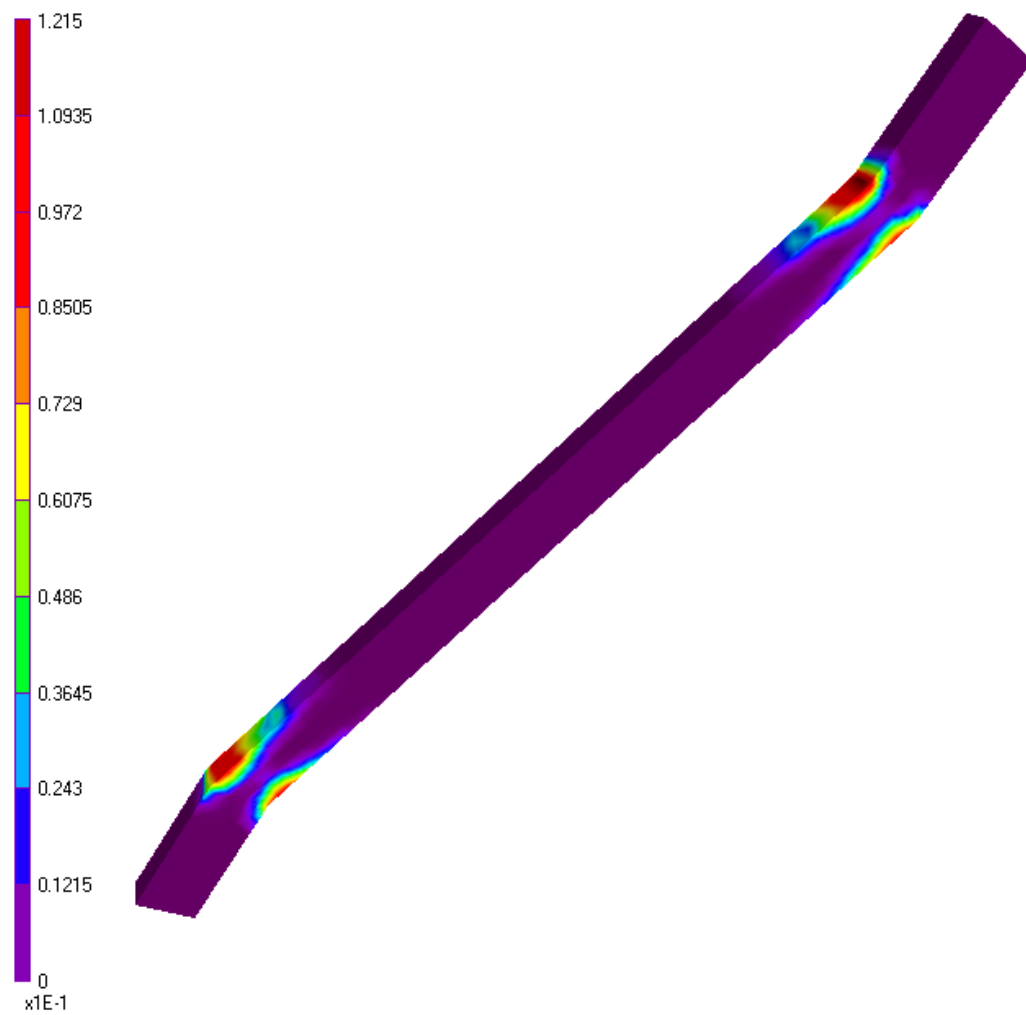


Fig. 49. M^d volume fraction in the 3-D SMA structural member at $T = 260 \text{ }^\circ\text{K}$.

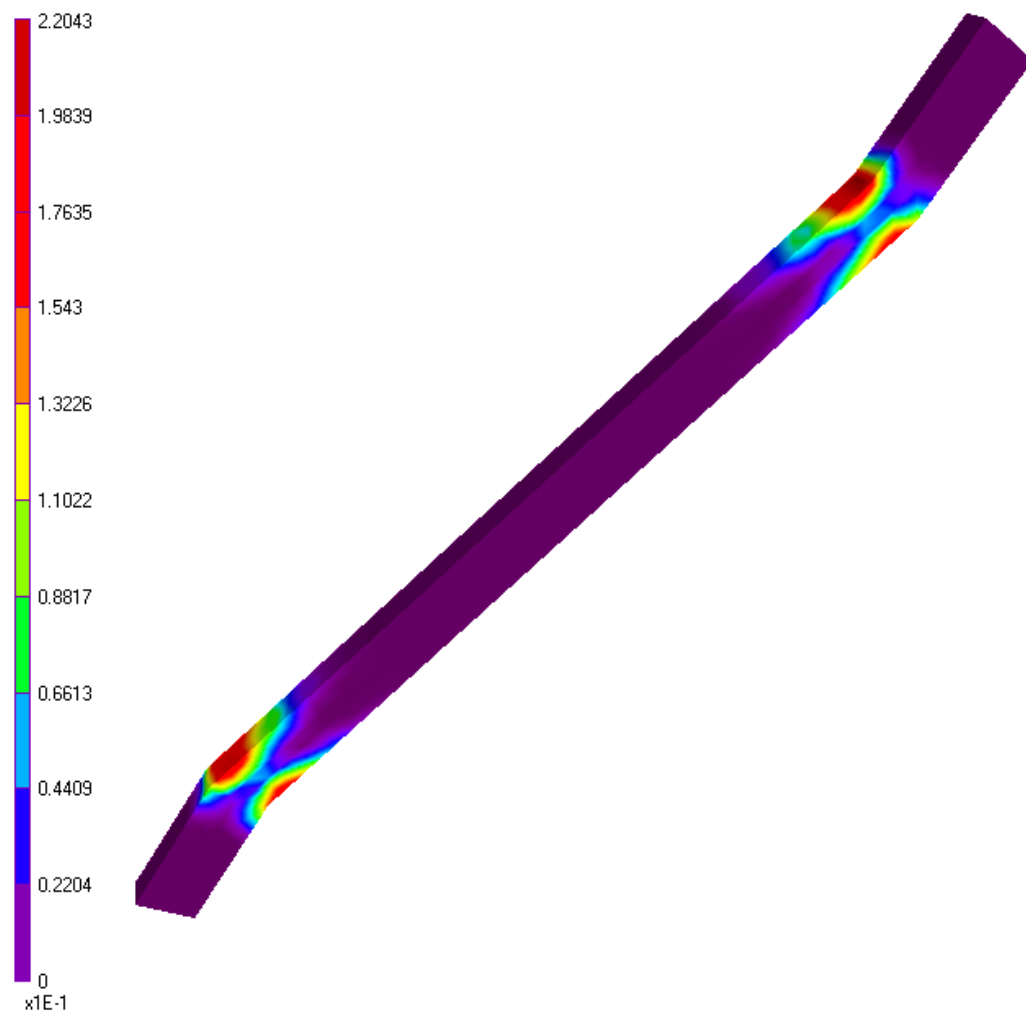


Fig. 50. M^d volume fraction in the 3-D SMA structural member at $T = 330 \text{ }^\circ\text{K}$.

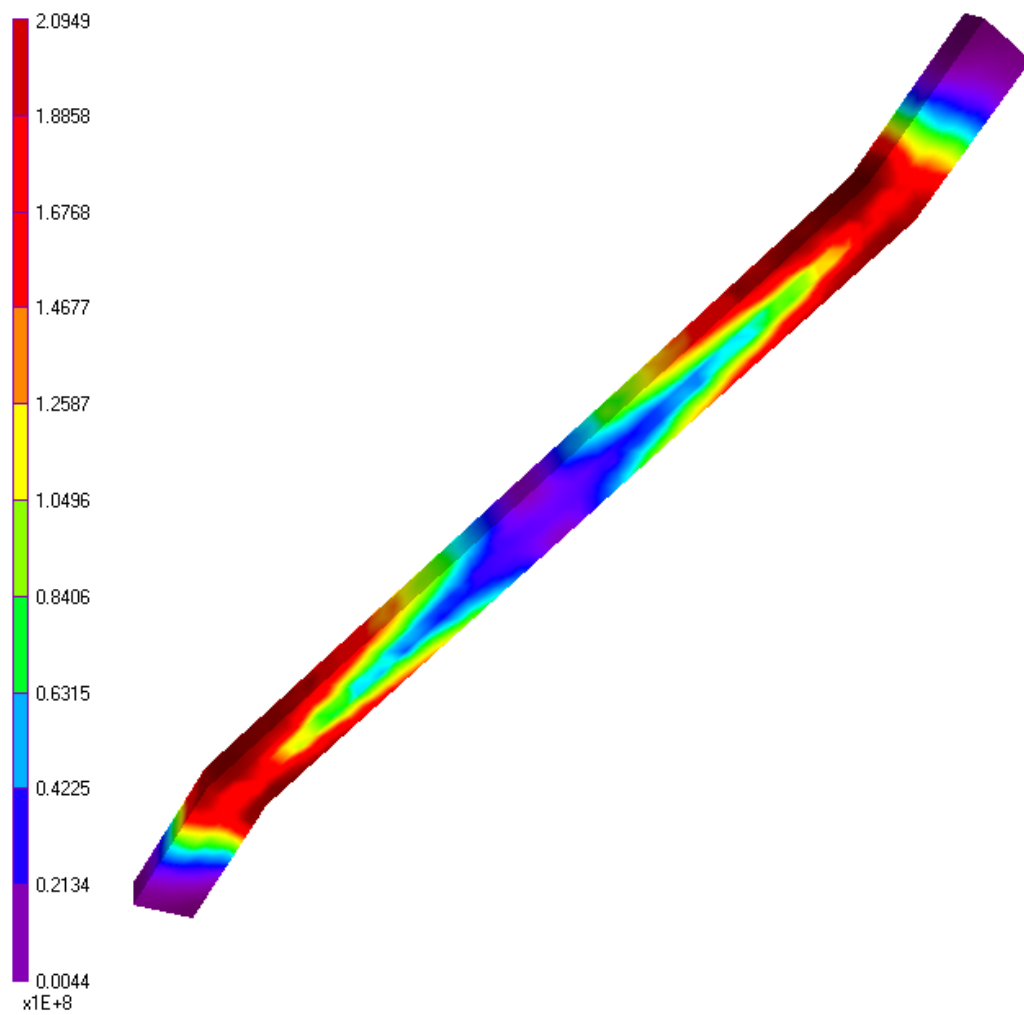


Fig. 51. Von-Mises stress in the 3-D SMA structural member at $T = 360 \text{ }^\circ\text{K}$.

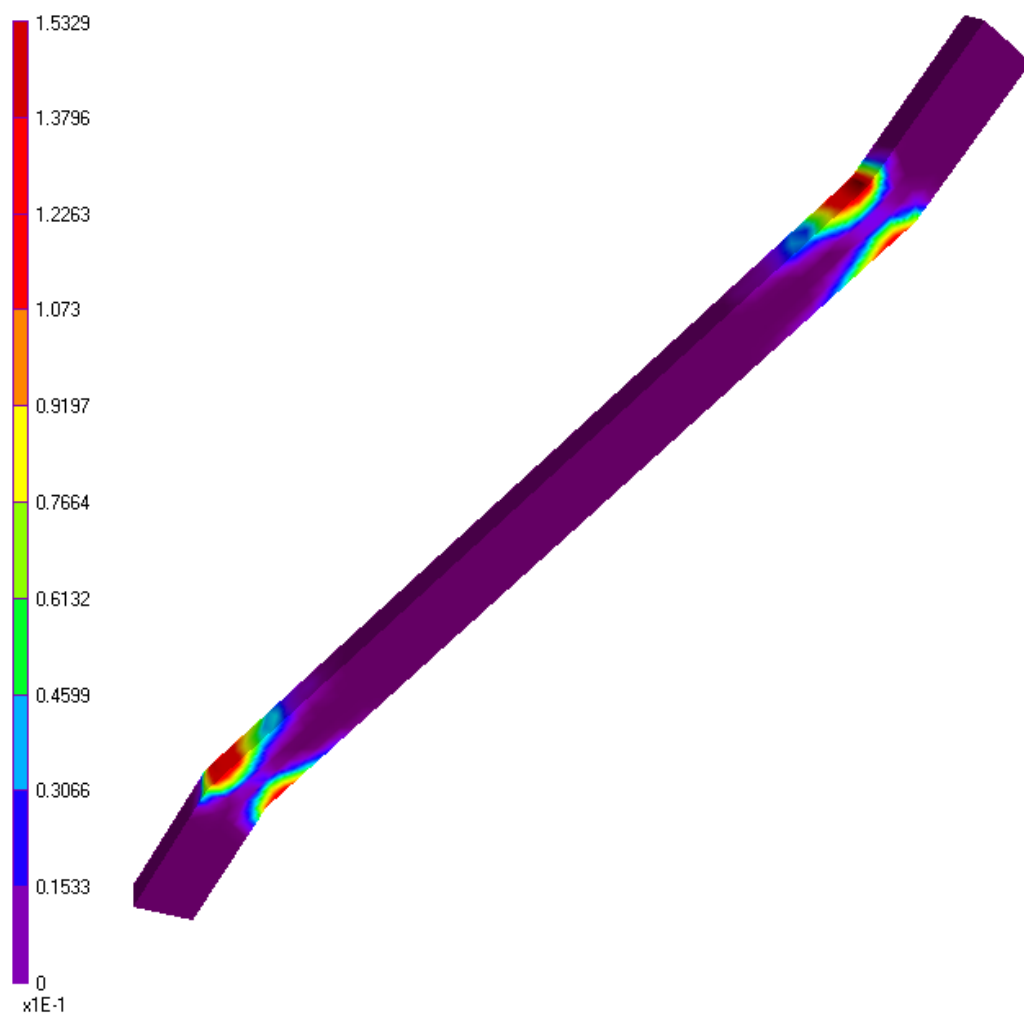


Fig. 52. M^d volume fraction in the 3-D SMA structural member at $T = 360 \text{ }^\circ\text{K}$.

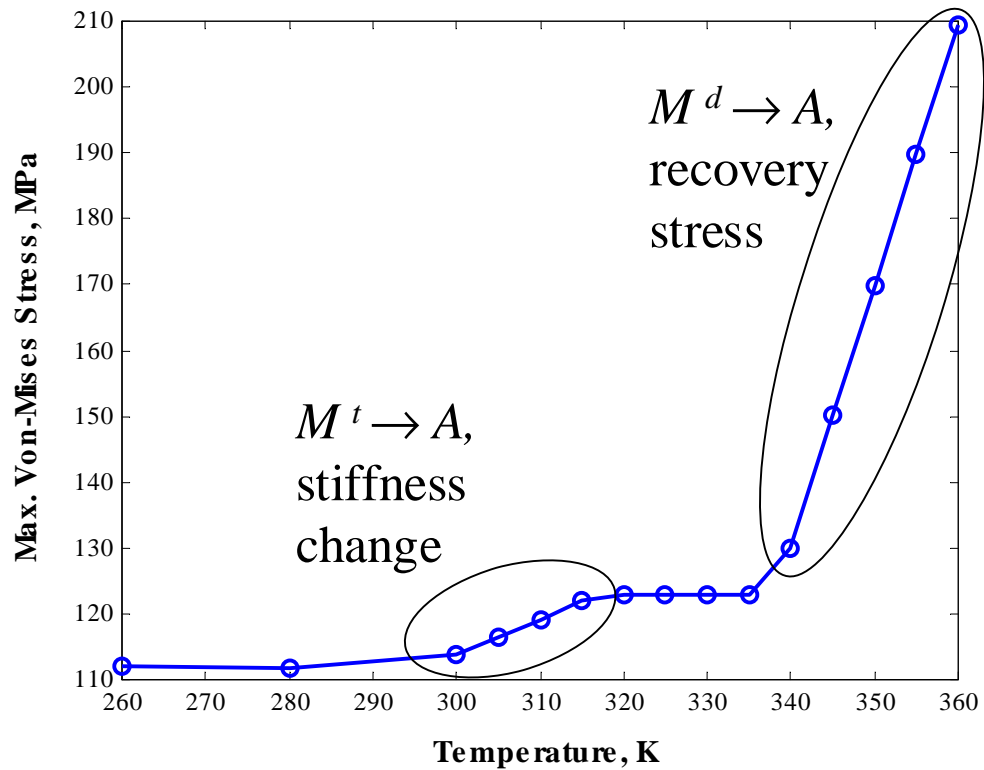


Fig. 53. Evolution of the effective (Von-Mises) stress during heating. Plotted is the maximal value. The first region, where the stress increases is between $295\text{ }^{\circ}\text{K}$ and $315\text{ }^{\circ}\text{K}$ and is caused by change in stiffness due to the $M^t \rightarrow A$ transformation. The second region, where the stress increases happens at $T \geq 335\text{ }^{\circ}\text{K}$, where the transformation strain decreases due to the $M^d \rightarrow A$ transformation. Because of the fixed displacement boundary conditions, in both cases the stress increases.

CHAPTER V

FLUID-STRUCTURE INTERACTION PROBLEMS WITH APPLICATIONS TO
SMA BASED DEVICES

In this chapter the problem of fluid flow in deformable porous media is studied. First, the stationary fluid-structure interaction (FSI) problem is formulated in terms of incompressible Newtonian fluid and a linearized elastic solid. The flow is assumed to be characterized by very low Reynolds number and is described by the Stokes equations. The strains in the solid are assumed small, however, no restrictions are applied on the magnitude of the displacements. Thus, even when linearized elasticity is used to describe the solid, the resulting FSI problem is nonlinear. For most of this chapter, the solid is assumed to be a linearized elastic one, but SMA materials are also considered. The FSI problem is solved numerically by an iterative procedure which solves sequentially fluid and solid subproblems. Each of the two subproblems is discretized by the Finite Element Method (FEM) and the fluid-structure coupling is reduced to an interface boundary condition. Several numerical examples are presented that illustrate the behavior of the numerical method.

The FSI problem, in general, does not possess closed form solutions even for simple geometries. This makes very difficult validating numerical schemes for its solution. Asymptotic results however are available in the literature. Iliev et al. (2004) recently derived an asymptotic solution to the FSI problem restricted to channel geometries and isotropic linear elastic solids. The derivation is summarized in Appendix A. The objectives of this chapter are twofold. First, a numerical computation is performed which is used to verify the numerical solutions in comparisons with an asymptotic solution to the FSI problem (Section 3.2). This increases the degree of confidence with which such numerical methods can be used in the absence of solid mathematical

understanding of their properties. Secondly, the numerical approach is then extended to FSI problems involving SMA materials instead of linearized elastic solids. The strong temperature dependence of the SMA allows to construct temperature sensitive flow regulating devices with many practical applications. A simulation of a flow in an channel with SMA segment at different temperatures is performed in the end of this chapter (Section 4). The model developed in Chapter III is especially suited for simulating such devices since it captures correctly the phenomena occurring in SMAs over a wide temperature range.

1. The fluid-structure interaction (FSI) Problem

Before the fluid structure problem is presented, we begin with a brief summary of the notation used, the formulation of the fluid, and solid problems alone. Consider a continuum body, defined as an open subset $\Omega \subset \mathbb{R}^3$. The reference configuration representing the body before the deformation has begun is denoted by Ω_0 and the deformed configuration by Ω . Further, we are concerned with stationary processes so the both the Lagrangian fields associated with the solid are time independent as are the Eulerian ones for the fluid. The material coordinates are denoted by \mathbf{p} and the spatial ones by \mathbf{x} .

1.1. Solid

Consider the solid first. It undergoes a continuous, invertible deformation

$$\mathbf{x} = \mathbf{x}(\mathbf{p}),$$

so the reference and deformed configurations are connected by $\Omega = \mathbf{x}(\Omega_0, t)$. The deformation gradient is denoted by \mathbf{F} :

$$\mathbf{F}(\mathbf{p}) = \nabla \mathbf{x}(\mathbf{p}), \quad (5.1)$$

and the displacements $\mathbf{u}(\mathbf{p})$ are given by

$$\mathbf{u}(\mathbf{p}) = \mathbf{x}(\mathbf{p}) - \mathbf{p}. \quad (5.2)$$

The usual infinite small strain tensor is introduced:

$$\mathbf{E}(\mathbf{p}) = \frac{1}{2} (\nabla \mathbf{u}(\mathbf{p}) + \nabla \mathbf{u}(\mathbf{p})^T) \quad (5.3)$$

and the Cauchy stress is denoted by $\mathbf{T}(\mathbf{x})$. Note that, $\mathbf{T}(\mathbf{x})$ is a spatial field representing the stress in the deformed configuration. The field equations for elastic bodies are best formulated in material description, so the first Piola-Kirchhoff stress tensor $\mathbf{S}(\mathbf{p})$ will be used. It is related to $\mathbf{T}(\mathbf{x})$ by:

$$\mathbf{S}(\mathbf{p}) = \det(\mathbf{F}(\mathbf{p})) \mathbf{T}(\mathbf{x}(\mathbf{p})) \mathbf{F}^{-T}(\mathbf{p}). \quad (5.4)$$

For most of this chapter only linearized elastic solids will be considered. Fluid structure problems involving SMA materials will be solved numerically in Section 4 in the end of this chapter. To simplify the presentation, the formulation of the solid problem will be in terms of a linearized elastic solid. The necessary changes to model nonlinear hysteretic materials in a FSI problem will be discussed in Section 4.

Hyperelastic bodies in general are defined as materials for which \mathbf{S} is the gradient of a potential¹, that is, there exists a scalar function $W(\mathbf{F})$, called energy-density, such

¹Equivalently, hyperelastic materials can be defined as materials which produce non-negative work on a closed cycle, (cf., e.g., Gurtin, 1981, pg. 184-191)

that:

$$\mathbf{S}(\mathbf{F}) = \frac{\partial W(\mathbf{F})}{\partial \mathbf{F}}. \quad (5.5)$$

Then, given a body force \mathbf{b}_0 in the reference configuration, the boundary value problem for a solid is stated (in the reference configuration) as follows: *Find $\mathbf{u}(\mathbf{p})$ such that:*

$$\nabla \cdot \mathbf{S} + \mathbf{b}_0 = \mathbf{0} \text{ in } \Omega_0, \quad (5.6)$$

with Dirichlet

$$\mathbf{u} = \hat{\mathbf{u}} \text{ on } \Gamma_0^D \quad (5.7)$$

and/or Neumann

$$\mathbf{S}\mathbf{n}_0 = \hat{\mathbf{s}} \text{ on } \Gamma_0^N \quad (5.8)$$

boundary data, with the usual conditions $\Gamma_0^D \cap \Gamma_0^N = \emptyset$ and $\Gamma_0^D \cup \Gamma_0^N = \Gamma_0$. Under the assumption that $\mathbf{S}(\mathbf{I}) = \mathbf{T}(\mathbf{I}) = \mathbf{0}$ and the assumption that $\nabla \mathbf{u}$ is small one can introduce the fourth-order, linearized elasticity tensor

$$\mathbf{C} = D\mathbf{S}(\mathbf{I}) \quad (5.9)$$

It is a simple calculation to show (see the proposition on pg. 194 in Gurtin (1981)) that:

$$\mathbf{C} = D\mathbf{T}(\mathbf{I}) \quad (5.10)$$

and

$$\mathbf{S} = \mathbf{C} : \mathbf{E} + o(\nabla \mathbf{u}). \quad (5.11)$$

As a result, the balance of linear momentum (5.6) is linearized as

$$-\nabla \cdot (\mathbf{C} : \mathbf{E}) = \mathbf{b}_0. \quad (5.12)$$

The relation (5.11) is known as Hooke's law. Note that, for an isotropic material,

the elasticity tensor \mathbf{C} is necessarily expressed in terms of the two Lamé constants, λ_s and μ_s (cf., eg., Gurtin, 1981; Malvern, 1969), so the stress tensor reduces:

$$\mathbf{S} = \lambda_s \text{tr}(\mathbf{E}) \mathbf{I} + 2\mu_s \mathbf{E}. \quad (5.13)$$

1.2. Newtonian fluid at low Reynolds number

Newtonian fluids are best described using spatial fields. For stationary problems (the spatial description of all involved quantities is time independent), one has a velocity $\mathbf{v}(\mathbf{x})$ and correspondingly, the symmetric part of the velocity gradient, namely, the stretching tensor $\mathbf{D}(\mathbf{x})$ given by:

$$\mathbf{D}(\mathbf{x}) = \frac{1}{2} (\nabla \mathbf{v}(\mathbf{x}) + \nabla \mathbf{v}(\mathbf{x})^T). \quad (5.14)$$

By definition, a Newtonian fluid is one for which (cf., e.g., Gurtin, 1981; Malvern, 1969):

$$\mathbf{T} = -p\mathbf{I} + 2\mu\mathbf{D}, \quad (5.15)$$

where μ is the absolute viscosity of the fluid². The fluid must satisfy the conservation of mass, that is:

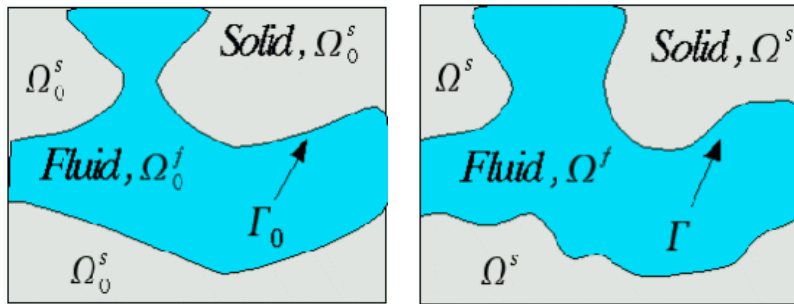
$$\nabla \cdot \mathbf{v} = \mathbf{0}, \quad (5.16)$$

and conservation of linear momentum

$$(\mathbf{v} \cdot \nabla) \mathbf{v} = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \Delta \mathbf{v} + \frac{1}{\rho} \mathbf{b}, \quad (5.17)$$

where \mathbf{b} is a distributed body force (per unit volume) acting on the fluid. With the additional assumption that the term $(\mathbf{v} \cdot \nabla) \mathbf{v}$ is small with respect to the rest of the

²To be precise, a Newtonian fluid is one for which the Cauchy stress \mathbf{T} is a linear function of the velocity gradient \mathbf{D} . There is a beautiful theorem (cf., e.g., Gurtin, 1981, pg. 147-151) which states that invariance under change of observer implies that the fluid is isotropic. In other words, μ is a single scalar.



(a) Reference configuration (b) Deformed configuration

Fig. 54. Schematic of the fluid and solid domains of the FSI problem.

terms in (5.17) and can be disregarded, the balance of linear momentum takes the form

$$-\mu\Delta\mathbf{v} + \nabla p = \mathbf{b}. \quad (5.18)$$

The system (5.16),(5.18) is known as the Stokes equations.

In order to simplify notation, it is convenient to introduce the symmetric part of gradient operator:

$$\mathbf{e}(\mathbf{w}) = \frac{1}{2} (\nabla\mathbf{w} + (\nabla\mathbf{w})^T),$$

for some field \mathbf{w} . Observe, that $\mathbf{E} = \mathbf{e}(\mathbf{u})$ and $\mathbf{D} = \mathbf{e}(\mathbf{v})$.

1.3. Statement of the coupled FSI problem

Consider now the stationary fluid-structure problem (Figure 54) in the deformed configuration $\Omega = \Omega^f \cup \Omega^s$, where the fluid occupies Ω^f , the solid occupies Ω^s and $\Omega^f \cap \Omega^s = \emptyset$. The part of the boundary shared between the fluid and the solid is denoted by $\Gamma^f = \partial\Omega^f \cap \partial\Omega^s$. Further, in order to keep things simple (but not simpler!) it is assumed that the deformation $\mathbf{x}(\mathbf{p})$ is such that contact problems, break-up of the boundary, cavitation of the fluid and other bizzarrities do not occur. There are two

conditions on the interface. First kinematic compatibility between the deformations in the solid and fluid must be satisfied, i.e. the deformation $\mathbf{x}(\mathbf{p})$ must be continuous across the interface. Consider for a moment a time-dependent deformation which, starting from fluid and solid at rest ($\mathbf{x}(\mathbf{p})|_{t=0} = \mathbf{0}$ in Ω_0), leads to some stationary, deformed configuration. It is clear that a sufficient condition for compatibility is that the velocity of the fluid on the interface should be equal to the velocity of the interface itself throughout the deformation. This, for a stationary problem, implies that

$$\mathbf{v} = \mathbf{0} \text{ on } \Gamma^I. \quad (5.19)$$

Further, continuity of tractions on the interface is also required, namely (cf., e.g., Lee and Mei, 1997):

$$\mathbf{T}^f \mathbf{n} = \mathbf{T}^s \mathbf{n} \quad \text{on } \Gamma^I \quad (5.20)$$

where $\mathbf{n} = \mathbf{n}^s$ and \mathbf{n}^s is the outward normal to the solid domain. The stress \mathbf{T}^f in the fluid is given by equation (5.15). Further, the Cauchy stress \mathbf{T}^s can be expressed in terms of the Piola-Kirchhoff stress using equation (5.4), which together with Hooke's law (5.11) imply:

$$-p\mathbf{n} + 2\mu\mathbf{e}(\mathbf{v})\mathbf{n} = \det(\mathbf{F})^{-1}(\mathbf{C} : \mathbf{e}(\mathbf{u}))\mathbf{F}^T \mathbf{n} \quad \text{on } \Gamma^I. \quad (5.21)$$

The FSI problem therefore consists of finding the interface between the two domains, a velocity, pressure and displacements which solve the Stokes (5.16), (5.18), and Navier's equations (5.12) respectively, and also satisfy the interface conditions (5.19) and (5.21). More formally, the FSI boundary-value problem is summarized below in terms of the unknowns Γ^I , \mathbf{v} , p and \mathbf{u} : *Find Γ^I , \mathbf{v} , p and \mathbf{u} such that:*

$$\Gamma^I = \{\mathbf{p} + \mathbf{u}(\mathbf{p}) | \forall \mathbf{p} \in \Gamma_0^I\}, \quad (5.22)$$

$$\begin{aligned} -\mu\Delta\mathbf{v} + \nabla p &= \mathbf{b} && \text{in } \Omega^f, \\ \nabla \cdot \mathbf{v} &= \mathbf{0} && \text{in } \Omega^f, \end{aligned} \tag{5.23}$$

$$-\nabla \cdot (\mathbf{C} : \mathbf{e}(\mathbf{u})) = \mathbf{b}_0 \quad \text{in } \Omega_0^s,$$

$$\det(\nabla\mathbf{u} + \mathbf{I})(-p\mathbf{I} + 2\mu\mathbf{e}(\mathbf{v}))(\nabla\mathbf{u} + \mathbf{I})^{-T} \mathbf{n}_0 = (\mathbf{C} : \mathbf{e}(\mathbf{u})) \mathbf{n}_0 \quad \text{on } \Gamma_0^I, \tag{5.24}$$

\mathbf{v} satisfies the kinematic interface condition (5.19) and in addition \mathbf{v} , p and \mathbf{u} should also satisfy any boundary conditions that might be specified on $\partial\Omega \setminus \Gamma^I$. Equation (5.24) is the continuity of tractions (5.21) expressed on the reference position Γ_0^I by inverting the deformation gradient on the interface. Note that this can always be done because of the kinematic compatibility condition. Observe also that the position of the interface is part of the boundary value problem, and the solid-fluid coupling term (5.24) makes it a nonlinear one.

1.4. Weak form of the elasticity, Stokes and FSI problems

Consider a bounded Lipschitz domain Ω and let $(\cdot, \cdot)_\Omega$ be the usual inner product on $L^2(\Omega)$ and, as there is no chance of confusion, also the inner product on $[L^2(\Omega)]^d$, where $d = 2, 3$ is the size of the spatial dimension. Let $H_0^s(\Omega)$, $-1 \leq s \leq 1$ be the Sobolev spaces and $L_0^2(\Omega)$ be Hilbert space of functions in L^2 having zero mean. For complete development discussion on these subjects, including fractional Sobolev spaces, see Lions and Magenes (1968). Suppose that both Ω_0^s and Ω^f are Lipschitz domains.

To formulate the elasticity problem, one introduces the bilinear form

$$a_{\Omega_0^s}(\mathbf{u}, \mathbf{w}) = \int_{\Omega_0^s} (\mathbf{C} : \mathbf{e}(\mathbf{u})) : \mathbf{e}(\mathbf{w}) d\mathbf{p}.$$

Let $\hat{\mathbf{u}}_0 \in [H^{1/2}(\Gamma_0^D)]^d$ be the Dirichlet data given on $\Gamma_0^D \subset \partial\Omega_0^s$, \hat{s} be the Neumann data given on $\Gamma_0^N \subset \partial\Omega_0^s$, and let $\mathbf{b}_0 \in [H^{-1}(\Omega^f)]^d$ be the distributed body force.

The weak form of the *linear* elasticity problem is: *Find* $\mathbf{u} \in [H^1(\Omega_0^s)]^d$ *such that*:

$$a_{\Omega_0^s}(\mathbf{u}, \mathbf{w}) = (\mathbf{b}_0, \mathbf{w})_{\Omega_0^s} + (\hat{\mathbf{s}}, \mathbf{w})_{\Gamma_0^N}, \quad \forall \mathbf{w} \in [H_D^1(\Omega_0^s)]^d, \quad (5.25)$$

$$\mathbf{u} = \hat{\mathbf{u}}_0, \quad \text{on } \Gamma_0^D. \quad (5.26)$$

The first of the above equations is obtained by multiplying equation (5.12) by a test function and integrating by parts. Note that the Neumann boundary condition appears on the right hand side as the surface integral.

The stability of the weak elasticity problem follows from the classical Korn's inequality: *There exists a positive constant* $C_1 = C_1(\Omega_0^s) > 0$ *independent of* \mathbf{u} , *such that*:

$$\int_{\Omega_0^s} \mathbf{e}(\mathbf{u}) : \mathbf{e}(\mathbf{u}) d\mathbf{p} \geq C_1 \|\mathbf{u}\|_{1, \Omega_0^s}^2 \quad \forall \mathbf{u} \in [H_D^1(\Omega_0^s)]^d. \quad (5.27)$$

The conditions for its validity and a proof of this nontrivial inequality can be found, for example, in Girault and Raviart (1986); Nečas and Hlaváček (1981). It will be assumed here that the elasticity tensor \mathbf{C} and boundary conditions (5.7) and (5.8) are such that (5.27) is satisfied. This is a standard subject which will not be discussed further, the reader is instead referred to Nečas and Hlaváček (1981).

In the case of the Stokes problem assume, again for simplicity, that homogeneous Dirichlet boundary data is given and let $\mathbf{b} \in [H^{-1}(\Omega^f)]^d$. The weak form of the Stokes equation is: *Find* $\mathbf{v} \in [H_0^1(\Omega^f)]^d$, $p \in L_0^2(\Omega^f)$ *such that*:

$$\begin{aligned} D_{\Omega^f}(\mathbf{v}, \mathbf{w}) - (p, \nabla \cdot \mathbf{w})_{\Omega^f} &= (\mathbf{b}, \mathbf{w})_{\Omega^f}, \quad \forall \mathbf{w} \in [H_0^1(\Omega^f)]^d, \\ -(\nabla \cdot \mathbf{v}, q)_{\Omega^f} &= 0, \quad \forall q \in L_0^2. \end{aligned} \quad (5.28)$$

Here $D_{\Omega^f}(\mathbf{v}, \mathbf{w})$ is the vector Dirichlet form

$$D_{\Omega^f}(\mathbf{v}, \mathbf{w}) = \int_{\Omega^f} \mu \nabla \mathbf{v} : \nabla \mathbf{w} d\mathbf{x}.$$

The existence and uniqueness of solution to the Stokes problem follows from the clas-

sical inf-sup condition³ (cf., e.g., Girault and Raviart, 1986): *There exists a positive constant $C_2 = C_2(\Omega^f) > 0$, independent of \mathbf{v} and p , and such that:*

$$\inf_{\forall p \in L_0^2(\Omega^f)} \sup_{\forall \mathbf{v} \in [H_0^1(\Omega^f)]^d} \frac{(p, \nabla \cdot \mathbf{v})_{\Omega^f}^2}{D_{\Omega^f}(\mathbf{v}, \mathbf{v}) \|p\|^2} > C_2. \quad (5.29)$$

For a complete discussion of the weak problem (5.28) and condition (5.29) the reader is referred to the book of Girault and Raviart (1986).

To write a weak form of the FSI problem, observe that the interface condition (5.24) can be treated as a nonlinear Neumann boundary condition for the solid problem only. Accordingly, we introduce the nonlinear form:

$$g_{\Gamma_0^I}(\mathbf{v}, \mathbf{u}, p, \mathbf{w}) = \int_{\Gamma_0^I} \left\{ \det(\nabla \mathbf{u} + \mathbf{I})(-p\mathbf{I} + 2\mu \mathbf{e}(\mathbf{v})) (\nabla \mathbf{u} + \mathbf{I})^{-T} \mathbf{n}_0 \right\} \cdot \mathbf{w} ds. \quad (5.30)$$

After integrating by parts the balance of linear momentum for the solid and fluid, it is trivial to check that the boundary value problem (5.23)-(5.24) can be restated in the following weak form: *Find the interface Γ^I , the deformed configuration of the fluid domain Ω^f , the displacements $\mathbf{u} \in [H^1(\Omega_0^s)]^d$, velocity $\mathbf{v} \in [H_0^1(\Omega^f)]^d$ and pressure $p \in L_0^2(\Omega^f)$ such that:*

$$\begin{aligned} D_{\Omega^f}(\mathbf{v}, \mathbf{w}) - (p, \nabla \cdot \mathbf{w})_{\Omega^f} &= (\mathbf{b}, \mathbf{w})_{\Omega^f}, & \forall \mathbf{w} \in [H_0^1(\Omega^f)]^d, \\ -(\nabla \cdot \mathbf{v}, q)_{\Omega^f} &= 0, & \forall q \in L_0^2, \\ a_{\Omega_0^s}(\mathbf{u}, \mathbf{w}) &= (\mathbf{b}_0, \mathbf{w})_{\Omega_0^s} + g_{\Gamma_0^I}(\mathbf{v}, \mathbf{u}, p, \mathbf{w}), & \forall \mathbf{w} \in [H_D^1(\Omega_0^s)]^d, \\ \Gamma &= \{\mathbf{p} + \mathbf{u}(\mathbf{p}) \mid \forall \mathbf{p} \in \Gamma_0\}. \end{aligned} \quad (5.31)$$

In addition \mathbf{v} , p and \mathbf{u} should also satisfy the appropriate boundary conditions specified on $\partial\Omega \setminus \Gamma^I$.

³The inf-sup condition is also referred to by the name Ladyzenskaya-Babuska-Brezzi (LBB) condition.

2. Solution methods for the coupled FSI system

Upon introducing the finite-dimensional subspaces $U_{\mathbf{u}}$, $U_{\mathbf{v}}$ and U_p for the displacements, velocity and pressure respectively, the first three equations in (5.31) lead to the following nonlinear system of algebraic equations for \mathbf{v} , p and \mathbf{u} :

$$\begin{pmatrix} \mathbf{A}(\mathbf{u}) & \mathbf{C}^T(\mathbf{u}) & \mathbf{0} \\ \mathbf{C}(\mathbf{u}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ p \\ \mathbf{u} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_f(\mathbf{u}) \\ \mathbf{d} \\ \mathbf{b}_s + \mathbf{g}(\mathbf{u}, \mathbf{v}, p) \end{pmatrix}, \quad (5.32)$$

where the blocks $\mathbf{A}(\mathbf{u})$ and \mathbf{K} correspond to the bilinear forms $D_{\Omega^f}(\cdot, \cdot)$ and $a_{\Omega_0^s}(\cdot, \cdot)$, while the blocks $\mathbf{C}(\mathbf{u})$ and $\mathbf{C}(\mathbf{u})^T$ couple the velocity and pressure unknowns. Since the position of Γ^I and hence Ω^f depends on \mathbf{u} , both \mathbf{A} and \mathbf{C} are functions of the displacement. The vector-columns $\mathbf{b}_f(\mathbf{u})$ and \mathbf{b}_s correspond to the body force in the fluid and solid respectively, modified by application of essential (Dirichlet, periodic, etc.) boundary conditions. Note that the vector \mathbf{d} appears in the right-hand side of (5.32) when the essential boundary conditions are applied by matrix transformations. If for example they are applied by a penalty method, then $\mathbf{d} \equiv \mathbf{0}$.

It is important to observe that the coupling between the fluid and the structure (5.21) appears on the right hand side of (5.32) as the nonlinear vector-function $\mathbf{g}(\mathbf{u}, \mathbf{v}, p)$ which corresponds to the form $g_{\Gamma_0^I}(\cdot, \cdot, \cdot, \cdot)$. Note that the fluid tractions on acting on the solid are evaluated in the reference configuration, i.e. on Γ_0^I .

2.1. Dirichlet-Neumann iterative scheme

One way to solve the system (5.31) is to use an iterative scheme which successively solves separate problem on the two domains. Considering the following approach:

- Solve the Stokes equation in the fluid domain treating the solid as a rigid body;

- Transfer the forces to the solid;
- Calculate the displacement field in the solid and then update the fluid domain.

The second step needs some further clarification. In order to solve an elasticity problem the boundary conditions should be specified on the reference configuration, while the tractions computed from a fluid solution are given in the deformed configuration. For given velocities and pressure, the interface condition (5.21) specifies the traction on the solid domain in the deformed configuration, so we have to use the definition (5.4) of the Piola-Kirchhoff stress tensor and convert the tractions to the reference configuration. That is exactly what happens when $g(\dots)$ is evaluated. It is not difficult to check that the above algorithm corresponds to a fixed point iteration using the following linearization of (5.32): *Set $\mathbf{u}_0 = \mathbf{0}$, $\mathbf{v}_0 = \mathbf{0}$, $\mathbf{p}_0 = 0$; given $(\mathbf{u}_k, \mathbf{v}_k, p_k)^T$, find $(\mathbf{u}_{k+1}, \mathbf{v}_{k+1}, p_{k+1})^T$ such that:*

$$\begin{pmatrix} \mathbf{A}(\mathbf{u}_k) & \mathbf{C}^T(\mathbf{u}_k) & \mathbf{0} \\ \mathbf{C}(\mathbf{u}_k) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K} \end{pmatrix} \begin{pmatrix} \mathbf{v}_{k+1} \\ \mathbf{p}_{k+1} \\ \mathbf{u}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_f(\mathbf{u}_k) \\ \mathbf{d} \\ \mathbf{b}_s + \mathbf{g}(\mathbf{u}_k, \mathbf{v}_{k+1}, \mathbf{p}_{k+1}) \end{pmatrix}. \quad (5.33)$$

Since the matrix on the left-hand side of the above equation is block-diagonal the block corresponding to the fluid is solved first:

$$\begin{pmatrix} \mathbf{v}_{k+1} \\ \mathbf{p}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{A}(\mathbf{u}_k) & \mathbf{C}^T(\mathbf{u}_k) \\ \mathbf{C}(\mathbf{u}_k) & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{b}_f(\mathbf{u}_k) \\ \mathbf{d} \end{pmatrix}. \quad (5.34)$$

Once \mathbf{v}_{k+1} and \mathbf{p}_{k+1} are available, the block corresponding to the solid, i.e.,

$$\mathbf{u}_{k+1} = \mathbf{K}^{-1} (\mathbf{b}_s + \mathbf{g}(\mathbf{u}_k, \mathbf{v}_{k+1}, \mathbf{p}_{k+1})) \quad (5.35)$$

is solved. This iterative scheme can be expressed more explicitly in the following

Algorithm 2.1.1. (*Dirichlet-Neumann domain decomposition method for the FSI problem*) Set $\mathbf{u}_0 = \mathbf{0}$. For $k = 0, 1, \dots$ until convergence do:

1. Find \mathbf{v}_k, p_k which satisfy the Stokes equations (5.16),(5.18) in Ω_k^f with the no-slip boundary condition on the interface Γ_k^I and the appropriate boundary conditions on $\partial\Omega_k^f \setminus \Gamma_k^I$.
2. Compute the traction $\mathbf{t}_k = \mathbf{T}\mathbf{n}_k$ on the interface Γ_k^I using equation (5.21).
3. Based on \mathbf{t}_k compute the tractions \mathbf{s}_k in the reference configuration of the interface, i.e. Γ_0^I using equation (5.4) and the current iterate for the displacements \mathbf{u}_k .
4. Find \mathbf{u}_{k+1} which satisfies the balance of linear momentum (5.12) in Ω_k^s with $\mathbf{S}\mathbf{n}_0 = \mathbf{s}_k$ and the appropriate boundary data on $\partial\Omega_0^s \setminus \Gamma_0^I$.
5. Compute $\Gamma_{k+1}^I = \{\mathbf{p} + \mathbf{u}_{k+1}(\mathbf{p}) | \forall p \in \Gamma_0^I\}$ and Ω_{k+1}^f :
6. Check convergence: $\|\mathbf{u}_{k+1} - \mathbf{u}_k\|_{\Gamma_{k+1}^I} < TOLERANCE * \|\mathbf{u}_{k+1}\|_{\Gamma_{k+1}^I}$. The norm is the discrete euclidian norm of the interface nodal values.

It is clear that if the interface converges to a fixed position then the velocity and pressure field will satisfy the Stokes equation (5.16),(5.18), the displacement field will satisfies the elasticity equations (5.12), and as a results of the converged interface, the interface condition (5.21) will also be satisfied.

2.2. FEM approximation of the FSI problem

In this section the FEM approximation for the FSI problem will be introduced. Both the solid and fluid problems are discretized using the FEM method. The elasticity problem is solved by standard linear triangular elements. That is, given a triangula-

tion \mathcal{T}_h^s of Ω_0^s , the approximation space for the displacements is chosen to be:

$$U_{\mathbf{u}} = \left[\left\{ u \in C^0(\Omega_0^s) \mid u \text{ is linear on } \forall \tau \in \mathcal{T}_h^s \right\} \right]^d \subset [H^1(\Omega_0^s)]^d. \quad (5.36)$$

The Stokes problem is solved using the P_2P_1 (Taylor-Hood) element pair. Given a triangulation \mathcal{T}_h^f of Ω^f the approximation spaces for the velocity and pressure are defined by

$$U_{\mathbf{v}} = \left[\left\{ v \in C^0(\Omega^f) \mid v \text{ is quadratic polynomial on } \forall \tau \in \mathcal{T}_h^f \right\} \right]^d \subset [H^1(\Omega^f)]^d \quad (5.37)$$

$$U_p = \{ p \in C^0(\Omega^f) \mid p \text{ is linear on } \forall \tau \in \mathcal{T}_h^f \} \subset H^1(\Omega^f) \subset L^2(\Omega^f) \quad (5.38)$$

respectively.

It follows from the inclusion $U_{\mathbf{u}} \subset [H^1(\Omega_0^s)]^d$ that Korn's inequality is satisfied (5.27) on the subspace $U_{\mathbf{u}}$, therefore the discretized elasticity problem has a unique solution. The selection of the approximation spaces for the Stokes problems is governed by the fact that they must satisfy the discrete version of the inf-sup, that is,

$$\inf_{\forall p \in U_p} \sup_{\forall \mathbf{v} \in U_{\mathbf{v}}} \frac{(p, \nabla \cdot \mathbf{v})_{\Omega^f}^2}{D_{\Omega^f}(\mathbf{v}, \mathbf{v}) \|p\|_{\Omega^f}^2} > C_2 \quad (5.39)$$

in order to have a stable approximation. Such elements are said to be LBB stable. While many other elements are known to be LBB stable (Gresho and Sani, 1998), the Taylor-Hood element was chosen because it provides a balanced approximation for both velocity and pressure (Axelsson and Neytcheva, 2003).

The two discretized domains have piecewise straight boundaries so the use of linear approximation for the displacement field simplifies things because Γ_k^I will remain piecewise straight and the two meshes will be point-wise conforming at all times. The interface condition (5.19) however is enforced only weakly, because the stresses in the solid are piecewise constants while the stresses in the fluid are piecewise lin-

ear functions⁴. Note that both are discontinuous across elements. Investigation of curvilinear interface boundaries and/or higher order approximation spaces for the elasticity problem was outside the scope of this work.

2.3. Solution methods for the solid and fluid subproblems

In this section the methods used to solve the linear systems of algebraic equations (5.33) arising in the discretization (5.31), (5.36) - (5.38) of the previous section are discussed. Some other practical issues of the implementation of Algorithm 2.1.1 such as mesh regeneration are also described.

In general, the block matrices appearing in (5.33) are large sparse matrices so they can be stored efficiently in machine memory, however inverting them explicitly is both computationally expensive and the resulting matrices are dense. Therefore, at each iterations of Algorithm 2.1.1 instead of computing and applying directly the inverse matrices in (5.34) and (5.35) one solves the equivalent linear systems by an iterative method. Since these two systems have to be solved once per each iteration of Algorithm 2.1.1, it is important that this is done so efficiently. The methodology used to solve these two linear systems is described next.

2.3.1. Linearized elasticity problem

The linear system corresponding to equation (5.35) is

$$\mathbf{K}\mathbf{u}_{k+1} = \mathbf{b}_s + \mathbf{g}(\mathbf{u}_k, \mathbf{v}_{k+1}, \mathbf{p}_{k+1}). \quad (5.40)$$

⁴Since the contributions to the stresses in the fluid come both from the pressure and the velocity gradients things will not change if one uses other approximations for the fluid, e.g. P_2P_0 .

The stiffness matrix \mathbf{K} is symmetric, positive definite and sparse, making it ideal for the application of the Conjugate Gradient (CG) method.

The CG method was first proposed in the classical works of Hestenes and Stiefel (1952); Lanczos (1952) and is the method of choice when solving large, sparse, symmetric and positive definite linear systems. This standard method will not be described in detail and the reader is referred for example to Golub and Van Loan (1996); Saad (1996). It is sufficient to note that at each iteration of the conjugate gradient, one has to apply the action of the matrix \mathbf{K} on a vector \mathbf{w} , that is the operation $\mathbf{K}\mathbf{w}$ should be computationally inexpensive. Clearly this is the case with the stiffness matrix \mathbf{K} .

The number of iterations required by the CG method to reduce the discrete L^2 norm of any initial guess by certain fixed amount is proportional to

$$\ln(\kappa(\mathbf{K}))\sqrt{\kappa(\mathbf{K})},$$

where $\kappa(\mathbf{K})$ is the condition number of \mathbf{K} (cf., e.g., Golub and Van Loan, 1996; Saad, 1996). The larger the condition number, the more iterations it takes the CG to converge. Hence, in order to improve the convergence rate of the CG, it is necessary to precondition equation (5.40), i.e. instead of solving the original system, one solves

$$\tilde{\mathbf{K}}^{-1}\mathbf{K} = \tilde{\mathbf{K}}^{-1}(\mathbf{b}_s + \mathbf{g}(\mathbf{u}_k, \mathbf{v}_{k+1}, \mathbf{p}_{k+1})),$$

where $\tilde{\mathbf{K}}$ is an approximation to \mathbf{K} in a sense that $\tilde{\mathbf{K}}^{-1}\mathbf{K}$ is symmetric, positive definite, it is better conditioned than \mathbf{K} , i.e. $\kappa(\tilde{\mathbf{K}}^{-1}\mathbf{K}) < \kappa(\mathbf{K})$, and finally, the action of $\tilde{\mathbf{K}}^{-1}$ is easily computable. If these three requirements are met, it is more efficient to apply the CG to the latter system, instead of the original one. Finding suitable preconditioner $\tilde{\mathbf{K}}$ is not trivial.

The condition number of the stiffness matrix \mathbf{K} depends on the mesh parameter

h as (cf., e.g., Axelsson and Barker, 1984; Bramble and Zhang, 2000):

$$\kappa(\mathbf{K}) = \mathcal{O}(h^{-2}),$$

so the rate of convergence of the CG is proportional to $\sqrt{\kappa(\mathbf{K})} = \mathcal{O}(h^{-1})$. Taking into account that the cost per iteration of the conjugate gradient is $\mathcal{O}(h^{-d})$, the total complexity of applying the unpreconditioned CG to (5.40) is $\mathcal{O}(h^{-(1+d)}) = \mathcal{O}(N_u^{1+\frac{1}{d}})$, where N_u is the dimension of the displacement space $U_{\mathbf{u}}$. Preconditioning of (5.40) is therefore necessary.

The preconditioner used in this work is based upon a block diagonal factorization of the stiffness matrix. Denote by $\phi_i^{(j)}$, $i = 1, \dots, d$, $j = 1, \dots, N_u$, the nodal basis functions for the displacement space $U_{\mathbf{u}}$. If the nodal unknowns are ordered by displacement component, i.e. the vector column $\mathbf{u} = (u_1^{(1)}, u_1^{(2)}, \dots, u_1^{(N_u)}, u_2^{(1)}, \dots)^T$ contains all the nodal unknowns of the first displacement component, followed by all the unknowns for the second one, and so on, then the stiffness matrix \mathbf{K} has the following block structure:

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_{11} & \cdots & \mathbf{K}_{1d} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{d1} & \cdots & \mathbf{K}_{dd} \end{pmatrix}. \quad (5.41)$$

This ordering allows for a robust preconditioning of the linear system. Consider the matrix $\mathbf{K}_{SDC} = \text{diag}(\mathbf{K}_{11}, \dots, \mathbf{K}_{dd})$ which contains only the diagonal blocks of \mathbf{K} . It can be shown that \mathbf{K}_{SDC} is an optimal preconditioner for \mathbf{K} (Blaheta, 1994), that is, the condition number of the preconditioned matrix satisfies

$$\kappa((\mathbf{K}_{SDC})^{-1}\mathbf{K}) \leq \frac{d-1}{C_1} \frac{1-\nu}{1-2\nu}. \quad (5.42)$$

where C_1 is the mesh independent constant, appearing in (5.27) and ν is Poisson

ratio. For proof and improvements in the above estimate, see Blaheta (1994); Nečas and Hlaváček (1981). In general the application of $(\mathbf{K}_{SDC})^{-1}$ at each CG iteration for (5.40) can be done by multigrid in linear time, leading to an optimal method. A different (and simpler) approach, selected here, is to use a $MIC(0)$ factorization of \mathbf{K}_{SDC} which results in condition number of order $\mathcal{O}(h^{-1})$. As a result, the CG finds a solution in $\mathcal{O}(h^{-1/2})$ iterations and the overall algorithm complexity is $\mathcal{O}\left(N_u^{1+\frac{1}{2d}}\right)$. For detailed description of the $MIC(0)$ preconditioner see Blaheta (1994). For the problems under consideration this preconditioner proved to be adequate enough.

2.3.2. Stokes problem

Solving the linear system of algebraic equations for the Stokes problem is an active area of research and there is no general agreement on what is the most efficient way (Axelsson and Neytcheva, 2003; Bramble and Pasciak, 1997; Turek, 1999). In this work, the fluid problem (5.34) is solved again by the CG method, but applied to the Schur complement for the pressure variables. A comprehensive study of pressure Schur complement methods can be found in (Turek, 1999). For notational simplicity, in this section the implicit dependence on \mathbf{u}_k of the matrices \mathbf{A} and \mathbf{C} will be suppressed.

The Stokes system corresponding to equation (5.34)

$$\begin{pmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_f \\ \mathbf{d} \end{pmatrix} \quad (5.43)$$

is indefinite, which makes impossible direct application of the CG. While generalized Krylov subspace methods such as MINRES (Golub and Van Loan, 1996; Saad, 1996) can still be used, preconditioning is not an obvious task. Therefore the Schur complement method for the pressure variable is used (Axelsson and Neytcheva, 2003; Bramble and Pasciak, 1997). Observe that \mathbf{A} is an invertible matrix, so one can

eliminate the first row of (5.43) to obtain

$$\mathbf{S}\mathbf{p} = \mathbf{C}\mathbf{A}^{-1}\mathbf{b}_f - \mathbf{d}. \quad (5.44)$$

where \mathbf{S} is the Schur complement matrix⁵:

$$\mathbf{S} = \mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T.$$

The invertibility of \mathbf{S} , which is also a symmetric matrix, follows from the inf-sup condition. To see this, take an arbitrary element $\mathbf{q} \in U_p$ of the discrete pressure space and observe that:

$$(\mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T\mathbf{q}, \mathbf{q}) = \sup_{\mathbf{w} \in U_v} \frac{(\mathbf{q}, \mathbf{C}\mathbf{w})^2}{(\mathbf{A}\mathbf{w}, \mathbf{w})}$$

It follows from (5.39) that the right-hand side of the last equation satisfies

$$\sup_{\mathbf{w} \in U_v} \frac{(\mathbf{q}, \mathbf{C}\mathbf{w})^2}{(\mathbf{A}\mathbf{w}, \mathbf{w})} \geq C\|\mathbf{w}\|^2, \quad \forall \mathbf{q} \in U_p,$$

therefore \mathbf{S} is a positive definite matrix, and hence invertible. Secondly, since it is symmetric, the CG method can be applied to the reduced system (5.44). Now, in general, \mathbf{S} is a dense matrix and it is expensive to evaluate it explicitly. However, the CG algorithm only requires the computation of the action of \mathbf{S} on a vector, and this can be done, provided that one can solve linear systems with \mathbf{A} efficiently.

To see how the last can be done, denote again by $\phi_i^{(j)}$, $i = 1, \dots, d$, $j = 1, \dots, N_v$, the nodal basis functions of the velocity space U_v and by $\psi^{(j)}$, $j = 1 \dots N_p$ the nodal basis of pressure space U_p . Upon ordering the velocity unknowns by component, the

⁵The notation for the Schur complement is unique to this section alone, therefore there is no chance of confusion with the Piola-Kirchhoff stress.

block \mathbf{A} has a block-diagonal structure:

$$\mathbf{A} = \text{diag}(\mathbf{A}_1, \dots, \mathbf{A}_d),$$

where the components of each block \mathbf{A}_l are given by (no implicit summation by repeated indices assumed!):

$$(A_l)_{ij} = \mu \int_{\Omega^f} \sum_{k=1}^d \phi_{l,k}^{(i)} \phi_{l,k}^{(j)} d\mathbf{x}, \quad l = 1, \dots, d. \quad (5.45)$$

Similarly, $\mathbf{C} = (\mathbf{C}_1, \dots, \mathbf{C}_d)$, where (no implicit summation):

$$(C_l)_{ij} = - \int_{\Omega^f} \phi_{l,i}^{(i)} \psi^{(j)} d\mathbf{x}, \quad l = 1, \dots, d. \quad (5.46)$$

Since \mathbf{A} is block-diagonal, inverting it reduces to inverting each block. Observe, however, that each block \mathbf{A}_l corresponds to a Laplacian stiffness matrix, and these can be inverted efficiently (in $\mathcal{O}(1)$ CG iterations with a multigrid preconditioner, for example). That is, when the action of $(\mathbf{A}_l)^{-1}$ is required, one solves the corresponding linear system for \mathbf{A}_l using the CG with an appropriate preconditioner. In the current work, a threshold ILU factorization (Saad, 1996) proved to be sufficient for the problems under consideration.

Returning back to the reduced linear system for the Schur complement (5.44), the condition number of \mathbf{S} is independent of the mesh parameter h , although it depends on the geometry of the domain Ω^f (Turek, 1999). It is nevertheless desirable to precondition it. As a preconditioner for \mathbf{S} one can use a mass matrix \mathbf{M}^p on the pressure space:

$$M^p_{ij} = \int_{\Omega^f} \psi^{(i)} \psi^{(j)} d\mathbf{x}, \quad (5.47)$$

which, as shown in Turek (1999), reduces the total number of iterations several times. In the current work, \mathbf{M}^p is the selected preconditioner for \mathbf{S} and the blocks \mathbf{A}_l are

preconditioned with a threshold ILU factorization (Saad, 1996). As a final remark, \mathbf{M}^p is a mass matrix, therefore the application of $(\mathbf{M}^p)^{-1}$ (by CG iteration) requires $\mathcal{O}(1)$ CG iterations.

To summarize, the fluid problem is solved by applying the CG to the reduced system (5.44). At each CG iteration, the application of \mathbf{A}^{-1} in the matrix-vector multiplication with \mathbf{S} is performed by an internal CG iteration with an ILU preconditioner for \mathbf{A} . Also, the application of the preconditioner $(\mathbf{M}^p)^{-1}$ for the Schur complement \mathbf{S} is performed again by an internal CG iteration.

2.3.3. Fluid mesh regeneration

At the beginning of Algorithm 2.1.1 one has a conforming triangulation of both the solid and fluid domains. At the end of each iteration (step 5) the fluid domain is updated and as a result, the mesh has to also change. On the other hand, the elasticity equation is always solved in the original configuration, so the elastic mesh remains unchanged. Since only conforming meshes are considered, the modification of the fluid mesh must be such that conformity is maintained on Γ^I . That is, when the interface is deformed using the computed displacement \mathbf{u}_{k+1} , solid vertices will coincide with fluid vertices and solid segments (faces in 3D) should coincide with fluid ones.

The easiest approach is to move the interface vertices at the end of step 5 of Algorithm 2.1.1 which will affect only the elements which contain them. This will work as long as the interface displacements are small compared to the local mesh size. If the mesh size near the interface is comparable to the displacements of the interface the mesh can lose quality or completely degenerate if a vertex is moved into another element (Figures 55 and 56). As a result, given the domain and boundary condition, the mesh size in the fluid domain cannot be too small. Such a restriction is clearly

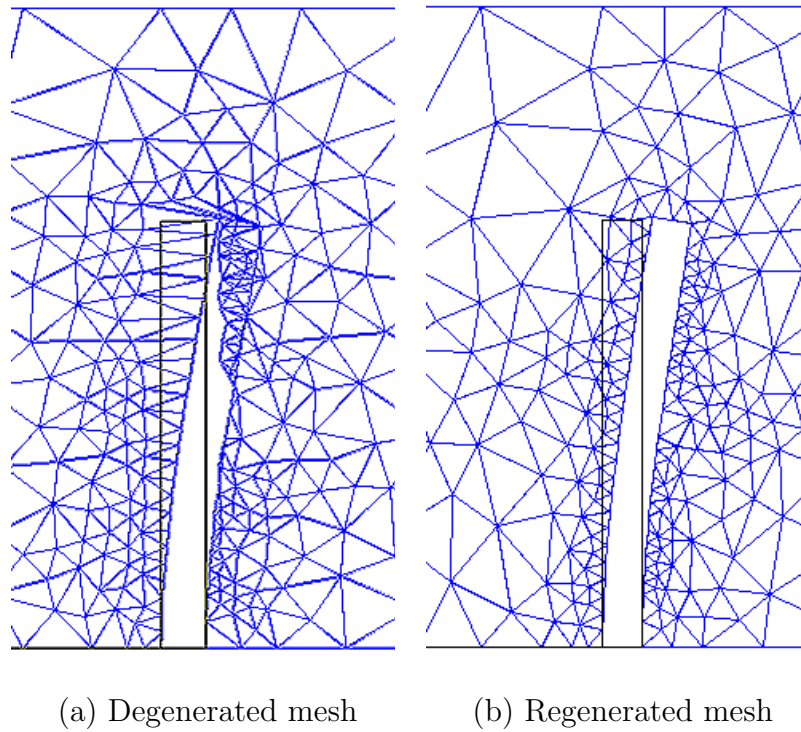
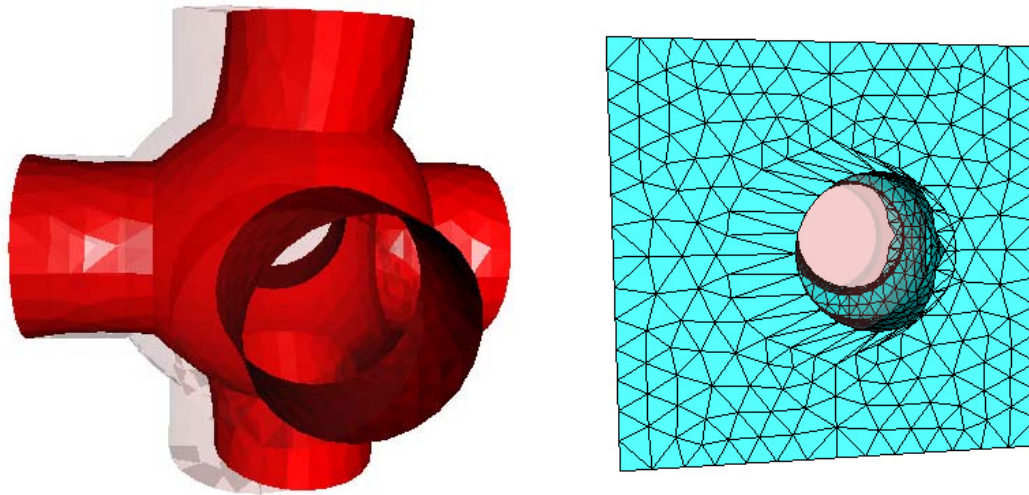


Fig. 55. A 2D degenerate mesh. In this example an elastic obstacle deforms to the left in response to flow in the channel. The solid lines indicate its initial configuration. If only the boundary nodes of the fluid mesh are moved, it degenerates (left). The second mesh (right) is obtained after remeshing the fluid domain.



(a) Flow external to a elastic skeleton (b) Degenerated mesh after first iteration

Fig. 56. A 3D degenerate mesh. Another case when the mesh degenerates after a new position of the interface is computed. Shown are the initial (transparent shade) and final position (solid red) of the interface Γ^I (a) and the degenerated mesh after the first iteration (b). The domain is the unit cube and the solid geometry is formed by the intersection of three perpendicular cylinders and a central circle. The flow is from left to right, is exterior to the solid (the caps on the cylinders are not shown) and the boundary conditions allow the caps of the two cylinders perpendicular to the flow direction to slide on the side of the unit cube. Unlike the 2D case, the regeneration of the mesh involves both the interior and part of the surface mesh.

unacceptable. It can be overcome by either globally modifying the existing mesh, for example by solving an artificial elasticity problem in the fluid domain or by remeshing it (locally or globally).

The global remeshing approach is selected here because of the ready availability of mesh generators which could do that. At the conclusion of step 5, the elements with vertices on the interface are modified and if the resulting mesh has poor quality and/or it degenerates then the entire fluid domain is remeshed, retaining the same boundary segments in 2D and the same interface faces in 3D. Note that in 3D, in addition to the entire volume mesh, one also has to modify part of the surface mesh as can be seen from figure Figure 56, while keeping the interface conforming. In principle local regeneration of the mesh (after removing low quality/degenerate elements) is an interesting possibility which can significantly speed up the process, however developing the software necessary to utilize this strategy was outside of the scope of this research.

3. Model problems with a linearized elastic solid

Three model problems were considered in order to test the Dirichlet-Neumann iterative scheme of Section 2.1. The first two model problems involve flow in the elastic channel geometry of the previous section, while the third one is for a channel with elastic segment. The first model problem is used to demonstrate the convergence properties of the iterative scheme with regard to various problem parameters. The second problem is set up in the same type of geometry, but with slightly different boundary conditions, so that a comparison can be made between the asymptotic solution of the previous sections and numerical solutions. The last problem of flow in a channel with an elastic segment demonstrates the highly nonlinear dependence of

permeability on the pressure gradient across the channel.

The *Triangle*, version 1.4, mesh generating software was used for meshing 2-D domains (Shewchuk, 2002) and the *Netgen*⁶, version 4.0, was used for meshing 3-D domains. The FEM software used for the solid and fluid subproblems, the FSI iterative Algorithm 2.1.1, and all other related components (excluding mesh generation) such as numerical linear algebra subroutines and graphical post-processing tools were developed by the author. They were implemented in the *Object Pascal* programming language and compiled with *Borland*[®] *Delphi*[™], version 2.01.

3.1. Flow through an elastic channel

The first model problem to be considered is one for a flow through an elastic channel. The geometry is shown on Figure 57. The channel has length 5 and thickness 1. The thickness of each of the elastic slabs is 0.2. The material parameters used for the solid are $E = 1.44$ and $\nu = 0.2$ while the fluid has viscosity $\mu = 0.1$ and density $\rho = 1$. The velocity distribution at the inlet is that of a developed Poiseuille flow: $v_x = 4 * V_{max} y(1 - y)$, where the constant V_{max} is the maximal inflow velocity (achieved at $y = 0.5$). First the numerical algorithm is tested for $V_{max} = 1/4$ and a (triangular) mesh with $h \sim 16 * 10^{-2}$ in the fluid domain and $h \sim 4 * 10^{-2}$ in the solid domain. No triangle had internal angles less than 30 degrees. The mesh is chosen coarser in the fluid domain because of the 2-nd order accuracy for the velocity (and 1-st order for the pressure/stress) compared to the 1-st order approximation in the solid region. The computed pressure and velocity profiles are displayed on Figures 58(a) and 58(b).

Next, the problem is solved for several values of V_{max} and three different mesh

⁶*Netgen* is developed by Joachim Schöberl and can be downloaded (as of March 2005) from <http://www.hpfem.jku.at/netgen/>.

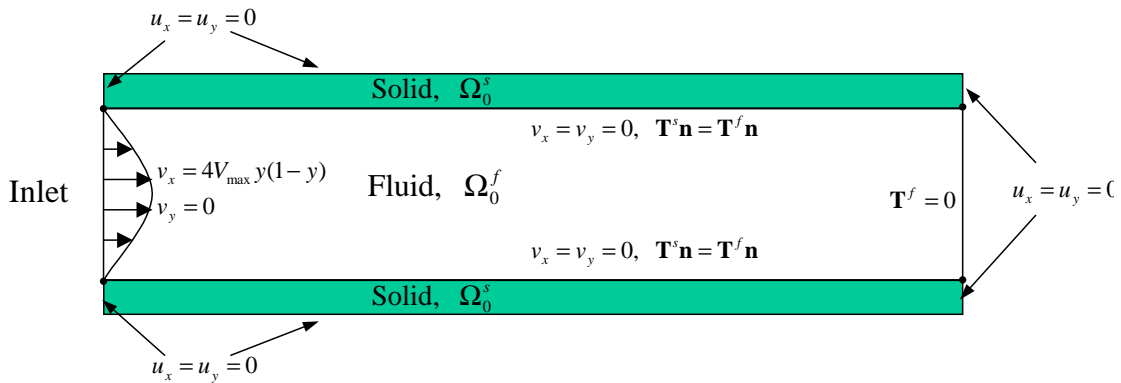
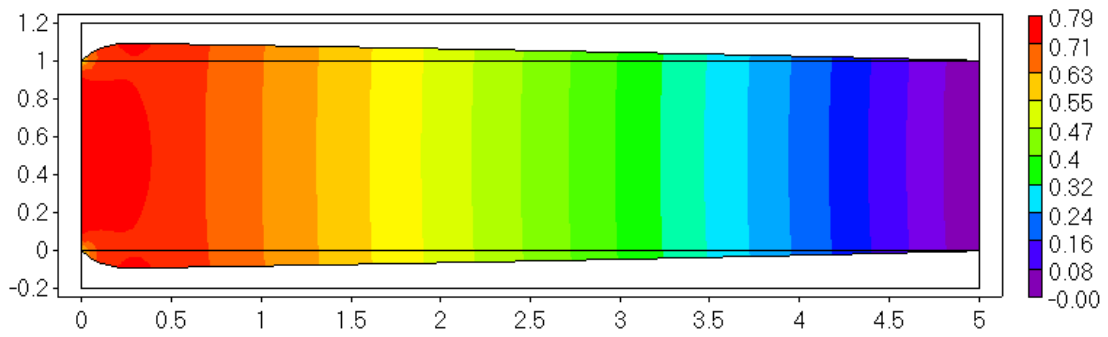


Fig. 57. Geometry and boundary conditions for flow in channel with elastic walls.

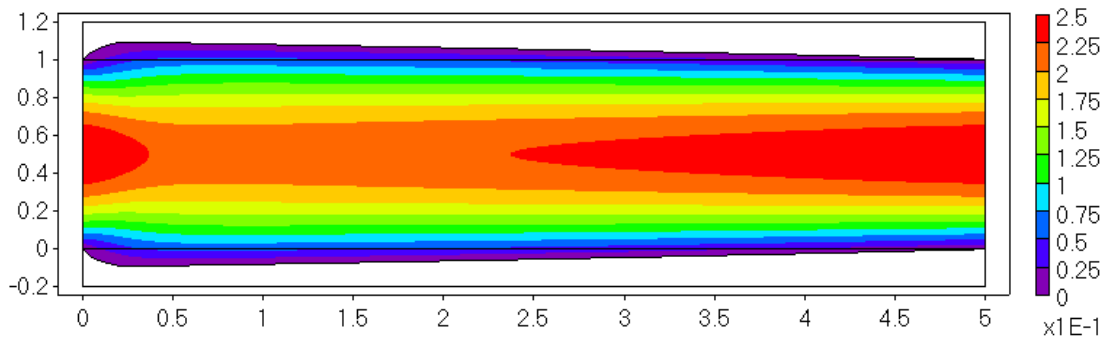
Table V. Convergence of the Dirichlet-Neumann iterative scheme for the elastic channel FSI problem.

V_{max}	h		$h/2$		$h/4$	
	Iterations	max u_y	Iterations	max u_y	Iterations	max u_y
1/4	9	0.0894687	9	0.0894628	9	0.0894809
1/8	7	0.0504095	7	0.0505136	7	0.0505101
1/16	6	0.0271444	6	0.0271742	6	0.0271846
1/32	5	0.0141473	5	0.0141589	5	0.0141654
1/64	5	0.0072323	5	0.0072312	5	0.0072402

sizes. This is done in order to get a preliminary idea of convergence rate sensitivity of the iterative scheme of Algorithm 2.1.1 to the inflow velocity and mesh size. The number of iterations it took to reach relative precision 10^{-6} is reported in Table V. Also given in the table is the maximum vertical displacement in the solid. Even though a proper non-dimensional analysis is not done it is felt that for this problem varying input velocity V_{max} is the same as varying the elastic stiffness E or fluid viscosity μ . It can be concluded from this table that the iteration numbers depend on the inlet



(a) Pressure profile



(b) Horizontal velocity profile

Fig. 58. Solution of the FSI problem for $V_{max} = 1/4$. Shown are the velocity and pressure in the deformed configuration of the fluid domain Ω^f .

velocity which in turn directly affects the magnitude of the interface displacement. In general, numerical experiments suggest that the important parameter is the magnitude of the interface displacements. On the other hand the iteration convergence rate seem not to depend on the mesh parameters. Finally, since the maximum displacement of the interface for each case of inlet velocity stabilizes as h decreases one can infer that the algorithm is convergent ⁷.

3.2. Permeability of a long elastic channel

While the figures presented in the previous numerical example look reasonable from a physical stand point, in the absence of exact analytical solutions it is difficult to verify the quality of the numerical solution. It may happen that the continuous problem (5.31) is not well posed. It may also happen that the selected numerical approximation (5.36)-(5.38) is not a stable one, that is, the stationary FSI problem (5.31) does have a unique solution but the FEM subspaces (5.36)-(5.38) do not lead to a convergent method.

The asymptotic solution developed in Appendix A for a long elastic channel can however be compared with a numerical one. The asymptotic formulae (A.44) and (A.47) are derived based on several assumptions (see Appendix A) and without analysis of the rate of convergence with respect to the small parameter ε . However if both the asymptotic solution and the numerical one converge to the actual one, then we should also observe convergence between the two of them. In this section we use the numerical method described in Section 2.1 in order to validate the asymptotic formulae (A.44) and (A.47) and vice-versa.

To do this, the elastic channel of Figure 67 (on page 230) is discretized for

⁷Of course this can only be verified with a known analytical solution.

$l = \delta = 0.5$. In this way, a y-periodic arrangement of this geometry will have solid and fluid regions of equal unit width. The boundary conditions are also modified, compared to the previous model problem. Instead of fixing both displacements at $x = 0$ and $x = L$ we only constrain u_1 , i.e. the end of the channel is now free to move in the vertical direction. Note that this does not represent a y-periodic boundary value problem in, because $u_1 = 0$ at $y = \pm(l + \delta)$. Also, the boundary condition for the fluid at the inlet $x = 0$ is a prescribed pressure, i.e. $p(0, y) = P^0$. As in Appendix A the half-width of the channel in the deformed configuration is denoted by $\gamma(x)$.

The asymptotic expansion in the appendix depends on two parameters - P^0 and $\varepsilon = l/L$. Several numerical results comparing the computed values for γ_h, K_h with the analytical ones γ, K are given in Tables VI and VII. The first shows the L^2 norm of the error and the second one - the error at a fixed point $x = 0.2 * L$. The numerical solutions used were consequently refined, until the discretization error did not influence the first two digits of the results. It can be seen from Table VI

Table VI. Comparisons of analytical and asymptotic results for a long elastic channel in the L^2 norm.

P^0	$\varepsilon = 10$ ($l = 0.5, L = 5$)		$\varepsilon = 20$ ($l = 0.5, L = 10$)	
	$\frac{\ \gamma_h - \gamma\ _{L^2}}{\ \gamma\ _{L^2}}$	$\frac{ K_h - K }{K}$	$\frac{\ \gamma_h - \gamma\ _{L^2}}{\ \gamma\ _{L^2}}$	$\frac{ K_h - K }{K}$
0.32	2.41×10^{-3}	6.63×10^{-3}	8.47×10^{-4}	1.82×10^{-3}
0.16	1.19×10^{-3}	3.33×10^{-3}	4.21×10^{-4}	1.06×10^{-3}
0.08	5.96×10^{-4}	1.65×10^{-3}	2.10×10^{-4}	5.34×10^{-4}
0.04	2.98×10^{-4}	8.19×10^{-4}	1.05×10^{-4}	2.68×10^{-4}

that formulae (A.44) and (A.47) are in very good agreement with the numerically computed solution to the FSI problem. This indicates that as $\varepsilon \rightarrow 0$ and $h \rightarrow 0$ both

Table VII. Point-wise comparisons of analytical and asymptotic results of a long elastic channel.

P^0	$\varepsilon = 10$ ($l = 0.5, L = 5$)		$\varepsilon = 20$ ($l = 0.5, L = 10$)	
	$\frac{ \gamma_h(1) - \gamma(1) }{ \gamma(1) }$	$\frac{ K_h(1) - K(1) }{ K(1) }$	$\frac{ \gamma_h(2) - \gamma(2) }{ \gamma(2) }$	$\frac{ K_h(2) - K(2) }{ K(2) }$
0.32	2.59×10^{-4}	6.02×10^{-4}	8.97×10^{-5}	2.57×10^{-4}
0.16	4.58×10^{-5}	1.71×10^{-5}	2.49×10^{-5}	8.15×10^{-5}
0.08	5.02×10^{-6}	5.79×10^{-5}	5.85×10^{-6}	6.56×10^{-6}
0.04	1.06×10^{-5}	4.38×10^{-5}	4.76×10^{-7}	1.00×10^{-6}

the asymptotic and numerical solution to the FSI problem converge to the actual one.

3.3. Flow through a channel with elastic segment

In this problem a channel with a deformable segment (Figure 59) is considered. The channel has height $H = 1$ and length $L = 14$ and an elastic segment is located in the middle of the top channel wall. The segment has of length 10, thickness 0.1 and elastic properties $E = 12 * 10^6$ and $\nu = 0.2$. The fluid viscosity and density are again $\mu = \rho = 1$. The segment is built into the rest of the wall at its two ends. The appropriate boundary conditions at the ends of the segment are therefore homogeneous Dirichlet boundary conditions for both displacements. Natural boundary conditions (zero tractions) are imposed on the top side of the segment. The lower side is the fluid-structure interface Γ_0^f . The input flow velocity is again a developed Poiseuille flow: $v_x = \frac{6}{14}Qy(1 - y)$, with

$$Q = \int_{\Omega} v_x d\mathbf{x}$$

being the total volumetric flow rate. The right end of the channel has an outflow boundary condition ($\mathbf{T}^f = \mathbf{0}$). The purpose of this problem is to investigate the permeability of the channel at various values for Q . In the case of an entirely rigid

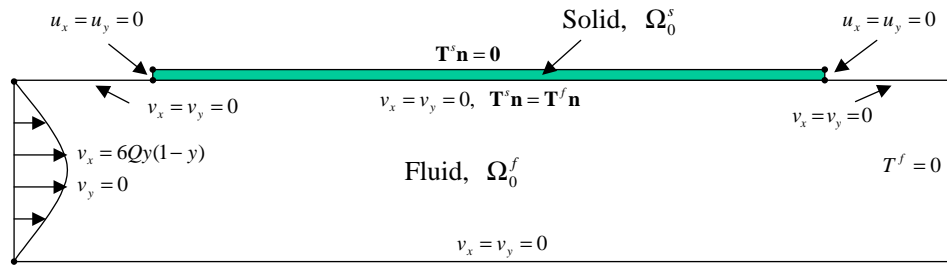


Fig. 59. Geometry of a channel with an elastic segment (Figure not drawn to scale).

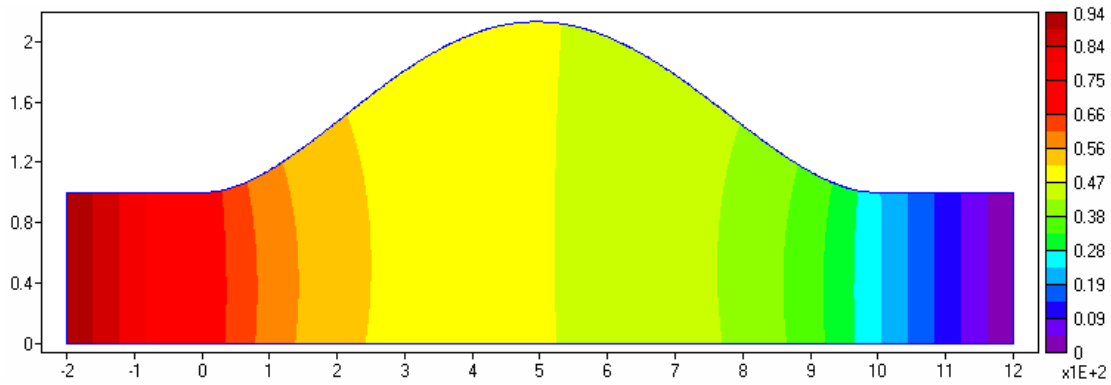
channel the pressure drop $\Delta p/L$ is proportional to the mass flux Q . Since the coupled problem is nonlinear it is expected that for sufficiently large deformations of the interface the resulting mass flow for a given pressure drop will depart from the linear relationship of the Darcy law. The fluid structure problem is solved for several different values of Q and the resulting pressure, pressure gradients in the fluid and displacements in the solid are computed. Two measures for permeability are considered. One is the ratio of the average velocity and average pressure gradient:

$$\bar{K} = \frac{Q}{\int_{\Omega} \frac{\partial p}{\partial x} d\mathbf{x}}$$

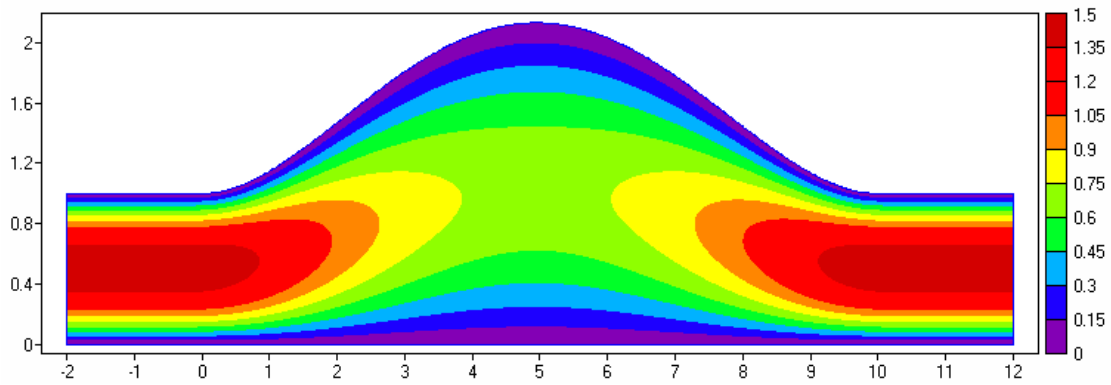
which is similar to the standard homogenization results for flow through rigid skeleton. Since the geometry allows only nonzero net flow in the x direction it also makes sense to consider the pressure drop as an alternative the average pressure gradient:

$$\hat{K} = \frac{Q}{\Delta p/L}.$$

While the pressure gradient can be readily calculated from the flow solution, it is not immediately clear how to evaluate the pressure drop across the channel. However, thanks to the selection of material parameters and geometry dimensions the inlet is sufficiently away from the segment so the change in the flow downstream does not



(a) Pressure profile (Figure not drawn to scale).



(b) Profile of the horizontal velocity component (Figure not drawn to scale).

Fig. 60. Final configuration of the fluid domain Ω^f for $Q = 15$.

Table VIII. Permeability of a channel with elastic segment as function of the flow rate.

Q	$\int_{\Omega} \frac{\partial p}{\partial x} d\mathbf{x}$	$p _{inlet}$	$Q / \int_{\Omega} \frac{\partial p}{\partial x} d\mathbf{x}$	$\frac{Q}{\Delta p/L}$	$\max u_y$
14.0000	-106.0853266	-93.8605	0.13196	0.14916	1.161
7.0000	-60.3754287	-53.9478	0.11594	0.12975	0.651
3.5000	-33.6888175	-30.8411	0.10389	0.11348	0.373
1.7500	-18.3178061	-17.2482	0.09553	0.10146	0.209
0.8750	-9.7023494	-9.35044	0.09018	0.09358	0.113
0.4375	-5.0280238	-4.923492	0.08701	0.08889	

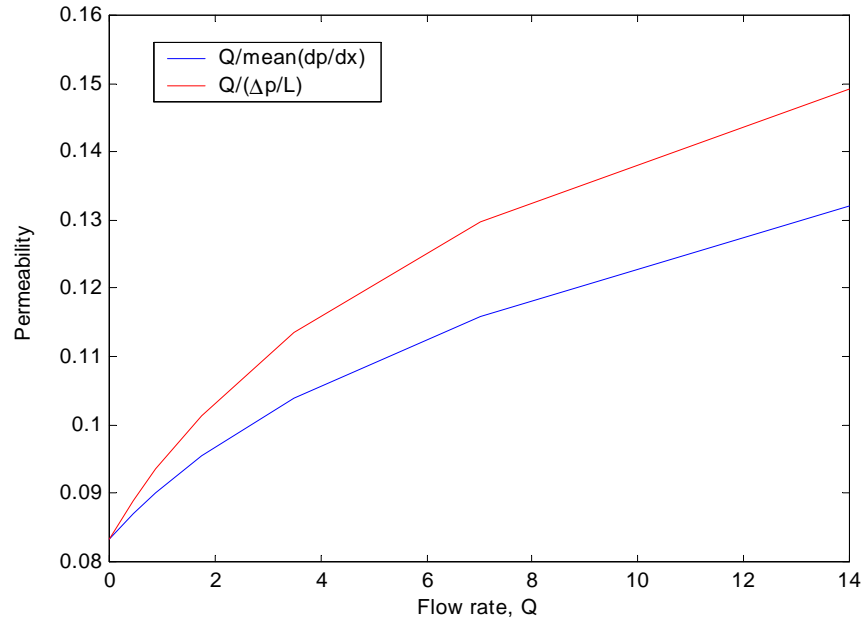


Fig. 61. Permeability of a channel with elastic segment.

affect much the inlet pressure distribution, which is essentially constant along the y direction. Similarly, the outlet is sufficiently separated from the end of the elastic segment and the flow has time to redevelop to the parabolic Poiseuille distribution as can be seen on Figures 60(a) and 60(b). The results are summarized on Table VIII and it is clear that both permeability measures behave nonlinearly as Q is varied. This can also be seen on Figure 61 which shows \bar{K} and \hat{K} as functions of Q . As $Q \rightarrow 0$ both of them tend to the permeability of a straight, rigid channel which can be calculated directly from the Poiseuille solution and in this case is equal to $1/12$.

4. Modelling of flow regulating SMA device

In this section a built-in SMA beam within a rigid channel is studied as the first step in design a temperature actuated flow-regulating device. The device considered

here is an SMA segment built-in into a 2D rigid channel. The initial geometry of the channel can be seen on Figure 62. The dimensions of the channel are $0.5\text{cm} \times 5\text{cm}$ and the SMA occupies length of 3cm . The shape of the SMA was selected so that the thickness near the middles of the channel is half of its usual one. The lower part of the SMA segment (i.e. the interface Γ_0^f) is the graph of the function $1 - 0.5 \exp(-2 * (x - 1.5)^2) + 0.5 \exp(-2 * (1.5)^2)$, $x \in [0, 3]$. The vertical thickness of the segment is 0.1cm . The boundary conditions applied are analogous to that of the previous section, with the exception that the pressure ($p_0 = 3.5\text{MPa}$) is specified at the channel inlet, instead of velocity and the fluid has unit viscosity. The properties of the SMAs are the same as used in Chapter IV and are listed in Table IV.

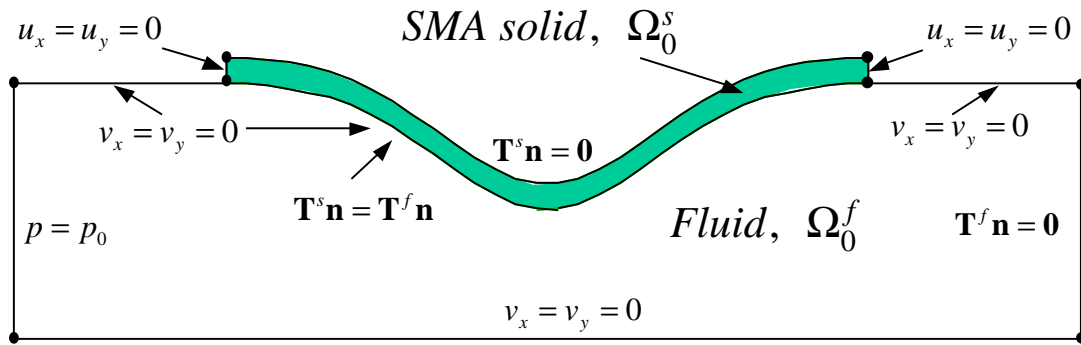


Fig. 62. Geometry of SMA flow regulating device. Shown are the boundary conditions and reference configuration of both domains.

The motivation for this geometry is driven by the need to have a device with higher permeability at lower temperatures compared to its permeability at high temperature. The simulation is performed in the following way. First the stationary solution of the FSI problem is sought at a temperature $T = 320 \text{ }^\circ\text{K}$, which is above the austenitic finish temperature of the SMA. The reference configuration of the SMA is therefore stress free and in the austenitic phase. Once the solution is found the temperature of the fluid is decreased, while the pressure differential acting at the

ends of the channel is kept fixed. It is assumed that the heat conduction between the fluid and the solid is instantaneous, so that the temperature of the fluid becomes the temperature of the SMA. As it is decreased, the SMA undergoes the $A \rightarrow M^d$ phase transformation, associated with inelastic strain and change in material properties. As it becomes more compliant and develops inelastic strains, therefore it is expected that the same pressure, which was in equilibrium with the SMA at higher temperature will opens up the channel, leading to increase in the flow rate.

Solving a FSI problem with a nonlinear solid is not fundamentally different from a linear one. The coupling between the solid and the fluid comes through interface conditions on the interface and are not related to the type of nonlinearities involved in the field equations for either of the solid or fluid domain. The main characteristic of the iterative method of Algorithm (2.1.1) is finding balance of momentum on the interface Γ_0 and the nature of the solid solver is not important. Using a nonlinear solid instead of a linearized one involves replacing the linear solver of step 4 of (2.1.1) with a nonlinear one (see Appendix B for a short summary of a nonlinear, displacement based FEM). When more than one load steps are involved, care has to be taken that the correct initial state of the material is used every time algorithm (2.1.1) is started. The reference configuration of the domain however does not need to be changed between different loading step as is the case here, when the FSI problem is solved for consecutively decreasing temperatures.

The main result of the simulation was in agreement with the expectations (Figure 63). At the initial temperature of 320 a small region near the left built-in edge of the beam transformed partially, and small phase transformation was observed in the region left of the segments lowest point (Figure 64(a)). Most of the displacement was observed in the left part of the segment and is due to factors: the higher pressure in that part of the channel (Figure 64(c)) and the bending moment which comes from

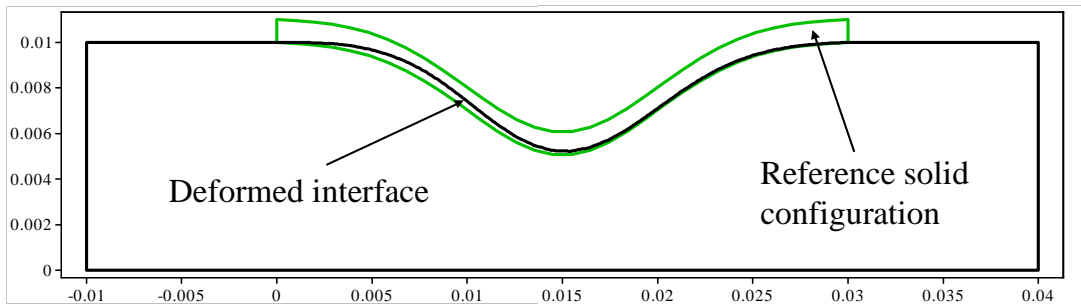
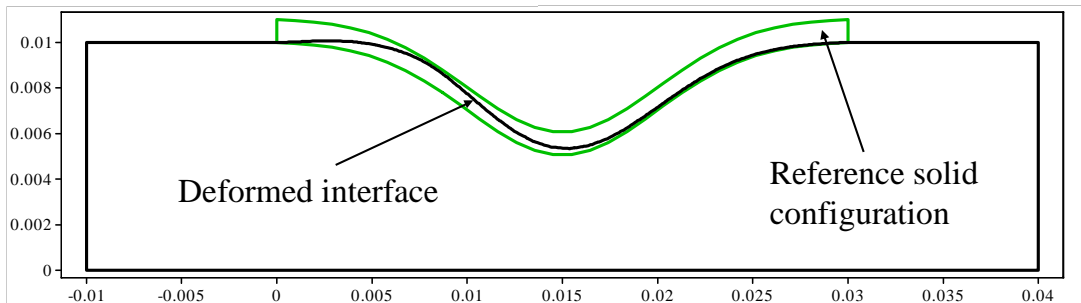
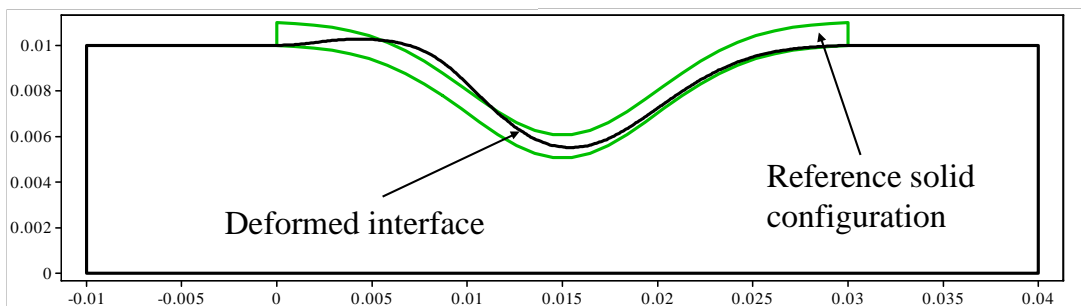
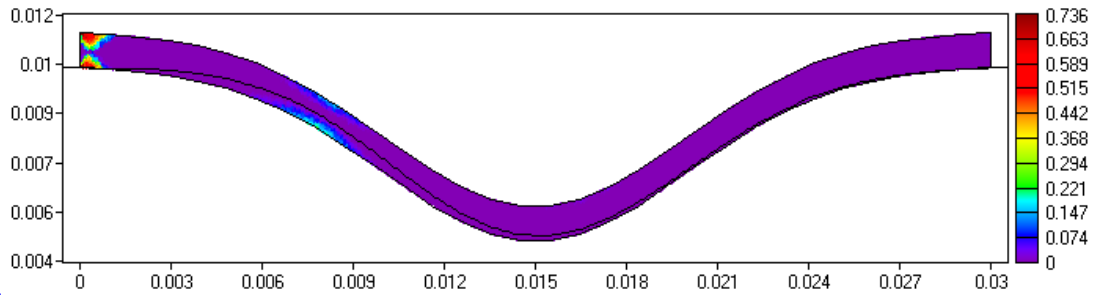
(a) $T = 320 \text{ }^\circ\text{K}$ (b) $T = 310 \text{ }^\circ\text{K}$ (c) $T = 300 \text{ }^\circ\text{K}$

Fig. 63. Geometry of the initial (green) and deformed configuration (black) of the channel while cooling. Shown are the positions at three different temperatures.

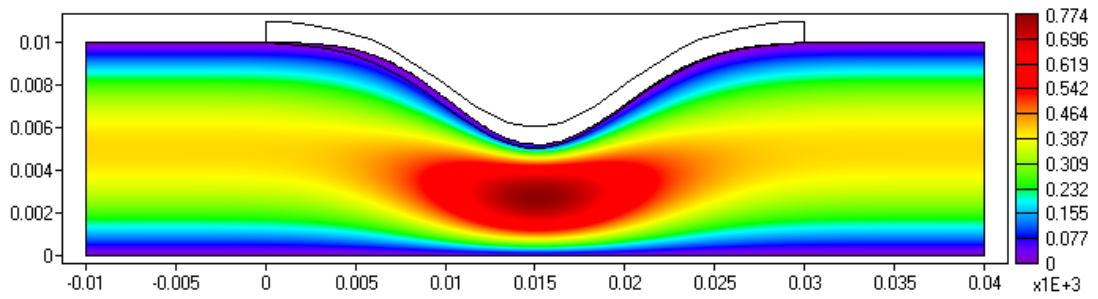
the pressure acting on the section between the left end and the SMA's lowest point. Cooling of the channel resulted in further stress-induced phase transformation and resulting softening of the segment and its movement upwards as seen from Figure 65. Most of the upward displacement occurs near in the region prior to the segment's lowest point and is generally in direction normal to the interface Γ^I . The displacement of the lowest point of the segment however is not as big as was desired - about 0.6mm at $300\text{ }^\circ\text{K}$, or approximately 2.5% opening of the channels at its narrowest point.

The main motivation for this numerical simulation was to assess the ability of SMAs to change the permeability of the channel. The mass flow rate Q , as a function of temperature is shown on Figure 66. At the beginning of the cooling process ($T = 320\text{ }^\circ\text{K}$), the flow rate is approximately $0.274\text{m}^3/\text{s}$ and when the temperature has reached $T = 295\text{ }^\circ\text{K}$ it has increased up to $0.317\text{m}^3/\text{s}$. The increase is about 15%, which is probably related to the displacement of the lowest point of the SMA, where the channel has smallest cross-section.

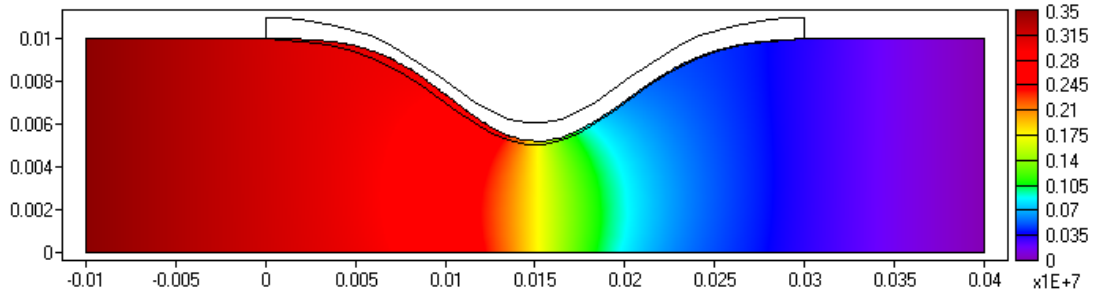
The demonstrated change in mass flux should be viewed as a first approximation to a real flow-regulating device. The main reason for the little change is the fact that most of the deformation in the SMA beam happens near the left built-in end. A broader study is therefore necessary to find an optimal design. The center region influences the permeability of the channel most and therefore a beam which is thinner in that region may lead to higher change in permeability as temperature changes. The effects of boundary condition is also important and needs to be taken into account. As a final remark, the channel geometry used here is of the simplest possible type and probably does not reflect a practical design. Therefore, continued work in building an temperature actuated SMA flow-regulator should start from a realistic 3-D valve designs.



(a) Detwinned volume fraction (c_2) plotted at the reference configuration. Shown is also the deformed interface position

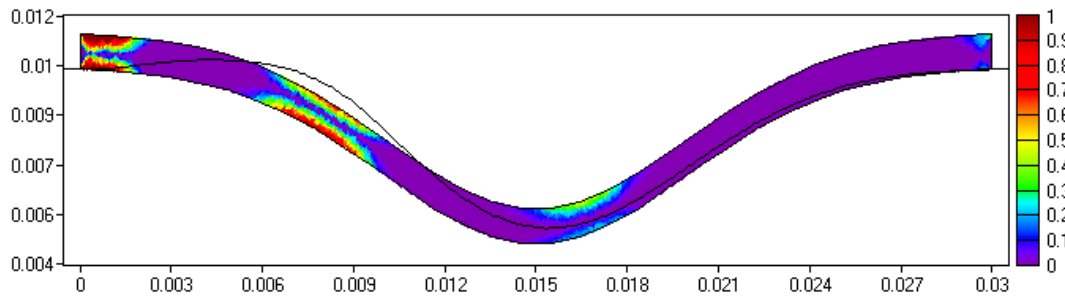


(b) Pressure profile in Ω^f . The initial solid geometry ($\partial\Omega_0^s$) is also shown.

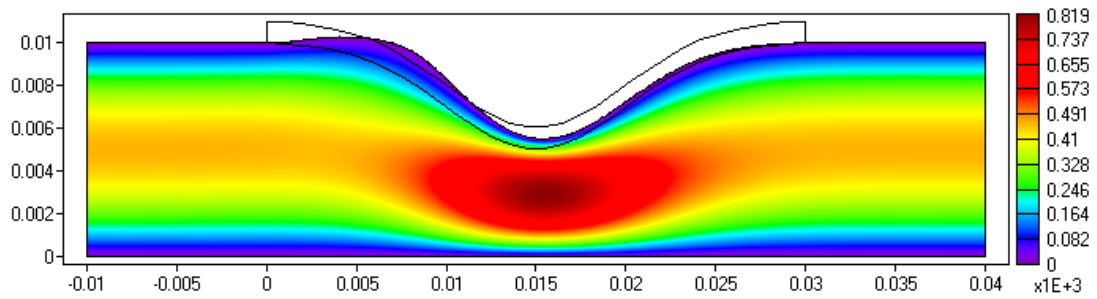


(c) Horizontal velocity component. The initial solid geometry ($\partial\Omega_0^s$) is also shown.

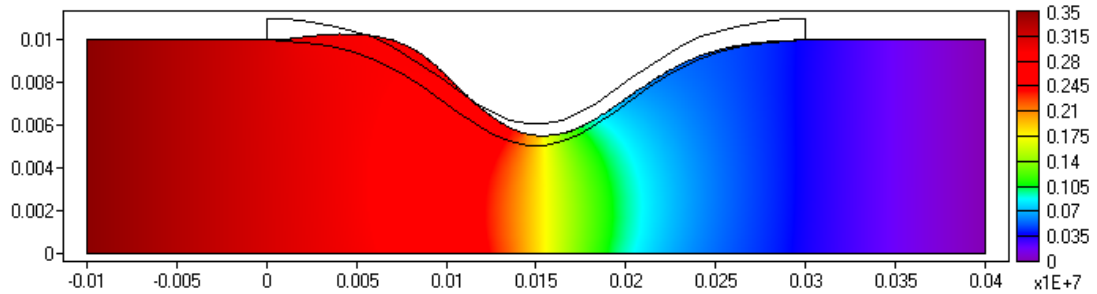
Fig. 64. FSI Solution at $T = 320 \text{ }^\circ\text{K}$.



(a) Detwinned volume fraction (c_2) plotted at the reference configuration. Shown is also the deformed interface position



(b) Pressure profile in Ω^f . The initial solid geometry ($\partial\Omega_0^s$) is also shown.



(c) Horizontal velocity component. The initial solid geometry ($\partial\Omega_0^s$) is also shown.

Fig. 65. FSI Solution at $T = 300 \text{ }^\circ\text{K}$.

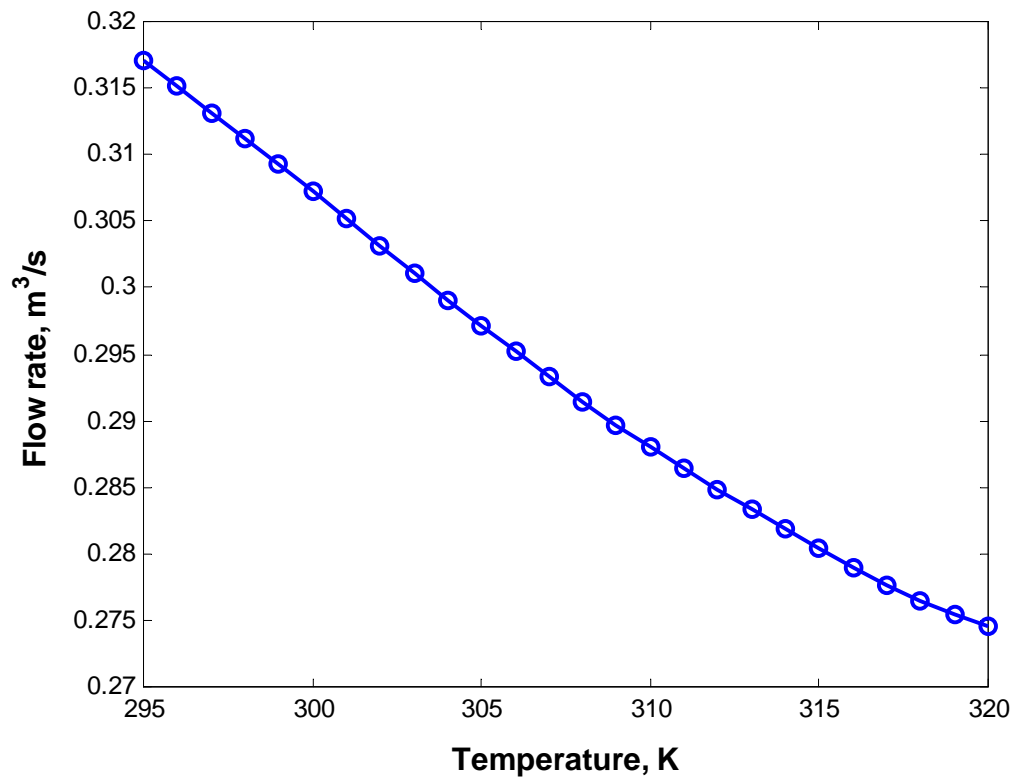


Fig. 66. Dependence of the flow rate in the channel as a function of temperature during cooling.

CHAPTER VI

SUMMARY AND CONCLUSIONS

A new 3-D constitutive model for polycrystalline SMAs based on thermodynamic potentials is presented. The model can account for both development of stress induced martensite directly from austenite (pseudoelasticity) as well as detwinning of twinned martensite. This is accomplished by describing the material state as a mixture of three phases - twinned martensite, detwinned martensite and austenite and by using the three possible "reactions" between these phases as internal variables.

The model is made consistent with a modified phase diagram in stress-temperature space. A key new experimental finding is the existence of separate reverse transformation temperatures for detwinned and twinned martensite. This is obtained through a series of calorimetric measurements and is incorporated in the model. The phase diagram also incorporates a single transformation strip for the $M^t \rightarrow M^d$ inelastic deformation over the temperature range $T \leq A_f^t$. It also assumed the same temperature independent transformation regions for the $A \rightarrow M^t$ and $M^t \rightarrow A$ phase transformations. These modifications allow for a robust model that can simulate complex thermomechanical loading paths, such as cyclic paths involving a mixture of all the three phases.

The constitutive model was numerically implemented using return mapping algorithms. The implementation was integrated into an numerical implementation and tested for several model problems. Presented in this work are two cooling/heating loading paths of a rod in uniaxial stress state. They demonstrate the ability of the model to handle the development of simultaneous transformation, the cut-off stress for the forward detwinning deformation and the behavior of mixtures of the three phases in cyclic paths. In order to demonstrate the 3-D capabilities of the model a

complex loading path for a perforated square under conditions of plane strain was also presented.

An FEM based Fluid-Structure solver is also presented. The solver successfully utilized a Dirichlet-Neumann iterative a scheme for solving the FSI problem. A numerical computation is performed which are used to verify the numerical solutions in comparisons with an asymptotic solution to the FSI problem for a channel geometry. This increases the degree of confidence with which such numerical methods can be used in the absence of solid mathematical understanding of their properties. The numerical approach was also extended to FSI problems involving SMA materials. The strong temperature dependence of the SMA was used to simulate a precursor temperature sensitive flow regulating devices. The 3-D SMA model developed in this work was especially suited for simulating such devices since it captures correctly the phenomena occurring in SMAs over a wide temperature range.

There are several areas for future work on this types of models. First, the selection made for the independent variables of the hardening functions are not the only possible ones. The current selection imposes certain constraints on the arrangement of the transformation regions in the phase diagram. Other choices, for example a dependence of the $A \leftrightarrow M^t$ hardening function on the total amount of austenite, rather than twinned martensite is possible and should be explored. Further, there are class of SMA materials for which the critical critical temperatures M_s , M_f , A_s^t , A_f^t define overlapping regions. The current model should be tested in such cases and modified, if necessary.

Also, the phase diagram itself evolves as the material is cycled through a certain thermomechanical loading path. The evolution is fairly well understood in the special case of pseudoelastic loading paths. More general cases however are not explored either from a modeling or an experimental point of view. It should me noted that

translational movements of the transformation strips of the phase diagram can be accomplished by using ξ_i instead of c_i as the independent variables for the hardening functions.

It should be noted that no rigorous mathematical analysis has been performed on the return-mapping algorithms used for the implementation of the SMA model in this work. Such work, while time-consuming, will be tremendously useful researchers interested in further developing the field of SMA models.

Work on FSI problems can also progress in multiple directions. Among this one should point out the detailed engineering analysis of SMA based flow regulating devices, the incorporation of heat transfer in the simulations and the analysis and development of homogenization methods for coupled fluid and porous SMA problem. A definite plus, would also be a mathematical analysis of the numerical algorithms used for the FSI problem.

REFERENCES

- Abeyaratne, R., Kim, S.-J., 1997. Cyclic effects in shape-memory alloys: A one-dimensional continuum model. *Int. J. Solids Struct.* 34 (25), 3273–3289.
- Abeyaratne, R., Knowles, J. K., 1991. Kinetic relations and the propagation of phase boundaries in solids. *Arch. Ration. Mech. Anal.* 114 (2), 119–154.
- Abeyaratne, R., Knowles, J. K., 1994a. Dynamics of propagating phase boundaries: Adiabatic theory for thermoelastic solids. *Physica D* 79, 269–288.
- Abeyaratne, R., Knowles, J. K., 1994b. Dynamics of propagating phase boundaries: Thermoelastic solids with heat conduction. *Arch. Ration. Mech. Anal.* 126 (3), 203–230.
- Abeyaratne, R., Knowles, J. K., 1997. Impact-induced phase transitions in thermoelastic solids. *Phil. Trans. R. Soc. Lond. A* 355, 843–867.
- Achenbach, M., 1989. Model for an alloy with shape memory. *Int. J. Plasticity* 5, 371–395.
- Anand, L., Gurtin, M., 2003. Thermal effects in the superelasticity of crystalline shape-memory materials. *J. Mech. Phys. Solids* 51 (6), 1015–1058.
- Argyris, J. H., Doltsinis, J. S., Knudson, W. C., Vaz, L. E., William, K. J., 1979. Numerical solution of transient nonlinear problems. *Computer Methods in Applied Mechanics and Engineering* 17/18, 341–409.
- Auricchio, F., 2001. A robust integration-algorithm for a finite-strain shape-memory-alloy superelastic model. *Int. J. Plasticity* 17, 971–990.

- Auricchio, F., Taylor, R. L., Lubliner, J., 1997. Shape-memory alloys: Macromodelling and numerical simulations of the superelastic behavior. *Computer Methods in Applied Mechanics and Engineering* 146, 281–312.
- Axelsson, O., Barker, V., 1984. *Finite Element Solution of Boundary Value Problems—Theory and Computations*. Academic Press, London.
- Axelsson, O., Neytcheva, M., 2003. Preconditioning methods for linear systems arising in constrained optimization problems. *Numerical Linear Algebra with Applications* 10, 3–31.
- Bakhvalov, N., Panasenko, G., 1990. *Homogenization: Averaging Processes in Periodic Media*. Kluwer Academic Publishers, Dordrecht.
- Ball, J. M., James, R. D., 1987. Fine phase mixture as minimizers of energy. *Arch. Ration. Mech. Anal.* 100, 13–52.
- Bazant, Z. P., 1984. Microplane model for strain controlled inelastic behavior. In: Desai, C. S., Gallagher, R. H. (Eds.), *Mechanics of Engineering Materials*. Wiley, London, Ch. 3, pp. 45–59.
- Bekker, A., Brinson, L. C., 1997. Temperature-induced phase transformation in a shape memory alloy: Phase diagram based kinetics approach. *J. Mech. Phys. Solids* 45 (6), 949–988.
- Bekker, A., Brinson, L. C., 1998. Phase diagram based description of the hysteresis behavior of shape memory alloys. *Acta Materialia* 46 (10), 3649–3665.
- Bekker, A., Victory, J. C., Popov, P., Lagoudas, D. C., 2002. Impact induced propagation of phase transformation in a shape memory alloy rod. *Int. J. Plasticity* 18 (11), 1425–1647.

- Bell, J., 1960. Propagation of large amplitude waves in annealed aluminum. *J. Appl. Phys.* 31, 227–282.
- Bell, J., 1962. Experimental study of dynamic plasticity at elevated temperatures. *Experimental Mechanics* 2, 1–6.
- Bensoussan, A., Lions, J., Papanicolaou, G., 1978. *Asymptotic Analysis for Periodic Structures*. Vol. 5 of *Studies in Mathematics and Its Applications*. North-Holland, Amsterdam.
- Berveiller, M., Morreale, J., Reubrez, E., 1994. Comportement élastoplastique des aciers lors de la mise en forme: Théorie micromécanique, simulations numériques et résultats expérimentaux. *Revue Européennes des éléments finis* 3 (4), 491–514.
- Berveiller, M., Patoor, E., Buisson, M., 1991. Thermomechanical constitutive equations for shape memory alloys. *Journal de Physique IV* 1, C.4,387, European Symposium on Martensitic Transformation and Shape Memory Properties.
- Birman, V., 1997. Review of mechanics of shape memory alloy structures. *Appl. Mech. Rev.* 50 (11), 629–645.
- Blaheta, R., 1994. Displacement decomposition - incomplete factorization preconditioning techniques for linear elasticity problems. *Numerical Linear Algebra with Applications* 1, 107–128.
- Bo, Z., Lagoudas, D. C., 1999a. Thermomechanical modeling of polycrystalline SMAs under cyclic loading, Part I: Theoretical derivations. *Int. J. Eng. Sci.* 37, 1089–1140.
- Bo, Z., Lagoudas, D. C., 1999b. Thermomechanical modeling of polycrystalline SMAs under cyclic loading, Part III: Evolution of plastic strains and two-way memory effect. *Int. J. Eng. Sci.* 37, 1175–1204.

- Bo, Z., Lagoudas, D. C., 1999c. Thermomechanical modeling of polycrystalline SMAs under cyclic loading, Part IV: Modeling of minor hysteresis loops. *Int. J. Eng. Sci.* 37, 1205–1249.
- Bodner, S., Clifton, J., 1967. An experimental investigation of elastic-plastic pulse propagation in aluminum rods. *J. App. Mech.* 34 (1), 91–99.
- Boyd, J. G., Lagoudas, D. C., 1994a. A constitutive model for simultaneous transformation and reorientation in shape memory materials. In: *Mechanics of Phase Transformations and Shape Memory Alloys*. Vol. AMD 189/PVP 292. ASME, Chicago, pp. 159–172.
- Boyd, J. G., Lagoudas, D. C., 1994b. Thermomechanical response of shape memory composites. *J. Intell. Mater. Systems Struct.* 5, 333–346.
- Boyd, J. G., Lagoudas, D. C., 1996a. A thermodynamic constitutive model for the shape memory materials. Part I. The monolithic shape memory alloys. *Int. J. Plasticity* 12 (6), 805–842.
- Boyd, J. G., Lagoudas, D. C., 1996b. A thermodynamical constitutive model for shape memory materials. Part II. The SMA composite material. *Int. J. Plasticity* 12 (7), 843–873.
- Bramble, J., Pasciak, J., 1997. Iterative techniques for time dependent stokes problems. *Computers and Mathematics with Applications* 33 (1/2), 13–30.
- Bramble, J., Zhang, X., 2000. The analysis of multigrid methods. In: Ciarlet, P., Lions, J. (Eds.), *Techniques of Scientific Computing*. Vol. 7 of *Handbook of Numerical Analysis*. North-Holland, Amsterdam, pp. 173–415.

- Brinson, L. C., 1993. One-dimensional constitutive behavior of shape memory alloys: Thermomechanical derivation with non-constant material functions and redefined martensite internal variable. *J. Intell. Mater. Systems Struct.* 4, 229–242.
- Brinson, L. C., Lammering, R., 1993. Finite element analysis of the behavior of shape memory alloys and their applications. *Int. J. Solids Struct.* 30 (23), 3261–3280.
- Brocca, M., Brinson, L. C., Bažant, Z. P., 2002. Three-dimensional constitutive model for shape memory alloys based on microplane model. *J. Mech. Phys. Solids* 50, 1051–1077.
- Chen, Y.-C., Lagoudas, D. C., 2000. Impact induced phase transformation in shape memory alloys. *J. Mech. Phys. Solids* 48 (2), 275–300.
- Chiddister, J., Malvern, L., 1963. Compression-impact testing of aluminum at elevated temperatures. *Experimental Mechanics* 3, 81–90.
- Christensen, R. M., 1991. *Mechanics of Composite Materials*. Krieger Publishing Company, Malabar, FL.
- Ciarlet, P., 2002. *The Finite Element Method for Elliptic Problems*. No. 40 in *Classics in Applied Mathematics*. SIAM, Philadelphia.
- Clifton, R. J., Bodner, S. R., 1966. An analysis of longitudinal elastic-plastic pulse propagation. *Trans. ASME* 33, 248–255.
- Cory, J., McNichols, J., 1985. Nonequilibrium thermostatics. *J. Appl. Phys.* 58, 3282–3294.
- Cross, W. B., Kariotis, A. H., Stimler, F. J., 1969. Nitinol characterization study. Tech. Rep. CR-1433, NASA, Langley, VA.

- De Vos, J., Aernoudt, E., Delaey, L., 1978. The crystallography of the martensitic transformation of BCC into 9R: A generalized mathematical model. *Z. Metallkde* 69 (H7), 438–444.
- Delaey, L., 1990. Diffusionless transformations. In: Cahn, R. W., Haasen, P., Kramen, E. J. (Eds.), *Phase Transformations in Materials*. Vol. 5 of *Material Science and Technologies*. VCH Publishers, New York, Ch. 6, pp. 339–404.
- Escobar, J. C., Clifton, R. J., 1993. Pressure-shear impact induced phase transformation in Cu-14.44Al-4.19Ni single crystals. *Material Science and Engineering A* 170, 125–142.
- Falk, F., 1990. Pseudoelastic stress-strain curves of polycrystalline shape memory alloys calculated from single crystal data. *Int. J. Eng. Sci.* 27 (3), 277–284.
- Feng, Z. C., Li, D. Z., 1996. Dynamics of a mechanical system with a shape memory alloy bar. *J. Intell. Mater. Systems Struct.* 7, 399–410.
- Fischer, F. D., Oberaigner, E. R., Tanaka, K., Nishimura, F., 1998. Transformation induced plasticity revised, an updated formulation. *Int. J. Solids Struct.* 35 (18), 2209–2227.
- Fosdick, R., Ketema, Y., 1998. Shape memory alloys for passive vibration damping. *J. Intell. Mater. Systems Struct.* 9, 854–870.
- Funakubo, H., 1987. *Shape Memory Alloys*. Gordon and Bleach, New York.
- Gao, X., Brinson, L. C., 2002. A simplified multivariant SMA model based on invariant plane nature of martensitic transformation. *J. Intell. Mater. Systems Struct.* 13 (12), 795–810.

- Gaymonat, G., Müller, S., Triantafyllidis, N., 1993. Homogenization of nonlinearly elastic materials, macroscopic bifurcation and macroscopic loss of rank-one convexity. *Arch. Rational Mech. Anal.* 122, 231–290.
- Girault, V., Raviart, P., 1986. *Finite Element Methods for Navier-Stokes Equations.* Springer-Verlag, Berlin.
- Godlewsky, E., Raviart, P., 1996. *Numerical Approximation of Hyperbolic Systems of Conservation Laws.* Applied Mathematical Sciences. Springer-Verlag, New York.
- Golub, G. H., Van Loan, C. F., 1996. *Matrix Computations.* Johns Hopkins University Press, Baltimore.
- Govindjee, S., Hall, G. J., 2000. A computational model for shape memory alloys. *Int. J. Solids Struct.* 37, 735–760.
- Govindjee, S., Miehe, C., 2001. A multi-variant martensitic phase transformation model: Formulation and numerical implementation. *Computer Methods in Applied Mechanics and Engineering* 191, 215–238.
- Graesser, E. J., 1995. Effect of intrinsic damping on vibration transmissibility of nickel-titanium shape memory alloy springs. *Metallurgical and Materials Transactions A* 26, 2791–2796.
- Graesser, E. J., Cozzarelli, F. A., 1991. Shape memory alloys as new materials for aseismic isolation. *Journal of Engineering Mechanics* 117 (11), 2590–2608.
- Graff, K., 1975. *Wave Motion in Elastic Solids.* Oxford University Press, London.
- Gresho, P. M., Sani, R. L., 1998. *Incompressible Flow and the Finite Element Method: Advection - Diffusion and Isothermal Laminar Flow.* John Wiley & Sons, Inc, Chichester, UK.

- Gurtin, M. E., 1981. *An Introduction to Continuum Mechanics*. Academic Press, San Diego, CA.
- Hershey, A. V., 1954. The elasticity of an isotropic aggregate of anisotropic cubic crystals. *J. App. Mech.* 21, 236–240.
- Hesteness, M., Stiefel, E., 1952. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards, Section B* 49, 409–436.
- Hill, R., 1965. Continuum micromechanics of elastoplastic polycrystals. *J. Mech. Phys. Solids* 13, 89–101.
- Hill, R., 1967. The essential structure of constitutive laws for metal composites and polycrystals. *J. Mech. Phys. Solids* 15, 79–95.
- Iliev, O., Mikelic, A., Popov, P., 2004. Fluid-structure interaction in deformable porous media. Tech. Rep. 65, Fraunhofer ITWM, Kaiserslautern, Germany.
- Jackson, C. M., Wagner, H. J., Wasilewski, R. J., 1972. 55-nitinol—The alloy with a memory: Its physical metallurgy, properties and applications. Tech. Rep. NASA SP-5110, NASA, Technology Utilization Office, Washington, DC.
- Jimenez-Victory, J. C., 1999. Dynamic analysis of impact induced phase transformation in shape memory alloys using numerical techniques. Master's thesis, Texas A&M University, College Station, TX.
- Juhasz, L., Schnack, E., Hesebeck, O., Andra, H., 2002. Macroscopic modeling of shape memory alloys under non-proportional thermo-mechanical loadings. *J. Intell. Mater. Systems Struct.* 13, 825–836.

- Jung, Y., Papadopoulos, P., Ritchie, R., 2004. Constitutive modelling and numerical simulation of multivariant phase transformation in superelastic shape-memory alloys. *Int. J. Numer. Meth. Eng.* 60, 429–460.
- Kestin, J., Bataille, J., 1978. Irreversible thermodynamics of continua and internal variables. In: Provan, J. W. (Ed.), *Continuum Models of Discrete Systems: Proceedings of the Second International Symposium on Continuum Models of Discrete Systems*. Vol. 2. University of Waterloo Press, Waterloo, Ontario, pp. 39–67.
- Kestin, J., Rice, J. R., 1970. Paradoxes in the application of thermodynamics to strained solids. In: Stuart, E. B., Gal-Or, B., Brainard, A. J. (Eds.), *A Critical Review of Thermodynamics*. Mono Book Corp., Baltimore, pp. 275–298.
- Khan, M. M., Lagoudas, D. C., Mayes, J. J., Henderson, B. K., 2004. Pseudoelastic SMA spring elements for passive vibration isolation: Part I - modeling. *J. Intell. Mater. Systems Struct.* 15, 415–441.
- Kohl, M., Krevet, B., Just, E., 2002. Sma microgripper system. *Sensors and Actuators A97-98*, 646–652.
- Kolsky, H., 1949. An investigation of the mechanical properties of materials at very high rates of loading. *Proc. R. Soc. Lond. B* 62, 676–700.
- Kolsky, H., 1963. *Stress Waves in Solids*. Dover Publications, New York.
- Kolsky, H., Douch, L., 1962. Experimental studies in plastic wave propagation. *J. Mech. Phys. Solids* 10, 195–223.
- Kotil, T., Sehitoglu, H., Maier, H. J., Chumlyakov, Y. I., 2003. Transformation and detwinning induced electrical resistance variations in NiTiCu. *Material Science and Engineering A* 359, 280–289.

- Kröner, E., 1958. Berechnung der elastischen konstanten des vielkristalls aus den konstanten des einkristalls. *Z. Phys.* 151, 504–518.
- Kumar, P., Entchev, P. B., Lagoudas, D. C., 2003. Thermomechanical characterization of SMA actuators under cyclic loading. In: *Proceedings of IMECE'03. ASME International Mechanical Engineering Congress, Washington DC*, pp. 15–21.
- Lagoudas, D. C., Bhattacharya, A., 1997. On the correspondence between micromechanical models for isothermal pseudoelastic response of shape memory alloys and Preisach model for hysteresis. *Mathematics and Mechanics of Solids* 2 (4), 405–440.
- Lagoudas, D. C., Bo, Z., 1999. Thermomechanical modeling of polycrystalline SMAs under cyclic loading, Part II: Material characterization and experimental results for a stable transformation cycle. *Int. J. Eng. Sci.* 37, 1205–1249.
- Lagoudas, D. C., Bo, Z., Qidwai, M. A., 1996. A unified thermodynamic constitutive model for SMA and finite element analysis of active metal matrix composites. *Mech. Composite Mater. Struct.* 3, 153–179.
- Lagoudas, D. C., Boyd, J. G., Bo, Z., 1994. Micromechanics of active composites with SMA fibers. *ASME Journal of Material Science and Technology* 5, 337–347.
- Lagoudas, D. C., Entchev, P. B., 2004. Modeling of transformation-induced plasticity and its effect on the behavior of porous shape memory alloys. Part I: Constitutive model for fully dense SMAs. *Mech. Mater.* 36 (9), 865–892.
- Lagoudas, D. C., Khan, M. M., Mayes, J. J., Henderson, B. K., 2004. Pseudoelastic SMA spring elements for passive vibration isolation: Part II - simulations and experimental correlations. *J. Intell. Mater. Systems Struct.* 15, 443–470.

- Lagoudas, D. C., Sarh, K., Ravi-Chandar, K., Popov, P., 2003. Dynamic loading of polycrystalline shape memory alloy rods. *Mech. Mater.* 35 (7), 716.
- Lagoudas, D. C., Shu, S., 1999. Residual deformations of active structures with SMA actuators. *Int. J. Mech. Sci.* 41, 595–619.
- Lanczos, C., 1952. Solution of systems of linear equations by minimized iterations. *Journal of Research of the National Bureau of Standards* 49, 33–53.
- Landau, L., Lifshitz, E. M., Pitaevskii, L., 1976. *Statistical Physics*, 3rd Edition. Vol. 5 of *Theoretical Physics*. Nauka, Moscow, (Russian edition).
- Leclercq, S., Lexcelent, C., 1996. A general macroscopic description of the thermo-mechanical behavior of shape memory alloys. *J. Mech. Phys. Solids* 44 (6), 953–980.
- Lee, C. K., Mei, C. C., 1997. Re-examination of the equations of poroelasticity. *Int. J. Eng. Sci.* 35, 329–352.
- Lexcelent, C., Bourbon, G., 1996. Thermodynamical model for cyclic behaviour of Ti-Ni and Cu-Zn-Al shape memory alloys under isothermal undulated tensile tests. *Mech. Mater.* 24, 59–73.
- Lexcelent, C., Leclercq, S., Gabry, B., Bourbon, G., 2000. The two way shape memory effect of shape memory alloys: An experimental study and a phenomenological model. *Int. J. Plasticity* 16, 1155–1168.
- Lexcelent, C., Vivet, A., Bouvet, C., Calloch, S., Blanc, P., 2002. Experimental and numerical determinations of the initial surface of phase transformations under biaxial loading in some polycrystalline shape-memory alloys. *J. Mech. Phys. Solids* 50, 2717–2735.

- Liang, C., Rogers, C. A., 1990. One-dimensional thermomechanical constitutive relations for shape memory materials. *J. Intell. Mater. Systems Struct.* 1, 207–234.
- Liang, C., Rogers, C. A., 1992. A multi-dimensional constitutive model for shape memory alloys. *Journal of Engineering Mathematics* 26, 429–443.
- Lions, J., Magenes, 1968. *Problèmes aux Limites non Homogènes et Applications*. Vol. 1. Dunod, Paris.
- Lu, Z. K., Weng, G. J., 1998. A self-consistent model for the stress-strain behavior of shape-memory alloy polycrystal. *Acta Metallurgica* 46 (15), 5423–5433.
- MacQueron, J. L., Morin, M., Guénin, G., Planes, A., Elgueta, J., Castan, T., 1991. Atomic ordering and martensitic transition in a Cu-Zn-Al shape memory alloy. *Journal de Physique IV* 1 (9), C4–259–263.
- Malvern, L. E., 1969. *Introduction to the Mechanics of a Continuous Medium*. Prentice-Hall, Inc., Englewood Cliffs, NJ.
- Marketz, F., Fischer, F., 1995. A mesoscale study on the thermodynamic effect of stress on martensitic transformation. *Metallurgical and Materials Transactions A* 26A, 267–278.
- Marketz, F., Fischer, F., Tanaka, K., 1995. A computational micromechanics study on variant-coalescence in a Cu-Al-Ni shape memory alloy. *Journal de Physique IV, Colloque C2* 5, 537–542.
- McNichols, J.L. and Cory, J., 1987. Thermodynamics of nitinol. *J. Appl. Phys.* 61, 972–984.

- Miller, D., 2000. Thermomechanical characterization of plastic deformation and transformation fatigue in shape memory alloys. Ph.D. dissertation, Texas A&M University.
- Mori, T., Tanaka, K., 1973. Average stress in matrix and average energy of materials with misfitting inclusions. *Acta Metallurgica* 21, 571–574.
- Nečas, J., Hlaváček, I., 1981. *Mathematical Theory of Elastic and Elasto-Plastic Bodies*. Elsevier, Amsterdam.
- Nemat-Nasser, S., Hori, M., 1993. *Micromechanics: Overall Properties of Heterogeneous Materials*. Vol. 37 of North-Holland Series in Applied Mathematics and Mechanics. North-Holland, Amsterdam.
- Newmark, N. M., 1959. A method of computation for structural dynamics. *Journal of Engineering Mechanics Division, ASCE* , 67–94.
- Nishiyama, Z., 1978. *Martensitic Transformations*. Academic Press, San Diego.
- Oberaigner, E. R., Tanaka, K., Fischer, F. D., 1996. Investigation of the damping behavior of a vibrating shape memory alloy rod using a micromechanical model. *Smart Mater. Struct.* 3, 456–463.
- Olander, A., 1932. An electrochemical investigation of solid cadmium-gold alloys. *Journal of the American Chemical Society* 54, 3819–3833.
- Olson, G. B., Cohen, M., 1982. Stress assisted isothermal martensitic transformation: Application to TRIP steels. *Metall. Trans. A* 13A, 1907–1914.
- Ortin, J., Planes, A., 1988. Thermodynamic analysis of thermal measurements in thermoelastic martensitic transformations. *Acta Metallurgica* 36 (8), 1873–1889.

- Ortin, J., Planes, A., 1989. Thermodynamics of thermoelastic martensitic transformations. *Acta Metallurgica* 37 (5), 1433–1441.
- Ortiz, M., Pinsky, P. M., 1981. Global analysis methods for the solution of elastoplastic and viscoplastic dynamic problems. Tech. Rep. UCB/SESM 81/08, Department of Civil Engineering, University of California at Berkeley.
- Ortiz, M., Popov, E. P., 1985. Accuracy and stability of integration algorithms for elastoplastic constitutive relations. *Int. J. Numer. Meth. Eng.* 21, 1561–1576.
- Ortiz, M., Simo, J. C., 1986. An analysis of a new class of integration algorithms for elastoplastic constitutive relations. *Int. J. Numer. Meth. Eng.* 23, 353–366.
- Otsuka, K., Wayman, C., Nakai, K., Sakamoto, H., Shimizu, K., 1976. Superelasticity effects and stress-induced martensitic transformations in Cu-Al-Ni alloy. *Acta Metallurgica* 24, 207–226.
- Otsuka, K., Wayman, C. M., 1999. *Shape Memory Materials*. Cambridge University Press, Cambridge.
- Patoor, E., Eberhardt, A., Berveiller, M., 1987. Potentiel pseudoélastique et plasticité de transformation martensitique dans les mono et polycristaux métalliques. *Acta Metallurgica* 35, 2779–2789.
- Patoor, E., Eberhardt, A., Berveiller, M., 1988. Thermomechanical behaviour of shape memory alloys. *Arch. of Mech.* 40 (5-6), 775–794.
- Patoor, E., Eberhardt, A., Berveiller, M., 1996. Micromechanical modelling of superelasticity in shape memory alloys. *Journal de Physique IV* 6, C1–277–292.

- Patoor, E., Lagoudas, D. C., Entchev, P. B., Brinson, L. C., Gao, X., 2005. Shape memory alloys - part I: General properties and modelling of single crystals. *Mech. Mater.* Accepted.
- Pence, T., 1986. On the emergence and propagation of a phase boundary in an elastic bar with a suddenly applied end load. *J. Elasticity* (16), 3–42.
- Qidwai, M. A., Lagoudas, D. C., 2000a. Numerical implementation of a shape memory alloy thermomechanical constitutive model using return mapping algorithms. *Int. J. Numer. Meth. Eng.* 47, 1123–1168.
- Qidwai, M. A., Lagoudas, D. C., 2000b. On thermomechanics and transformation surfaces of polycrystalline NiTi shape memory alloy material. *Int. J. Plasticity* 16, 1309–1343.
- Rajagopal, K. R., Srinivasa, A. R., 1998. Mechanics of the inelastic behavior of materials. Part II: Inelastic response. *Int. J. Plasticity* 14 (10-11), 969–995.
- Rakhmatulin, K. A., 1945. Propagation of an unloading wave (in Russian). *Prikl. Mat. Mekh.* 9, 91–100.
- Raniecki, B., Lexcellent, C., 1994. RL-models of pseudoelasticity and their specification for some shape memory solids. *Eur. J. Mech. A/Solids* 13 (1), 21–50.
- Raniecki, B., Lexcellent, C., 1998. Thermodynamics of isotropic pseudoelasticity in shape memory alloys. *Eur. J. Mech. A/Solids* 17 (2), 185–205.
- Raniecki, B., Rejzner, J., Lexcellent, C., 2001. Anatomization of hysteresis loops in pure bending of ideal pseudoelastic sma beams. *Int. J. Mech. Sci.* 43, 1339–1368.
- Reddy, J. N., 1993. *An Introduction to the Finite Element Method*. McGraw-Hill, Boston.

- Rejzner, J., LExcellent, C., Raniecki, B., 2002. Pseudoelastic behaviour of shape memory alloy beams under pure bending: Experiments and modelling. *Int. J. Mech. Sci.* 44, 665–686.
- Renardy, M., Rogers, R., 1996. *An Introduction to Partial Differential Equations*, 2nd Edition. Texts in Applied Mathematics. Springer-Verlag, New York.
- Rice, J. R., 1971. Inelastic constitutive relation for solids: An internal variable theory and its application to metal plasticity. *J. Mech. Phys. Solids* 19, 433–455.
- Rice, J. R., Tracey, D. M., 1973. Computational fracture mechanics. In: Fenres, S. J. (Ed.), *Proceedings of the Symposium on Numerical Methods in Structural Mechanics*. Academic Press, New York.
- Roubicek, T., 2004. Models of microstructure evolution in shape memory alloys. In: Ponte Castaneda, P., Telega, J., Gambin, B. (Eds.), *Nonlinear Homogenization and its Application*. Vol. 170 of NATO Science Series II: Mathematics, Physics and Chemistry. Dordrecht, pp. 269–304.
- Roytburd, A. L., Pankova, M. N., 1985. Effect of external stresses on habitus orientation and substructure of stress-induced martensite plates in ferrous alloys. *Phys. Met. Metall.* 59, 131–140.
- Saad, Y., 1996. *Iterative Methods for Sparse Linear Systems*. PWS, New York.
- Saburi, T., Wayman, C. M., Takata, K., Nenno, S., 1980. The shape memory mechanism in 18R martensitic alloys. *Acta Metallurgica* 28, 15–32.
- Sakamoto, H., 2002. Distinction between thermal and stress-induced martensitic transformations and inhomogeneity in internal stress. *Metall. Trans.* 43 (9), 2249–2255.

- Sanchez-Palencia, E., 1980. Non-Homogeneous Media and Vibration Theory. Vol. 127 of Lecture Notes in Physics. Springer-Verlag, Berlin.
- Sato, Y., Tanaka, K., 1988. Estimation of energy dissipation in alloys due to stress-induced martensitic transformation. *Res Mechanica* 23, 381–392.
- Shewchuk, J., 2002. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications* 22 (1-3), 21–74.
- Simo, J. C., Govindjee, S., 1991. Non-linear b-stability and symmetry preserving return mapping algorithms for plasticity and viscoplasticity. *Int. J. Numer. Meth. Eng.* , 151–176.
- Simo, J. C., Hughes, T. J. R., 1998. *Computational Inelasticity*. Springer-Verlag, New York.
- Simo, J. C., Ortiz, M., 1985. A unified approach to finite deformation elastoplastic analysis based on the use of hyperelastic constitutive equations. *Computer Methods in Applied Mechanics and Engineering* 49, 221–245.
- Šittner, P., Lukáš, P., Neov, D., Lugovyy, D., 2003. Martensitic transformations in NiTi polycrystals investigated by in-situ neutron diffraction. *Material Science Forum* 426-432, 2315–2320.
- Šittner, P., Vokoun, D., Dayananda, G., Stalmans, R., 2000. Recovery stress generation in shape memory Ti₅₀Ni₄₅Cu₅ thin wires. *Material Science and Engineering A* 286, 298–311.
- Sun, Q. P., Hwang, K. C., 1993a. Micromechanics modeling for the constitutive behavior of polycrystalline shape memory alloys — I. Derivation of general relations. *J. Mech. Phys. Solids* 41 (1), 1–17.

- Sun, Q. P., Hwang, K. C., 1993b. Micromechanics modelling for the constitutive behavior of polycrystalline shape memory alloys — II. Study of the individual phenomena. *J. Mech. Phys. Solids* 41 (1), 19–33.
- Sun, Q. P., Hwang, K. C., Yu, S. W., 1991. A micromechanics constitutive model of transformation plasticity with shear and dilatation effect. *J. Mech. Phys. Solids* 9 (4), 507–524.
- Tanaka, K., 1986. A thermomechanical sketch of shape memory effect: One-dimensional tensile behavior. *Res Mechanica* 18, 251–263.
- Tanaka, K., Hayashi, T., Itoh, Y., Tobushi, H., 1992. Analysis of thermomechanical behavior of shape memory alloys. *Mech. Mater.* 13, 207–215.
- Tanaka, K., Kobayashi, S., Sato, Y., 1986. Thermomechanics of transformation pseudoelasticity and shape memory effect in alloys. *Int. J. Plasticity* 2, 59–72.
- Tanaka, K., Nishimura, F., Hayashi, T., Tobushi, H., LExcellent, C., 1995. Phenomenological analysis on subloops and cyclic behavior in shape memory alloys under mechanical and/or thermal loads. *Mech. Mater.* 19, 281–292.
- Taylor, G., 1958. The plastic wave in a wire extended by an impact load. In: Bachelor, G. (Ed.), *The Scientific Papers of G.I. Taylor, Vol I. Mechanics of Solids*. Cambridge University Press, Cambridge.
- Thomson, P., Balas, G. J., Leo, P. H., 1995. The use of shape memory alloys for passive structural damping. *Smart Mater. Struct.* 4 (1), 36–41.
- Tong, H., Wayman, C., 1974. Characteristic temperatures and other properties of thermoelastic martensites. *Acta Metallurgica* 22, 887–896.

- Tsoi, K., Stlmans, R., Schrooten, J., 2003. Transformational behavior of constrained shape memory alloys. *Acta Materialia* 50, 3535–3544.
- Turek, S., 1999. *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*. Springer Verlag, New York.
- Von Karman, T., Duwez, P., 1950. The propagation of plastic deformation in solids. *J. Appl. Phys.* 21, 987–994.
- Wayman, C. M., 1983. Phase transformations, nondiffusive. In: Cahn, R. W., Haasen, P. (Eds.), *Physical Metallurgy*. North-Holland Physics Publishing, New York.
- Wilkins, M. L., 1964. Calculation of elastic-plastic flow. In: Alder, B., Fernback, S., Rotenberg, M. (Eds.), *Methods of Computational Physics* 3. Academic Press, New York.
- Wollants, P., De Bonte, M., Roos, J., 1979. A thermodynamic analysis of the stress-induced martensitic transformation in a single crystal. *Z. Metallkd.* 70, 113–117.
- Yiu, Y. C., Regelbrugge, M. E., 1995. Shape-memory alloy isolators for vibration suppression in space applications. In: *36th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. pp. 3390–3398.
- Zhikov, V., Kozlov, S., Oleinik, O., 1994. *Homogenization of Differential Operators and Integral Functionals*. Springer-Verlag, Berlin.
- Zienkiewicz, O., Zhu, J., 1987. A simple error estimator and an adaptive procedure for practical engineering analysis. *Int. J. Numer. Meth. Eng.* 24, 337–357.

APPENDIX A

AN ASYMPTOTIC SOLUTION OF THE FSI SYSTEM FOR AN ELASTIC CHANNEL

This appendix summarizes the derivation in (Iliev et al., 2004). Consider the steady state laminar flow of incompressible Newtonian fluid through a 2-D channel with elastic walls. The reference geometry of the channel is shown on Figure 67. Denote

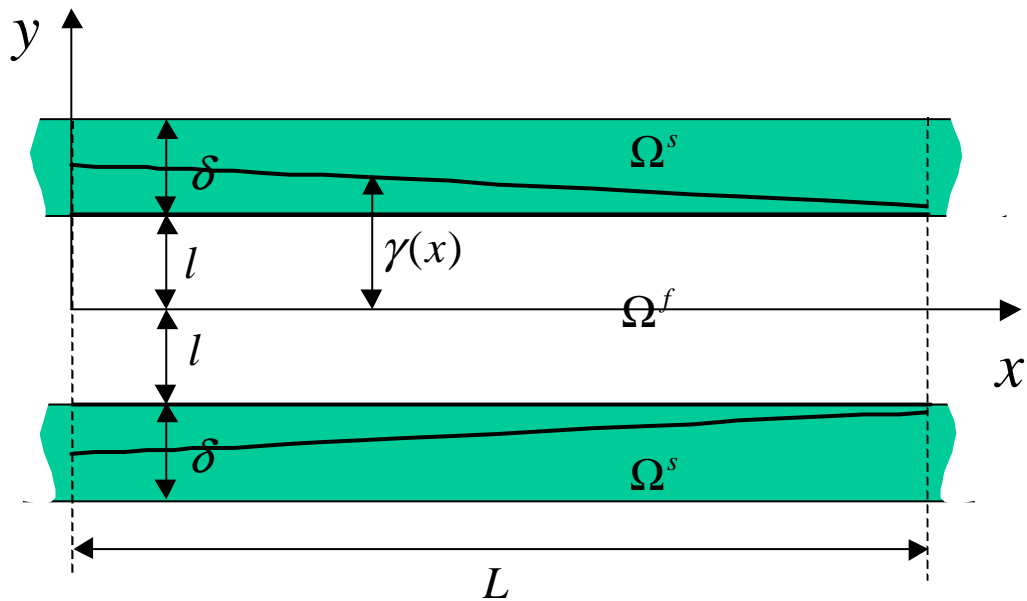


Fig. 67. Schematic of a section of length L a long elastic channel. The fluid is driven by the pressure gradient.

by L the length of the channel, by l the half of the channel width in the undeformed state, by δ the thickness of the walls in undeformed state. We assume that the channel is long, compared to its height, that is, the parameter ε :

$$\varepsilon = \frac{l}{L}. \quad (\text{A.1})$$

is small. Further we assume that $l \sim \delta$. The fluid is driven by a pressure gradient, and the outer wall of the channel are fixed. We will first normalize the FSI system (5.22)-(5.24) for this particular geometry and then we will use a formal expansion of the field variables (pressure, velocity and displacements) with respect to ϵ in order to obtain an asymptotic solution of (5.22)-(5.24).

Due to symmetry, we can only consider half of the fluid domain Ω^f and in the undeformed configuration they the fluid and solid occupy

$$\Omega_0^f = \{(x, y), 0 < x < L, 0 < y < l\}, \quad (\text{A.2})$$

$$\Omega_0^s = \{(x, y), 0 < x < L, l < y < l + \delta\}, \quad (\text{A.3})$$

respectively. Let us denote the height of the unknown boundary of the fluid-solid interface by $\gamma(x)$.

1. Dimensionless form

Consider the dimensionless variables:

$$x = L\tilde{x}, \quad y = l\tilde{y}, \quad p(x, y) = \bar{P}\tilde{p}(\tilde{x}, \tilde{y}), \quad (\text{A.4})$$

$$v_1(x, y) = \bar{V}_1\tilde{v}_1(\tilde{x}, \tilde{y}), \quad v_2(x, y) = \bar{V}_2\tilde{v}_2(\tilde{x}, \tilde{y}), \quad (\text{A.5})$$

$$u_1(x, y) = \bar{U}_1\tilde{u}_1(\tilde{x}, \tilde{y}), \quad u_2(x, y) = \bar{U}_2\tilde{u}_2(\tilde{x}, \tilde{y}). \quad (\text{A.6})$$

In these notations the fluid and solid domain are given by

$$\Omega_f = \{(\tilde{x}, \tilde{y}) : 0 < \tilde{x} < 1, \quad 0 < \tilde{y} < \tilde{\gamma}(\tilde{x})\},$$

$$\Omega_s = \left\{ (\tilde{x}, \tilde{y}) : 0 < \tilde{x} < 1, \quad \tilde{\gamma}(\tilde{x}) < \tilde{y} < 1 + \frac{\delta}{l} \right\},$$

where

$$\tilde{\gamma}(\tilde{x}) = \frac{\gamma(x)}{l} \quad (\text{A.7})$$

and, we also have ($i = 1, 2$):

$$\frac{\partial p}{\partial x} = \frac{\bar{P}}{L} \frac{\partial \tilde{p}}{\partial \tilde{x}}, \quad \frac{\partial p}{\partial y} = \frac{\bar{P}}{l} \frac{\partial \tilde{p}}{\partial \tilde{x}}, \quad (\text{A.8})$$

$$\frac{\partial v_i}{\partial x} = \frac{\bar{V}_i}{L} \frac{\partial \tilde{v}_i}{\partial \tilde{x}}, \quad \frac{\partial v_i}{\partial y} = \frac{\bar{V}_i}{l} \frac{\partial \tilde{v}_i}{\partial \tilde{x}}, \quad (\text{A.9})$$

$$\frac{\partial u_i}{\partial x} = \frac{\bar{U}_i}{L} \frac{\partial \tilde{u}_i}{\partial \tilde{x}}, \quad \frac{\partial u_i}{\partial y} = \frac{\bar{U}_i}{l} \frac{\partial \tilde{u}_i}{\partial \tilde{x}}. \quad (\text{A.10})$$

It is clear that the scaling parameters can not be chosen independently. Below we will discuss the relations between different scaling parameters.

1.1. Dimensionless Stokes equations

We first rewrite the Stokes system with respect to dimensionless variables. With the help of equations (A.8) and (A.9) we get:

$$-\varepsilon^2 \frac{\partial^2 \tilde{v}_1}{\partial \tilde{x}^2} - \frac{\partial^2 \tilde{v}_1}{\partial \tilde{y}^2} + \varepsilon^2 \frac{\bar{P}L}{\mu \bar{V}_1} \frac{\partial \tilde{p}}{\partial \tilde{x}} = 0 \quad (\text{A.11})$$

$$-\left(\varepsilon^2 \frac{\partial^2 \tilde{v}_2}{\partial \tilde{x}^2} + \frac{\partial^2 \tilde{v}_2}{\partial \tilde{y}^2} \right) + \varepsilon \frac{\bar{P}L}{\mu \bar{V}_2} \frac{\partial \tilde{p}}{\partial \tilde{y}} = 0 \quad (\text{A.12})$$

$$\frac{\partial \tilde{v}_1}{\partial \tilde{x}} + \frac{\bar{V}_2}{\varepsilon \bar{V}_1} \frac{\partial \tilde{v}_2}{\partial \tilde{y}} = 0. \quad (\text{A.13})$$

In general, velocity components and pressure can not be scaled independently. We choose \bar{V}_1 in accordance with maximal velocity of Poiseuille flow in a rigid channel:

$$\bar{V}_1 = 4V_{1,max}l^2. \quad (\text{A.14})$$

The other two scales, \bar{V}_2 and \bar{P} we express by \bar{V}_1 . In our case, the x - derivative of the pressure is the driving force for the flow, therefore we want to keep it of unit size.

Thus, without loss of generality,

$$\frac{\bar{P}L}{\mu \bar{V}_1} \varepsilon^2 = 1,$$

that is,

$$\bar{P} = \frac{\mu \bar{V}_1}{L \varepsilon^2} = \frac{\mu 4V_{1,max} l^2 L}{l^2} = 4\mu V_{1,max} L. \quad (\text{A.15})$$

Further, we want to have strong conservation of mass. Therefore we require

$$\frac{\bar{V}_2}{\varepsilon \bar{V}_1} = 1,$$

which gives us

$$\bar{V}_2 = \varepsilon \bar{V}_1 \quad (\text{A.16})$$

The Stokes system now can be rewritten as

$$\begin{aligned} -\varepsilon^2 \frac{\partial^2 \tilde{v}_1}{\partial \tilde{x}^2} - \frac{\partial^2 \tilde{v}_1}{\partial \tilde{y}^2} + \frac{\partial \tilde{p}}{\partial \tilde{x}} &= 0 \\ -\varepsilon^2 \frac{\partial^2 \tilde{v}_2}{\partial \tilde{x}^2} - \frac{\partial^2 \tilde{v}_2}{\partial \tilde{y}^2} + \varepsilon^{-2} \frac{\partial \tilde{p}}{\partial \tilde{y}} &= 0 \\ \frac{\partial \tilde{v}_1}{\partial \tilde{x}} + \frac{\partial \tilde{v}_2}{\partial \tilde{y}} &= 0. \end{aligned} \quad (\text{A.17})$$

Here we have used the fact that under the above assumptions we have

$$\frac{\bar{P} L}{\mu \bar{V}_2} \varepsilon \frac{\partial \tilde{p}}{\partial \tilde{y}} = \frac{\mu \bar{V}_1}{\varepsilon^2 L} \frac{L}{\mu \bar{V}_2} \varepsilon \frac{\partial \tilde{p}}{\partial \tilde{y}} = \frac{\bar{V}_1}{\varepsilon^2 \varepsilon \bar{V}_1} \varepsilon \frac{\partial \tilde{p}}{\partial \tilde{y}} = \varepsilon^{-2} \frac{\partial \tilde{p}}{\partial \tilde{y}}$$

Finally, we also need to express the stress tensor \mathbf{T}^f given by (5.15) in the fluid domain in terms of the non-dimensional variables:

$$\mathbf{T}^f = \frac{\mu \bar{V}_1 L}{l^2} \begin{bmatrix} 2\varepsilon^2 \frac{\partial \tilde{v}_1}{\partial \tilde{x}} - \tilde{p} & \varepsilon \left(\frac{\partial \tilde{v}_1}{\partial \tilde{y}} + \varepsilon^2 \frac{\partial \tilde{v}_2}{\partial \tilde{x}} \right) \\ \varepsilon \left(\frac{\partial \tilde{v}_1}{\partial \tilde{y}} + \varepsilon^3 \frac{\partial \tilde{v}_2}{\partial \tilde{x}} \right) & 2\varepsilon^2 \frac{\partial \tilde{v}_2}{\partial \tilde{y}} - \tilde{p} \end{bmatrix}$$

1.2. Dimensionless elasticity problem

Let us now consider the elastic domain. With the help of (A.10), the Piola-Kirchoff stress tensor \mathbf{S}^s for a linear isotropic material (5.13), can be expressed in terms of the

non-dimensional variables (A.6) as

$$\mathbf{S}^s = \begin{bmatrix} (\lambda_s + 2\mu_s) \frac{\bar{U}_1}{L} \frac{\partial \tilde{u}_1}{\partial \tilde{x}} + \lambda_s \frac{\bar{U}_2}{l} \frac{\partial \tilde{u}_2}{\partial \tilde{y}} & \mu_s \left(\frac{\bar{U}_1}{l} \frac{\partial \tilde{u}_1}{\partial \tilde{y}} + \frac{\bar{U}_2}{L} \frac{\partial \tilde{u}_2}{\partial \tilde{x}} \right) \\ \mu_s \left(\frac{\bar{U}_1}{l} \frac{\partial \tilde{u}_1}{\partial \tilde{y}} + \frac{\bar{U}_2}{L} \frac{\partial \tilde{u}_2}{\partial \tilde{x}} \right) & (\lambda_s + 2\mu_s) \frac{\bar{U}_2}{l} \frac{\partial \tilde{u}_2}{\partial \tilde{y}} + \lambda_s \frac{\bar{U}_1}{L} \frac{\partial \tilde{u}_1}{\partial \tilde{x}} \end{bmatrix}$$

Set

$$\bar{U}_2 = \delta, \quad \bar{U}_1 = \varepsilon^0 \bar{U}_2 = \delta.$$

Using this scaling for the displacements, the stress in the solid become

$$\mathbf{S}^s = \frac{\delta}{l} \begin{bmatrix} (\lambda_s + 2\mu_s) \varepsilon \frac{\partial \tilde{u}_1}{\partial \tilde{x}} + \lambda_s \frac{\partial \tilde{u}_2}{\partial \tilde{y}} & \mu_s \frac{\partial \tilde{u}_1}{\partial \tilde{y}} + \mu_s \varepsilon \frac{\partial \tilde{u}_2}{\partial \tilde{x}} \\ \mu_s \frac{\partial \tilde{u}_1}{\partial \tilde{y}} + \mu_s \varepsilon \frac{\partial \tilde{u}_2}{\partial \tilde{x}} & (\lambda_s + 2\mu_s) \frac{\partial \tilde{u}_2}{\partial \tilde{y}} + \lambda_s \varepsilon \frac{\partial \tilde{u}_1}{\partial \tilde{x}} \end{bmatrix}. \quad (\text{A.18})$$

Further, it is also necessary to write the system of elasticity equations (5.12) in non-dimensional form. For an isotropic solid, and in the absence of a body force, it is easy to see, the equation (5.12) reduces to

$$\varepsilon^2 \delta (\lambda_s + 2\mu_s) \frac{\partial^2 \tilde{u}_1}{\partial \tilde{x}^2} + \varepsilon \delta (\lambda_s + \mu_s) \frac{\partial^2 \tilde{u}_2}{\partial \tilde{x} \partial \tilde{y}} + \delta \mu_s \frac{\partial^2 \tilde{u}_1}{\partial \tilde{y}^2} = 0, \quad (\text{A.19})$$

$$\varepsilon^2 \delta \mu_s \frac{\partial^2 \tilde{u}_2}{\partial \tilde{x}^2} + \varepsilon \delta (\mu_s + \lambda_s) \frac{\partial^2 \tilde{u}_1}{\partial \tilde{x} \partial \tilde{y}} + \delta (\lambda_s + 2\mu_s) \frac{\partial^2 \tilde{u}_2}{\partial \tilde{y}^2} = 0. \quad (\text{A.20})$$

2. Asymptotic expansion

Consider now, an asymptotic expansions of the field variables with respect to the small parameter ε :

$$\tilde{v}_i = v_i^0 + \varepsilon \tilde{v}_i^1 + \varepsilon^2 \tilde{v}_i^2 + \dots \quad (\text{A.21})$$

$$\tilde{p} = \tilde{p}^0 + \varepsilon \tilde{p}^1 + \varepsilon^2 \tilde{p}^2 + \dots$$

$$\tilde{u}_i = \tilde{u}_i^0 + \varepsilon \tilde{u}_i^1 + \varepsilon^2 \tilde{u}_i^2 + \dots \quad (\text{A.22})$$

2.1. Asymptotic expansion for the Stokes system

Substituting these expansions in Stokes system (A.17), and collecting terms corresponding to different powers of ε , we get:

$$\varepsilon^{-2} : \quad \frac{\partial \tilde{p}^0}{\partial \tilde{y}} = 0. \quad (\text{A.23})$$

$$\varepsilon^{-1} : \quad \frac{\partial \tilde{p}^1}{\partial \tilde{y}} = 0. \quad (\text{A.24})$$

$$\varepsilon^0 : \quad \begin{aligned} -\frac{\partial^2 \tilde{v}_1^0}{\partial \tilde{y}^2} + \frac{\partial^2 \tilde{p}^0}{\partial \tilde{x}^2} &= 0 \\ -\frac{\partial^2 \tilde{v}_2^0}{\partial \tilde{y}^2} + \frac{\partial^2 \tilde{p}^2}{\partial \tilde{y}^2} &= 0 \\ \frac{\partial \tilde{v}_1^0}{\partial \tilde{x}} + \frac{\partial \tilde{v}_2^0}{\partial \tilde{y}} &= 0. \end{aligned} \quad (\text{A.25})$$

From (A.23) and (A.24) we have

$$\tilde{p}^0 = \tilde{p}^0(x), \quad \tilde{p}^1 = \tilde{p}^1(x). \quad (\text{A.26})$$

Further, we integrate the first equation from (A.25) with respect to \tilde{y} . The integration limits are from 0 (symmetry line) to \tilde{y} . Using the symmetry condition we get

$$-\frac{\partial \tilde{v}_1^0}{\partial \tilde{y}} = -\tilde{y} \frac{\partial \tilde{p}^0}{\partial \tilde{x}}$$

Integrating the last equation with respect to \tilde{y} from \tilde{y} to $\gamma(L\tilde{x})$, we get

$$-\tilde{v}_1^0(\tilde{x}, \gamma(L\tilde{x})) + v_1^0(\tilde{x}, \tilde{y}) = -\frac{1}{2} (\gamma^2(L\tilde{x}) - \tilde{y}^2) \frac{\partial \tilde{p}^0}{\partial \tilde{x}} \quad (\text{A.27})$$

Using the no-slip boundary condition $\tilde{v}_1^0(\tilde{x}, \gamma(L\tilde{x})) = 0$, we obtain:

$$\tilde{v}_1^0(\tilde{x}, \tilde{y}) = -\frac{\gamma^2(L\tilde{x}) - \tilde{y}^2}{2} \frac{\partial \tilde{p}^0}{\partial \tilde{x}} \quad (\text{A.28})$$

We will need further the \tilde{x} - derivative of \tilde{v}_1^0 :

$$\frac{\partial \tilde{v}_1^0}{\partial \tilde{x}} = -\frac{\gamma^2(L\tilde{x}) - \tilde{y}^2}{2} \frac{\partial^2 \tilde{p}^0}{\partial \tilde{x}^2} - L\gamma(L\tilde{x}) \frac{\partial \gamma(L\tilde{x})}{\partial \tilde{x}} \frac{\partial \tilde{p}^0}{\partial \tilde{x}} \quad (\text{A.29})$$

Integrating the continuity equation from 0 to γ and using the boundary conditions $v_1^0(\tilde{x}, 0) = v_1^0(\gamma) = 0$, we get

$$\begin{aligned} 0 = -\int_0^\gamma \frac{\partial \tilde{v}_1^0(\tilde{x}, s)}{\partial \tilde{x}} ds &= -\frac{1}{2} \frac{\partial^2 \tilde{p}^0}{\partial \tilde{x}^2} \left(\gamma^3 - \frac{\gamma^3}{3} \right) - L\gamma^2(L\tilde{x}) \frac{\partial \gamma(L\tilde{x})}{\partial \tilde{x}} \frac{\partial \tilde{p}^0}{\partial \tilde{x}} = \\ &= -\frac{\partial^2 \tilde{p}^0}{\partial \tilde{x}^2} \frac{\gamma^3}{3} - \frac{1}{3} \frac{\partial \gamma^3(L\tilde{x})}{\partial \tilde{x}} \frac{\partial \tilde{p}^0}{\partial \tilde{x}} = -\frac{1}{3} \frac{\partial}{\partial \tilde{x}} \left(\gamma^3(L\tilde{x}) \frac{\partial \tilde{p}^0}{\partial \tilde{x}} \right). \end{aligned}$$

That is, we have obtained an equation with respect to $p^0(x)$:

$$\frac{\partial}{\partial \tilde{x}} \left(\gamma^3(L\tilde{x}) \frac{\partial \tilde{p}^0}{\partial \tilde{x}} \right) = 0. \quad (\text{A.30})$$

2.2. Expansion for the solid-fluid interface

So, we obtained equations for v_1^0 , v_2^0 , p^0 , which depend on the unknown free boundary $\gamma(L\tilde{x})$. To get an equation for the free boundary we will use the interface condition for the continuity of the normal component of the stress tensor. So, we need to calculate this normal component. Substituting expansions for velocity and pressure from (A.21), we obtain

$$\mathbf{T}^f = \frac{\mu \bar{V}_1 L}{l^2} \begin{bmatrix} 2\varepsilon^2 \frac{\partial \tilde{v}_1^0}{\partial \tilde{x}} - p^0 - \varepsilon p^1 - \varepsilon p^2 + O(\varepsilon^3) & \varepsilon \left(\frac{\partial \tilde{v}_1^0}{\partial \tilde{y}} + \varepsilon \frac{\partial \tilde{v}_1^0}{\partial \tilde{y}} + \varepsilon^2 \frac{\partial \tilde{v}_2^0}{\partial \tilde{y}} \right) + O(\varepsilon^3) \\ \varepsilon \left(\frac{\partial \tilde{v}_1^0}{\partial \tilde{y}} + \varepsilon \frac{\partial \tilde{v}_1^0}{\partial \tilde{y}} + \varepsilon^2 \frac{\partial \tilde{v}_2^0}{\partial \tilde{y}} \right) + O(\varepsilon^3) & 2\varepsilon^2 \frac{\partial \tilde{v}_2^0}{\partial \tilde{x}} - p^0 - \varepsilon p^1 - \varepsilon p^2 + O(\varepsilon^3) \end{bmatrix}$$

From this we obtain the zeroth order approximation:

ε^0 :

$$\sigma^{F,0} = \frac{\mu \bar{V}_1 L}{l^2} \begin{bmatrix} -\tilde{p}^0 & 0 \\ 0 & -\tilde{p}^0 \end{bmatrix} \quad (\text{A.31})$$

The normal vector to the curve $\gamma(x)$ is given by

$$\mathbf{n} = \frac{\mathbf{e}_2 - lL\gamma'(L\tilde{x})\mathbf{e}_1}{C_\sigma}, \quad \text{where } C_\sigma = \frac{1}{\sqrt{1 + L^2\gamma'(L\tilde{x})l^2}}. \quad (\text{A.32})$$

Using this, we calculate the normal component of the zeroth order term for the stress tensor:

$$\mathbf{T}^{f,0} \mathbf{n} = \frac{1}{C_\sigma} \frac{\mu \bar{V}_1 L}{l^2} \{ (\tilde{p}^0 l L \gamma'(L\tilde{x})) \mathbf{e}_1 + -\tilde{p}^0 \mathbf{e}_2 \}. \quad (\text{A.33})$$

2.3. Asymptotic expansion for elasticity system

Now we substitute the asymptotic expansion for u_1 , u_2 in the elasticity system (A.19),(A.20). From the first elasticity equation (A.19), at order ε^0 , we obtain

$$\mu_s \frac{\partial^2 \tilde{u}_1^0}{\partial \tilde{y}^2} = 0.$$

Integrating with respect to \tilde{y} we get

$$\mu_s \frac{\partial \tilde{u}_1^0}{\partial \tilde{y}} = c_1(\tilde{x}). \quad (\text{A.34})$$

Integrating from \tilde{y} to $(1 + \frac{\delta}{l})$, and using the fact that $\tilde{u}_1^0 = 0$ at the upper boundary, we get

$$\tilde{u}_1^0(\tilde{x}, \tilde{y}) = -\frac{(1 + \frac{\delta}{l} - \tilde{y})}{\mu_s} c_1(\tilde{x}) \quad (\text{A.35})$$

The second equation (A.20) from the elasticity system gives, at order ε^0 ,

$$(\lambda_s + 2\mu_s) \frac{\partial \tilde{u}_2^0}{\partial \tilde{y}} = 0.$$

Integrating with respect to \tilde{y} we get

$$\frac{\partial \tilde{u}_2^0}{\partial \tilde{y}} = \frac{1}{\lambda_s + 2\mu_s} c_2(\tilde{x}) \quad (\text{A.36})$$

Integrating this equation from \tilde{y} to $(1 + \frac{\delta}{l})$, and using the fact that $\tilde{u}_2^0 = 0$ at the upper boundary, we get

$$\tilde{u}_2^0(\tilde{x}, \tilde{y}) = -\frac{(1 + \frac{\delta}{l} - \tilde{y})}{\lambda_s + 2\mu_s} c_2(\tilde{x}) \quad (\text{A.37})$$

Next, we can substitute the asymptotic expansion (A.22) for \tilde{u}_1 , \tilde{u}_2 into (A.18) and obtain the stresses in the solid at order ε^0 :

$$\mathbf{S}^{s,0} = \frac{\delta}{l} \begin{bmatrix} \lambda_s \frac{\partial \tilde{u}_2^0}{\partial \tilde{y}} & \mu_s \frac{\partial \tilde{u}_1^0}{\partial \tilde{y}} \\ \mu_s \frac{\partial \tilde{u}_1^0}{\partial \tilde{y}} & (\lambda_s + 2\mu_s) \frac{\partial \tilde{u}_2^0}{\partial \tilde{y}} \end{bmatrix}$$

Finally, we can substitute \tilde{u}_1^0 , \tilde{u}_2^0 as given by (A.35) and (A.37) respectively into the last equation and obtain:

$$\mathbf{S}^{s,0} = \frac{\delta}{l} \begin{bmatrix} \frac{\lambda_s}{\lambda_s + 2\mu_s} c_2(\tilde{x}) & c_1(\tilde{x}) \\ c_1(\tilde{x}) & c_2(\tilde{x}) \end{bmatrix}.$$

Using this, we calculate the normal component of the zeroth order term for the stress tensor:

$$\mathbf{S}^{s,0} \mathbf{n} = \frac{\delta}{l C_\sigma} \left\{ \left(-\frac{\lambda_s}{\lambda_s + 2\mu_s} c_2(\tilde{x}) l L \gamma'(L\tilde{x}) + c_1(\tilde{x}) \right) \mathbf{e}_1 + \left(-c_1(\tilde{x}) l L \gamma'(L\tilde{x}) + c_2(\tilde{x}) \right) \mathbf{e}_2 \right\}. \quad (\text{A.38})$$

Now using the interface condition for continuity of the normal component of the tensors we get two equations with $c_1(\tilde{x})$ and $c_2(\tilde{x})$ as unknowns:

$$\begin{aligned} \frac{\delta}{lC_\sigma} \left(c_1(\tilde{x}) - \frac{\lambda_s}{\lambda_s + 2\mu_s} c_2(\tilde{x}) lL\gamma'(L\tilde{x}) \right) &= -\frac{\mu\bar{V}_1 L}{l^2 C_\sigma} \tilde{p}^0(\tilde{x}) lL\gamma'(L\tilde{x}), \\ -\frac{\delta}{lC_\sigma} (c_1(\tilde{x}) lL\gamma'(L\tilde{x}) - c_2(\tilde{x})) &= \frac{\mu\bar{V}_1 L}{l^2 C_\sigma} \tilde{p}^0(\tilde{x}) \end{aligned}$$

Denoting $Q_1 = \frac{\mu\bar{V}_1 L}{l^2} = \bar{P}$, $Q_2 = lL\gamma'(L\tilde{x})$ and rearranging terms we obtain:

$$\begin{aligned} c_1(\tilde{x}) - \frac{\lambda_s}{\lambda_s + 2\mu_s} c_2(\tilde{x}) Q_2 &= Q_1 Q_2 \tilde{p}^0(\tilde{x}), \\ -Q_2 c_1(\tilde{x}) + c_2(\tilde{x}) &= -Q_1 \tilde{p}^0(\tilde{x}) \end{aligned}$$

From here we obtain

$$c_1(\tilde{x}) = \frac{Q_2 \left(-1 + \frac{\lambda_s}{\lambda_s + 2\mu_s} \right) Q_1}{1 - \frac{\lambda_s}{\lambda_s + 2\mu_s} Q_2^2} \tilde{p}^0(\tilde{x}) \quad (\text{A.39})$$

$$c_2(\tilde{x}) = \frac{Q_1 - Q_1 Q_2^2}{1 - \frac{\lambda_s}{\lambda_s + 2\mu_s} Q_2^2} \tilde{p}^0(\tilde{x}) \quad (\text{A.40})$$

Further, we use the fact that in our case $\gamma' = O(\varepsilon)$ and neglect terms with Q_2^2 .

Substituting in (A.35) and (A.37) we get

$$\tilde{u}_1^0(\tilde{x}, \tilde{y}) = -\frac{1 + \frac{\delta}{l} - \tilde{y}}{\mu_s} Q_2 \left(-1 + \frac{\lambda_s}{\lambda_s + 2\mu_s} \right) Q_1 \tilde{p}^0(\tilde{x}) \quad (\text{A.41})$$

$$\tilde{u}_2^0(\tilde{x}, \tilde{y}) = -\frac{1 + \frac{\delta}{l} - \tilde{y}}{\lambda_s + 2\mu_s} Q_1 \tilde{p}^0(\tilde{x}) \quad (\text{A.42})$$

Now recall that

$$\tilde{\gamma}(\tilde{x}) = \frac{\gamma(L\tilde{x})}{l} = 1 + \tilde{u}_2(\tilde{x}, 1),$$

which, together with equation (A.42), implies

$$\tilde{\gamma}(x) = 1 + \frac{\delta}{l} \frac{1}{\lambda_s + 2\mu_s} \bar{P} \tilde{p}^0(\tilde{x}). \quad (\text{A.43})$$

Equivalently, in dimensional variables,

$$\gamma(x) = l + \frac{\delta}{\lambda_s + 2\mu_s} p^0(\tilde{x}). \quad (\text{A.44})$$

Now we can return to the expression for the x - component of the velocity vector (A.28). For fixed \tilde{x} (equivalently, x) and a generic function $\tilde{\phi}(\tilde{x}, \tilde{y})$ ($= \phi(x, y)$), let us introduce the y -average operator $\langle \cdot \rangle$:

$$\langle \tilde{\phi}(\tilde{x}, \tilde{y}) \rangle := \frac{1}{2\tilde{\gamma}(\tilde{x})} \int_{-\tilde{\gamma}(\tilde{x})}^{\tilde{\gamma}(\tilde{x})} \tilde{\phi}(\tilde{x}, \tilde{y}) dy = \frac{1}{2\gamma(x)} \int_{-\gamma(x)}^{\gamma(x)} \phi(x, y) dy =: \langle \phi(x, y) \rangle. \quad (\text{A.45})$$

The factor of 2 in the denominator is because we have symmetry of the all field variables with respect of the x - axis and the interval 0 to γ is half of the channel. By averaging equation (A.28) we get:

$$\langle \tilde{v}_1(\tilde{x}) \rangle = -\frac{1}{3} \gamma^3(L\tilde{x}) \left\langle \frac{\partial \tilde{p}^0}{\partial \tilde{x}} \right\rangle$$

Substituting here the expression for γ from (A.44), we obtain

$$\langle \tilde{v}_1(\tilde{x}) \rangle = -\frac{1}{3} \left(l + \frac{\delta}{\lambda_s + 2\mu_s} \bar{P} \tilde{p}^0(\tilde{x}) \right)^3 \left\langle \frac{\partial \tilde{p}^0(\tilde{x})}{\partial \tilde{x}} \right\rangle.$$

Now we can consider the ratio of the mass flux and the pressure gradient which, in the rigid case, gives the permeability K :

$$K := K(\tilde{p}^0(\tilde{x}), \tilde{x}) = -\frac{\langle \tilde{v}_1(\tilde{x}) \rangle}{\left\langle \frac{\partial \tilde{p}^0(\tilde{x})}{\partial \tilde{x}} \right\rangle} = \frac{1}{3} \left(l + \frac{\delta}{\lambda_s + 2\mu_s} \bar{P} \tilde{p}^0(\tilde{x}) \right)^3 \quad (\text{A.46})$$

Observe, that for a deformable channel, permeability is not a constant, even for fixed fluid and solid materials. We can view it as a function of axial position and averaged pressure. Rewriting the last equation (A.46) by returning back to dimensional variables, we get:

$$K(p^0(x), x) = \frac{1}{3} \left(l + \frac{\delta}{\lambda_s + 2\mu_s} p^0(\mathbf{x}) \right)^3. \quad (\text{A.47})$$

APPENDIX B

INTEGRATION OF A RETURN MAPPING ALGORITHM IN A
DISPLACEMENT BASED FINITE ELEMENT METHOD

This appendix is intended as a brief summary of a displacement based FEM for a problem with material nonlinearity.

The notation follows that from Chapter V. Since there is not a chance of confusion the reference configuration of the solid is denoted by Ω instead of Ω_0^* . The loading of the body is assumed to have taken place at discrete instances $1, 2, \dots, n$ with some given boundary conditions $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{s}}_i$, $i = 1, \dots, n$, and the material state has already been determined at this instances. For the current loading step, $\hat{\mathbf{u}}_{n+1}$ and $\hat{\mathbf{s}}_{n+1}$ are given and the field variables $\boldsymbol{\varepsilon}_{n+1}, \mathbf{S}_{n+1}, \dots$ have to be computed. For simplicity, the superscript $n + 1$ will be dropped in the following discussion.

The nonlinear analog to the weak form (5.25),(5.26) reads:

$$(\mathbf{S}(\mathbf{e}(\mathbf{u})), \mathbf{e}(\mathbf{w}))_{\Omega} = (\mathbf{b}, \mathbf{w})_{\Omega} + (\hat{\mathbf{s}}, \mathbf{w})_{\Gamma^N}, \quad \forall \mathbf{w} \in [H_D^1(\Omega)]^d, \quad (\text{B.1})$$

$$\mathbf{u} = \hat{\mathbf{u}}, \quad \text{on } \Gamma^D. \quad (\text{B.2})$$

For convenience and without loss of generality, the path dependence of the stress is also omitted. A more precise notation for \mathbf{S} , which is also much more cumbersome, is

$$\mathbf{S} = \mathbf{S}(\mathbf{e}(\mathbf{u}); \boldsymbol{\varepsilon}^{in}, \boldsymbol{\xi}, \dot{\boldsymbol{\xi}}), \quad (\text{B.3})$$

see the remarks after equation (2.28).

Upon introducing a discrete space for the displacement $U_{\mathbf{u}}$, the discrete weak

form is obtained directly from (B.1),(B.2):

$$(\mathbf{S}(\mathbf{e}(\mathbf{u}^h)), \mathbf{e}(\mathbf{w}^h))_{\Omega} = (\mathbf{b}, \mathbf{w}^h)_{\Omega} + (\hat{\mathbf{s}}, \mathbf{w}^h)_{\Gamma^N}, \quad \forall \mathbf{w}^h \in U_{\mathbf{u}}, \quad (\text{B.4})$$

$$\mathbf{u}^h = \hat{\mathbf{u}}, \quad \text{on } \Gamma^D. \quad (\text{B.5})$$

where \mathbf{u}^h is the discrete displacement. Let the number of degrees of freedom be N and the nodal basis functions be denoted by $\psi^{(i)}$, $i = 1, \dots, N$. Also, for convenience, let the column vector of all the basis functions be denoted by $\Psi = (\psi^{(1)}, \dots, \psi^{(N)})$.

In this notation,

$$\mathbf{u}^h = \sum_{i=1}^N \psi^{(i)}(\mathbf{x}) U_i = \mathbf{U} \cdot \Psi,$$

where $\mathbf{U} = (U_1, \dots, U_N)^T$ is the column vector of the nodal values of the displacements.

To solve the nonlinear equation (B.1) Newton's method is applied to the residual function

$$\mathbf{F}^h(\mathbf{U}) = (\mathbf{S}(\mathbf{e}(\mathbf{U})), \mathbf{w})_{\Omega} - (\mathbf{b}, \mathbf{w})_{\Omega} - (\hat{\mathbf{s}}, \mathbf{w})_{\Gamma^N}, \quad (\text{B.6})$$

that is, an iterative solution is sought:

$$\mathbf{U} = \lim_{k \rightarrow \infty} \mathbf{U}^{(k)},$$

where the $k + 1$ iterate is defined by the recursive formula

$$\mathbf{F}^h(\mathbf{U}^{(k)}) + \frac{\partial \mathbf{F}^h(\mathbf{U}^{(k)})}{\partial \mathbf{U}} (\mathbf{U}^{(k+1)} - \mathbf{U}^{(k)}) = \mathbf{0}. \quad (\text{B.7})$$

Denoting by \mathbf{L} the Jacobian in the above equation:

$$\mathbf{L}(\mathbf{U}) := \frac{\partial \mathbf{F}^h(\mathbf{U})}{\partial \mathbf{U}}, \quad (\text{B.8})$$

the recursive relation (B.7) can be written as

$$\mathbf{U}^{(k+1)} = \mathbf{U}^{(k)} + \mathbf{L}^{-1}(\mathbf{U}^{(k)}) \mathbf{F}^h(\mathbf{U}^{(k)}). \quad (\text{B.9})$$

Given some displacements $\mathbf{U}^{(k)}$, the stress $\mathbf{S}(\mathbf{e}(\mathbf{u}^h))$ can be computed at an arbitrary location \mathbf{x} by evaluating the discrete strain at that point $\mathbf{e}(\mathbf{u}^h(\mathbf{x}))$ and then using the return mapping algorithm of Chapter IV, Section 1 to get all the state variables at this point. Evaluating the residual function is done by numerical integration of the integrals in (B.4), therefore the stress \mathbf{S} is required only at discrete integration points (cf., e.g., Ciarlet, 2002; Reddy, 1993).

Evaluating the Jacobian matrix \mathbf{L} requires some manipulations. Observe that:

$$\begin{aligned} \frac{\partial \mathbf{F}^h(\mathbf{U})}{\partial \mathbf{U}} &= \int_{\Omega} \frac{\partial \mathbf{S}(\mathbf{e}(\mathbf{u}^h))}{\partial \mathbf{U}} : \mathbf{e}(\mathbf{w}^h) d\mathbf{p} \\ &= \int_{\Omega} \left(\frac{\partial \mathbf{S}(\mathbf{e}(\mathbf{u}^h))}{\partial \mathbf{e}(\mathbf{u}^h)} : \frac{\partial \mathbf{e}(\mathbf{u}^h)}{\partial \mathbf{U}} \right) : \mathbf{e}(\mathbf{w}^h) d\mathbf{p} \\ &= \int_{\Omega} \left(\mathcal{L}(\mathbf{u}^h) : \frac{\partial \mathbf{e}(\mathbf{u}^h)}{\partial \mathbf{U}} \right) : \mathbf{e}(\mathbf{w}^h) d\mathbf{p}. \end{aligned} \quad (\text{B.10})$$

The last row in the above equation was obtained by defining $\mathcal{L}(\mathbf{u}^h)$ to be the derivative:

$$\mathcal{L}(\mathbf{u}^h) := \frac{\partial \mathbf{S}(\mathbf{e}(\mathbf{u}^h))}{\partial \mathbf{e}(\mathbf{u}^h)}. \quad (\text{B.11})$$

Now, in light of the suppressed path dependence (B.3) of the Piola-Kirchhoff stress, it is clear that $\mathcal{L}(\mathbf{u}^h)$ gives the variation of \mathbf{S} around the point $\boldsymbol{\varepsilon}^h$ and with respect to the previous material state $(\boldsymbol{\varepsilon}_n^h, \boldsymbol{\varepsilon}_n^{in,h}, \dots)$. Therefore, the last definition (B.11) is nothing else but the *algorithmic tangent stiffness* defined in (4.26). This explains why it was necessary to derive equation (4.33). The reader is referred to Simo and Hughes (1998) for further reading on this topic.

Coming back to the Jacobian matrix, a further simple calculation shows that

$$\frac{\partial \mathbf{e}(\mathbf{u}^h)}{\partial \mathbf{U}} = \frac{\partial}{\partial \mathbf{U}} (\mathbf{e}(\mathbf{U} \cdot \boldsymbol{\Psi}(\mathbf{p}))) = \frac{\partial}{\partial \mathbf{U}} (\mathbf{U} \cdot \mathbf{e}(\boldsymbol{\Psi}(\mathbf{p}))) = \mathbf{e}(\boldsymbol{\Psi}(\mathbf{p})), \quad (\text{B.12})$$

where $\mathbf{e}(\boldsymbol{\Psi}(\mathbf{p}))$ is the vector-column $\left(\mathbf{e}(\boldsymbol{\Psi}^{(1)}(\mathbf{p})), \dots, \mathbf{e}(\boldsymbol{\Psi}^{(N)}(\mathbf{p})) \right)^T$ consisting of the

strains for each degree of freedom as its components. Combining equations (B.10) and (B.12), the components of the Jacobian matrix (B.8) are found to be:

$$L_{ij} = \int_{\Omega} \mathbf{e} \left(\Psi^{(i)}(\mathbf{p}) \right) : \mathcal{L}(\mathbf{U} \cdot \Psi(\mathbf{p})) : \mathbf{e} \left(\Psi^{(j)}(\mathbf{p}) \right) d\mathbf{p}.$$

Now, the Jacobian (B.8) required in (B.7) is completely defined in terms of the selected FEM basis functions Ψ and the algorithmic tangent stiffness \mathcal{L} .

APPENDIX C

SMA SUBROUTINE SOURCE CODE LISTING

Listing VI.1 SMA numerical implementation header file

```

1  #ifndef __SMA2004_H
2  #define __SMA2004_H
3
4  #include <math.h>
5  #include "tensors.h"
6  #include "inelasticity.h"
7
8  #define PT_NONE 0
9  #define PT_SIM_FORWARD 1
10 #define PT_SIM_REVERSE 2
11 #define PT_SA_FORWARD 3
12 #define PT_SA_REVERSE 4
13 #define PT_REORIENT 5
14
15 #define XI_SA 1
16 #define XI_SIM 2
17 #define XI_REORIENT 3
18
19
20 class ESMAError{
21     private:
22         char *fMsg;
23     public:
24         ESMAError(char* AMsg);
25 };
26
27 class TSMAMaterial : public TInelasticMaterial{
28     public:
29         double rho, EM, EA, nuM, nuA, aM, aA, cM, cA, As, Af, Ms, Mf, H,
30             stress_s, stress_f, T0, delta_s0, delta_c, c01, c02, c03;
31
32         int f3deg;
33         double *f3;
34
35         double Y1, Y2, Y3, b1M, b2M, b1A, b2A, b3, delta_u1, delta_u2;
36
37         int poly3deg;
38         double *poly3; // polynomial hardening for detwinning
39
40         tensor4 CM, CA, dC; // Elastic moduli
41         tensor4 SM, SA, dS; // Compliance
42         tensor AlphaM, AlphaA, dAlpha;
43
44         TSMAMaterial() {};
45         TSMAMaterial(const char * str);
46         virtual char* __cdecl Name() { return "SMA"; }
47         virtual char* __cdecl AsString();
48         virtual int __cdecl InternalVarCount() { return 6; };
49         virtual bool __cdecl ParamByName(double &res, char* name);
50
51         void Update();
52 };
53
54 class TSMAMaterialState : public TInelasticState{
55     private:
56         void init() ;
57     public:
58         tensor strain; // total strain
59         double T;
60         tensor stress; // stress
61         tensor4* dstress_dstrain; // derivatives with respect to strain. If null,
62                                     // the FEM solver cannot use Newton's method.
63
64         tensor strain_in;
65         double xi1, xi2, xi3; // internal state variables;
66         double dbl_tag;
67
68         TSMAMaterialState(){ init(); };
69         TSMAMaterialState(TSMAMaterial* amtrl) { SetMaterial(amtrl); init(); };
70         ~TSMAMaterialState(){ if (dstress_dstrain != NULL) delete dstress_dstrain; }

```

```

71
72 virtual char* __cdecl GetAsString(char *res);
73 virtual void __cdecl Assign(TInelasticState *src);
74 virtual tensor* __cdecl Strain() { return &strain; };
75 virtual tensor* __cdecl Stress() { return &stress; };
76 virtual double* __cdecl Temperature() { return &T; };
77 virtual double* __cdecl InternalVar(int i);
78 // this function, if implemented, can provide with a way of splitting the
79 // stress tensor when used in a direct iteration method.
80 virtual tensor* __cdecl InelasticStrain(tensor *res, int i);
81 virtual tensor4* __cdecl EffctvC(tensor4* res);
82 virtual tensor4* __cdecl EffctvS(tensor4* res);
83 virtual tensor* __cdecl EffctvA(tensor* res);
84 virtual tensor4* __cdecl DStressDStrain();
85
86 virtual void __cdecl SetMaterial(PInelasticMaterial amtrl);
87 virtual void __cdecl SetAsString(const char* str);
88 virtual tensor4* __cdecl DStressDStrain_Alg(tensor4 *res, TInelasticState *old_state);
89 virtual tensor4* __cdecl DStressDStrain_Alg2(tensor4 *res, TInelasticState *old_state);
90
91 tensor* __cdecl InelasticPrediction(tensor* res, int pt_indx);
92 tensor* __cdecl HookeResidual(tensor* res);
93 tensor* __cdecl dHookeResidual_dPT(tensor* res, int pt_indx);
94 tensor4* __cdecl dHookeResidual_dstress(tensor4* res, int pt_indx, TSMASState *old_state);
95 tensor4* __cdecl dHookeResidual_dstress2(tensor4* res, int pt_indx1, int pt_indx2, TSMASState *old_state);
96
97 // Generic flow rules
98 tensor* J2FlowRule(tensor *res);
99 tensor4* dJ2_dstress(tensor4Ptr res);
100 double* get_h(double *res) { return &(*res = 1.0); };
101 double* get_h2(double *res) { return get_h(res); };
102 // Flow rules for the forward and backward  $M^t \rightarrow M^d$  transformation, i.e.  $\backslash\Lambda_{\lambda_3}$ 
103 double* get_h3(double *res) { return get_h(res); };
104
105 // Transformation surfaces
106 double Phi(int pt_indx);
107 tensor* dPhi_dstress(tensor* res, int pt_indx);
108 double dPhi_dxi(int pt_indx, int xi_indx);
109 tensor* Lambda(tensor* res, int pt_indx);
110 tensor4* dLambda_dstress(tensor4 *res, int pt_indx);
111 tensor* dxi_dstress(tensor* res, int pt_indx);
112
113 // Transformation surface, flow rule and derivatives for  $A \rightarrow M^t$  transformation,
114 double Phi1_forward();
115 tensor* dPhi1_forward_dstress(tensor* res);
116 double dPhi1_forward_dxi1();
117 double dPhi1_forward_dxi2();
118 double dPhi1_forward_dxi3();
119 tensor* Lambda1_forward(tensor *res) { zero_tensor(res); return res; };
120 tensor4* dLambda1_forward_dstress(tensor4 *res) { zero_tensor4(res); return res; };
121
122 // Transformation surface, flow rule and derivatives for  $M^t \rightarrow A$  transformation,
123 double Phi1_reverse();
124 tensor* dPhi1_reverse_dstress(tensor* res);
125 double dPhi1_reverse_dxi1();
126 double dPhi1_reverse_dxi2();
127 double dPhi1_reverse_dxi3();
128 tensor* Lambda1_reverse(tensor *res) { zero_tensor(res); return res; };
129 tensor4* dLambda1_reverse_dstress(tensor4 *res) { zero_tensor4(res); return res; };
130
131 // Transformation surface, flow rule and derivatives for  $A \rightarrow M^d$  transformation,
132 double Phi2_forward();
133 tensor* dPhi2_forward_dstress(tensor* res);
134 double dPhi2_forward_dxi1();
135 double dPhi2_forward_dxi2();
136 double dPhi2_forward_dxi3();
137 tensor* Lambda2_forward(tensor *res) { return J2FlowRule(res); }; //  $\backslash\Lambda_{\lambda_2}$ 
138 tensor4* dLambda2_forward_dstress(tensor4 *res) { return dJ2_dstress(res); };
139
140 // Transformation surface, flow rule and derivatives for  $M^d \rightarrow A$  transformation,
141 double Phi2_reverse();
142 tensor* dPhi2_reverse_dstress(tensor* res);
143 double dPhi2_reverse_dxi1();
144 double dPhi2_reverse_dxi2();
145 double dPhi2_reverse_dxi3();
146 tensor* Lambda2_reverse(tensor *res);
147 tensor4* dLambda2_reverse_dstress(tensor4 *res) { zero_tensor4(res); return res; };
148
149 // Transformation surface, flow rule and derivatives for  $M^t \rightarrow M^d$  transformation,
150 double Phi3();
151 tensor* dPhi3_dstress(tensorPtr res);
152 double dPhi3_dxi1();
153 double dPhi3_dxi2();
154 double dPhi3_dxi3();
155 tensor* Lambda3(tensor *res) { return J2FlowRule(res); };
156 tensor4* dLambda3_dstress(tensor4 *res) { return dJ2_dstress(res); };
157

```

```

158 // Transformation surfaces derivatives
159 // A->M^d
160 double dPhi2_forward_q();
161 // double dPhi2_reverse_q();
162
163 // A->M^t
164 double dPhi3_q() { return dPhi3_dxi3(); };
165 // Support functions
166 void restore_phi3();
167
168 double c1(){ return ((TSMAMaterial *)fmtrl)->c01 + xi1 - xi3; };
169 double c2(){ return ((TSMAMaterial *)fmtrl)->c02 + xi2 + xi3; };
170 double c3(){ return ((TSMAMaterial *)fmtrl)->c03 - xi1 - xi2; };
171
172 double min_xi1();
173 double min_xi2();
174 double min_xi3();
175
176 double max_xi1();
177 double max_xi2();
178 double max_xi3();
179
180 bool is_A2Mt_possible();
181 bool is_A2Md_possible();
182 bool is_Md2A_possible();
183 bool is_Mt2A_possible();
184 bool is_Mt2Md_possible();
185
186 bool check_consistency(TSMASState *ostate);
187 };
188
189 typedef double (TInelasticState::*TYieldSurfaceProc)();
190 typedef tensorPtr (TInelasticState::*TStrainFlowProc)(tensor *res);
191 typedef double* (TInelasticState::*TIntVarFlowProc)(double *res);
192 typedef tensorPtr (TInelasticState::*TDPHiDStressProc)(tensor *res);
193 typedef double (TInelasticState::*TDPHiDqProc)();
194
195 class TSMASolver: public TInelasticSolver{
196 protected:
197 double falpha;
198 TInelasticState *LinearLoad(TInelasticState *res, tensor* new_strain,
199 double new_T, TInelasticState *old_state);
200
201 // Generic, one variable Closest point projection method
202 TInelasticState *SimoCPP(TSMASState *res, tensor* new_strain,
203 double new_T, TSMASState* old_state, int pt_indx);
204 // Generic, two variable Closest point projection method
205 TInelasticState *SimoCPP2(TSMASState *res, tensor* new_strain,
206 double new_T, TSMASState* old_state, int pt_indx1, int pt_indx2);
207 // Optimized
208 TInelasticState *SimoCPP_forward(TSMASState *res, tensor* new_strain,
209 double new_T, TSMASState* old_state);
210 TInelasticState *SimoCPP_reverse(TSMASState *res, tensor* new_strain,
211 double new_T, TSMASState* old_state);
212
213 TInelasticState *OrtizPopov85(TInelasticState *res, tensor* new_strain,
214 double new_T, TInelasticState* old_state,
215 TYieldSurfaceProc YieldSurface, TStrainFlowProc GetR, TIntVarFlowProc GetH,
216 TDPHiDStressProc GetDPHiDStress, TDPHiDqProc GetDPHiDq, int q_num);
217
218 TSMASState *Detwin(TSMASState *res, tensor* new_strain, double new_T, TSMASState* old_state);
219 TSMASState *SIM(TSMASState *res, tensor* new_strain, double new_T, TSMASState* old_state);
220 TSMASState *SI_reverse(TSMASState *nstate, tensor* new_strain, double new_T, TSMASState* ostate);
221 TSMASState *SelfAccom_forward(TSMASState *res, tensor* new_strain, double new_T, TSMASState* old_state);
222 TSMASState *SelfAccom_reverse(TSMASState *res, tensor* new_strain, double new_T, TSMASState* old_state);
223 TSMASState *M2A(TSMASState *nstate, tensor* new_strain, double new_T, TSMASState* ostate);
224 TSMASState *A2M(TSMASState *nstate, tensor* new_strain, double new_T, TSMASState* ostate);
225 TSMASState *Mt2Md_and_Mt2A(TSMASState *res, tensor* new_strain, double new_T, TSMASState* ostate);
226
227 bool RestoreConsistency_Md2A(TSMASState *nstate, tensor* new_strain, double new_T, TSMASState *ostate);
228 bool RestoreConsistency_M2A(TSMASState *nstate, tensor* new_strain, double new_T, TSMASState *ostate, bool
229 type);
229 void RestoreConsistency_xi3(TSMASState *res, tensor* new_strain, double new_T, TSMASState *ostate);
230 bool RestoreConsistency_A2Mt(TSMASState *nstate, tensor* new_strain, double new_T, TSMASState *ostate);
231 void RestoreConsistency_A2Md(TSMASState *res, tensor* new_strain, double new_T, TSMASState *ostate);
232 bool RestoreConsistency_A2Md_xxx(TSMASState *res, tensor* new_strain, double new_T, TSMASState *ostate);
233 void RestoreConsistency_A2M(TSMASState *nstate, tensor* new_strain, double new_T, TSMASState *ostate);
234 void RestoreConsistency_Mt2A_Mt2Md(TSMASState *nstate, tensor* new_strain, double new_T, TSMASState *ostate)
235 ;
235 void Complete_Mt2A(TSMASState *nstate, tensor* new_strain, double new_T, TSMASState *ostate);
236 public:
237 TSMASolver() { falpha = 1.0; };
238
239 virtual PInelasticState __cdecl Load(PInelasticState res, tensor* new_strain,
240 double new_T, PInelasticState old_state);
241 virtual tensor4Ptr __cdecl NumericalJacobian(tensor4 L, TSMASState *nstate, TSMASState *ostate);
242 };

```

```

243
244 double phi1f(TSMAMaterial* smat, tensor* stress, double T, double c1);
245 double phi1r(TSMAMaterial* smat, tensor* stress, double T, double c2);
246 double phi2f(TSMAMaterial* smat, tensor* stress, double T, tensor* Lambda, double c2);
247 double phi2r(TSMAMaterial* smat, tensor* stress, double T, tensor* Lambda, double c2);
248
249
250 int PT2xi(int pt_indx);
251
252 #endif /* __SMA2004_H */

```

Listing VI.2 SMA numerical implementation source file

```

1 #include <stdlib.h>
2 #include <string.h>
3 #include <math.h>
4 #include <iostream.h>
5 #include <assert.h>
6 #include <fstream.h>
7 #include <cstring.h>
8 // #include <system.hpp>
9
10 #include "SMA2004.h"
11 #include "support.h"
12
13 #define LN(x) log(x)
14 #define SQR(x) (x*x)
15 #define TOL 1.0e-12
16 #define PHI_EPS 1.0e-6
17 #define EPS 1.0e-14
18 #define MAX_ITER_COUNT 16
19 #define SMAT(x) ((TSMAMaterial *)x)
20 #define SMATYPE(x) ((TSMATYPE *)x)
21
22 bool cnflag = true; // false;
23 int e_cnt = 899;
24 int e_indx = 1;
25 int allocated_mem;
26
27 ESMAError::ESMAError(char* AMsg) {
28     ofstream fout;
29
30     // fout.open("err.txt", ios::app);
31     // fout << AMsg << "\n\n";
32     // fout.close();
33
34 };
35
36
37 TSMAMaterial::TSMAMaterial(const char *str){
38     char* tmp = new char[strlen(str) + 1];
39
40     tmp = strcpy(tmp, str);
41
42     f3 = ReadDoubleArr(tmp, "f3", f3deg);
43     f3deg--;
44
45     rho = ReadDouble(tmp, "Density", ',', 6450.0);
46     EM = ReadDouble(tmp, "EM", ',', 30.0e9);
47     EA = ReadDouble(tmp, "EA", ',', 70.0e9);
48     nuM = ReadDouble(tmp, "PoissonM", ',', 0.3);
49     nuA = ReadDouble(tmp, "PoissonA", ',', 0.3);
50     aM = ReadDouble(tmp, "AlphaM", ',', 11.0e-6);
51     aA = ReadDouble(tmp, "AlphaA", ',', 6.6e-6);
52     cM = ReadDouble(tmp, "cM", ',', 2.12e6);
53     cA = ReadDouble(tmp, "cA", ',', 2.12e6);
54     As = ReadDouble(tmp, "As", ',', 295.0);
55     Af = ReadDouble(tmp, "Af", ',', 315.0);
56     Ms = ReadDouble(tmp, "Ms", ',', 291.0);
57     Mf = ReadDouble(tmp, "Mf", ',', 271.0);
58     H = ReadDouble(tmp, "H", ',', 0.05);
59     stress_s = ReadDouble(tmp, "stress_s", ',', 50.0e6);
60     stress_f = ReadDouble(tmp, "stress_f", ',', 100.0e6);
61     TO = ReadDouble(tmp, "TO", ',', 271.0);
62     delta_s0 = ReadDouble(tmp, "ds0", ',', -7.0e6*H/rho);
63     c01 = ReadDouble(tmp, "c01", ',', 1.0);
64     c02 = ReadDouble(tmp, "c02", ',', 0.0);
65     c03 = ReadDouble(tmp, "c03", ',', 0.0);
66     if (fabs(c01+c02+c03 - 1.0) > EPS){
67         throw ESMAError("Initial mass fractions have inconsistent sum.");
68     }
69     poly3 = NULL;
70     // Y1, Y2, Y3, b1, b2, b12, m1, m2

```



```

71     delta_c = cM - cA;
72
73
74     IsotropicStiffness(&CM, EM, nuM);
75
76
77     IsotropicCompliance(&SM, EM, nuM);
78     IsotropicStiffness(&CA, EA, nuA);
79     IsotropicCompliance(&SA, EA, nuA);
80     init_diag2(&AlphaM, aM);
81     init_diag2(&AlphaA, aA);
82
83     sub4(&dC, &CM, &CA);
84     sub4(&dS, &SM, &SA);
85     sub(&dAlpha, &AlphaM, &AlphaA);
86
87     Update();
88
89     delete[] tmp;
90 };
91
92 void TSMAMaterial::Update(){
93     double delta_s = 1.0/EM-1.0/EA;
94     double delta_a = aM-aA;
95     double tmp = 0.25*stress_s*(2.0*(H+(Ms-T0)*delta_a)+delta_s*stress_s);
96
97     tensor stress, lambda;
98
99     // Computing parameters related to Phi1
100    b1M = -(Mf-Ms)*(delta_c - delta_s0)+delta_c*(Mf*LN(Mf/T0)-Ms*LN(Ms/T0));
101    b1A = (Af-As)*(delta_c-delta_s0)+delta_c*(As*LN(As/T0)-Af*LN(Af/T0));
102    Y1 = 0.5*rho*(Af-Ms)*(delta_c-delta_s0)+delta_c*(Ms*LN(Ms/T0)-Af*LN(Af/T0));
103    delta_u1 =
104        0.5*((2.0*T0-Af-Ms)*delta_c+(Af+Ms)*delta_s0+delta_c*(Ms*LN(Ms/T0)+Af*LN(Af/T0)));
105
106    // checking Phi1
107    zero_tensor(&stress);
108    assert(fabs(phi1f(this, &stress, Ms, 0.0))<1e-7);
109    assert(fabs(phi1f(this, &stress, Mf, 1.0))<1e-7);
110    assert(fabs(phi1r(this, &stress, As, 1.0))<1e-7);
111    assert(fabs(phi1r(this, &stress, Af, 0.0))<1e-7);
112
113    // Computing parameters related to Phi2
114    /*****
115    b2M = (stress_f-stress_s)*(2*(H+(Ms-T0)*delta_a)+delta_s*(stress_f+stress_s))/(2.0*rho);
116    *****/
117    b2M = b1M + (Mf-Ms)*delta_a*stress_s/rho;
118    b2A = b1A;
119    Y2 = Y1 + tmp;
120    delta_u2 = delta_u1 + tmp/rho;
121
122    // Checking Phi2
123    stress[0][0] = stress_s;
124    zero_tensor(&lambda);
125    lambda[0][0] = H;
126    assert(fabs(phi2f(this, &stress, Ms, &lambda, 0.0))<1e-6);
127    /*****
128    stress[0][0] = stress_f;
129    assert(fabs(phi2f(this, &stress, Ms, &lambda, 1.0))<1e-7);
130    *****/
131    stress[0][0] = stress_s;
132    assert(fabs(phi2f(this, &stress, Mf, &lambda, 1.0))<1e-7);
133
134    zero_tensor(&stress);
135    zero_tensor(&lambda);
136    assert(fabs(phi2r(this, &stress, As, &lambda, 1.0))<1e-7);
137    assert(fabs(phi2r(this, &stress, Af, &lambda, 0.0))<1e-6);
138
139    // Computing parameters for Phi3
140    Y3 = H*stress_s;
141    b3 = (stress_f-stress_s)*H/rho;
142
143
144    // A = rho*b1M*c01 - rho*b2M*c02 - Y3;
145    // B = -rho*(b1M+b2M+b3);
146
147    delete[] poly3;
148
149    poly3deg = max(2, f3deg+1) - 1;
150    poly3 = new double[poly3deg + 1];
151    poly3[0] = 0.0;
152    poly3[1] = 0.0;
153    for (int i = 0; i <= f3deg; i++) poly3[i] -= f3[i];
154
155    assert(fabs(stress_s*H-Y3)<1e-7);
156    assert(fabs(stress_f*H-rho*b3-Y3)<1e-7);
157 }

```

```

158
159 /*
160 void TSMAMaterial::Update(){
161     double deltaS = 1.0/EM-1.0/EA;
162     double deltaA = aM-aA;
163     double A, B;
164
165     b1A = -delta_s0*(Af-As);
166     b1M = -delta_s0*(Ms-Mf);
167     // b2M = -delta_s0*(Ms-Mf) + (H + 0.5*deltaS*(stress_f+stress_s))*(stress_f-stress_s)/rho;
168     b2M = (H + 0.5*deltaS*(stress_f+stress_s))*(stress_f-stress_s)/rho;
169     b2A = -delta_s0*(Af-As);
170     m1 = 0.25*(b1A-b1M);
171     m2 = 0.25*(b2A-b2M);
172     Y1 = -0.25*rho*delta_s0*( Af + As - Ms - Mf );
173     Y2 = Y1 + 0.25*(stress_s + stress_f)*H + 0.125*deltaS*(SQR(stress_s) + SQR(stress_f));
174     Y3 = stress_s*H + rho*b1M + rho*(m1-m2);
175     delta_u10 = 0.5*delta_s0*(Ms+Af);
176     delta_u20 = delta_u10 + (0.5*stress_s*H + 0.25*deltaS*stress_s*stress_s)/rho;
177
178     b3 = (stress_f*H+rho*(-b2M+(m1-m2))-Y3)/rho;
179
180     A = rho*b1M*c01 - rho*b2M*c02 + rho*(m1-m2) - Y3;
181     B = -rho*(b1M+b2M+b3);
182
183     delete[] poly3;
184
185     poly3deg = max(2, f3deg+1) - 1;
186     poly3 = new double[poly3deg + 1];
187     poly3[0] = A;
188     poly3[1] = B;
189     for (int i = 0; i <= f3deg; i++) poly3[i] -= f3[i];
190
191     // assert(fabs(stress_s*H+0.5*deltaS*SQR(stress_s)+rho*delta_s0*Ms-rho*m2-rho*delta_u20-Y2)<1e-7);
192     // assert(fabs(stress_f*H+0.5*deltaS*SQR(stress_f)+rho*delta_s0*Mf-rho*b2M-rho*m2-rho*delta_u20-Y2)<1e-7);
193     // check consistency of A->Md
194     // assert(fabs(deltaA*(Ms-T0)+rho*delta_s0*Ms-rho*m1-rho*delta_u10-Y1)<1e-7);
195     // assert(fabs(deltaA*(Mf-T0)+rho*delta_s0*Mf-rho*b1M-rho*m1-rho*delta_u10-Y1)<1e-7);
196     // assert(fabs(rho*delta_s0*Af +rho*m2-rho*delta_u20+Y2)<1e-7);
197     // assert(fabs(rho*delta_s0*As-rho*b2A+rho*m2-rho*delta_u20+Y2)<1e-7);
198
199     // assert(fabs(rho*delta_s0*Ms-rho*m1-rho*delta_u10-Y1)<1e-7);
200 }
201 */
202 char* __cdecl TSMAMaterial::AsString(){
203     char buf[20];
204     char *res, *tmp;
205
206     tmp = res = StrCat("EA = ", gcvt(EA, 7, buf));
207     res = StrCat(res, "\nEM = "); delete[] tmp;
208     tmp = res; res = StrCat(res, gcvt(EM, 7, buf)); delete[] tmp;
209
210     tmp = res; res = StrCat(res, "\nnuA = "); delete[] tmp;
211     tmp = res; res = StrCat(res, gcvt(nuA, 7, buf)); delete[] tmp;
212
213     tmp = res; res = StrCat(res, "\nnuM = "); delete[] tmp;
214     tmp = res; res = StrCat(res, gcvt(nuM, 7, buf)); delete[] tmp;
215
216     tmp = res; res = StrCat(res, "\nAlphaM = "); delete[] tmp;
217     tmp = res; res = StrCat(res, gcvt(aM, 7, buf)); delete[] tmp;
218
219     tmp = res; res = StrCat(res, "\nAlphaA = "); delete[] tmp;
220     tmp = res; res = StrCat(res, gcvt(aA, 7, buf)); delete[] tmp;
221
222     tmp = res; res = StrCat(res, "\nnuA = "); delete[] tmp;
223     tmp = res; res = StrCat(res, gcvt(nuA, 7, buf)); delete[] tmp;
224
225     tmp = res; res = StrCat(res, "\ncM = "); delete[] tmp;
226     tmp = res; res = StrCat(res, gcvt(cM, 7, buf)); delete[] tmp;
227
228     tmp = res; res = StrCat(res, "\ncA = "); delete[] tmp;
229     tmp = res; res = StrCat(res, gcvt(cA, 7, buf)); delete[] tmp;
230
231     tmp = res; res = StrCat(res, "\nAs = "); delete[] tmp;
232     tmp = res; res = StrCat(res, gcvt(As, 7, buf)); delete[] tmp;
233
234     tmp = res; res = StrCat(res, "\nAf = "); delete[] tmp;
235     tmp = res; res = StrCat(res, gcvt(Af, 7, buf)); delete[] tmp;
236
237     tmp = res; res = StrCat(res, "\nMs = "); delete[] tmp;
238     tmp = res; res = StrCat(res, gcvt(Ms, 7, buf)); delete[] tmp;
239
240     tmp = res; res = StrCat(res, "\nMf = "); delete[] tmp;
241     tmp = res; res = StrCat(res, gcvt(Mf, 7, buf)); delete[] tmp;
242
243     tmp = res; res = StrCat(res, "\nH = "); delete[] tmp;
244     tmp = res; res = StrCat(res, gcvt(H, 7, buf)); delete[] tmp;

```

```

245
246     tmp = res; res = StrCat(res, "\nds0 = "); delete[] tmp;
247     tmp = res; res = StrCat(res, gcvt(delta_s0, 7, buf)); delete[] tmp;
248
249     return res;
250 }
251
252 bool TSMAMaterial::ParamByName(double &res, char* name){
253     if (strcmp(name, "T0") == 0){
254         res = T0;
255         return res;
256     }
257     else{
258         return false;
259     }
260 }
261
262 double J2(tensorPtr stress)
263 {
264     tensor s;
265
266     dev(&s, stress);
267     return norm(&s);
268 }
269
270
271 double P(TSMAMaterial* smat, tensor* stress, double T)
272 {
273     tensor tmp;
274     double res;
275
276     res = 0.5*contract(contract42(&tmp, &(smat->dS), stress), stress);
277     res += contract(&smat->dAlpha, stress)*(T-smat->T0);
278     res -= smat->rho*smat->delta_c*(T-smat->T0-T*LN(T/smat->T0));
279     res += smat->rho*smat->delta_s0*T;
280     return res;
281 }
282
283
284 tensor* dPdstress(TSMAMaterial* smat, tensor* res, tensor* stress, double T)
285 {
286     contract42(res, &(smat->dS), stress);
287     return add_smul(res, *res, smat->dAlpha, T-smat->T0);
288 }
289
290 double phi1P(TSMAMaterial* smat, tensor* stress, double T, double c1, double b)
291 {
292     double res;
293
294     res = (P(smat, stress, T) - smat->rho*(smat->delta_u1 + b*c1));
295     return res;
296 }
297
298 double phi1f(TSMAMaterial* smat, tensor* stress, double T, double c1)
299 {
300     return phi1P(smat, stress, T, c1, smat->b1M) - smat->Y1;
301 }
302
303 double phi1r(TSMAMaterial* smat, tensor* stress, double T, double c1)
304 {
305     return -phi1P(smat, stress, T, c1, smat->b1A) - smat->Y1;
306 }
307
308 double phi2P(TSMAMaterial* smat, tensor* stress, double T, tensor* Lambda, double c2, double b)
309 {
310     tensor tmp;
311     double res;
312
313     res = contract(stress, Lambda) + P(smat, stress, T);
314     res -= smat->rho*(smat->delta_u2 + b*c2);
315     return res;
316 }
317
318 double phi2f(TSMAMaterial* smat, tensor* stress, double T, tensor* Lambda, double c2)
319 {
320     return phi2P(smat, stress, T, Lambda, c2, smat->b2M) - smat->Y2;
321 }
322
323 double phi2r(TSMAMaterial* smat, tensor* stress, double T, tensor* Lambda, double c2)
324 {
325     return -phi2P(smat, stress, T, Lambda, c2, smat->b2A) - smat->Y2;
326 }
327
328
329 void TSMASState::init(){
330     zero_tensor(&strain);
331     zero_tensor(&stress);

```

```

332     zero_tensor(&strain_in);
333     dstress_dstrain = NULL;
334     xi1 = xi2 = xi3 = 0.0;
335     if (fmtrl) T = SMAT(fmtrl)->T0;
336 };
337
338 void TSMASState::SetMaterial(PInelasticMaterial amtrl){
339     TInelasticState::SetMaterial(amtrl);
340     if (fmtrl) T = SMAT(fmtrl)->T0;
341 };
342
343
344 void write_tnsr(string *s, tensor t, char *name){
345     char buf[30];
346
347     s->append("\n");
348     s->append(name);
349     s->append(" =\n");
350     for (int i = 0; i < 3; i++){
351         for (int j = 0; j < 3; j++){
352             s->append(gcvt(t[i][j], 14, buf));
353             s->append(" ");
354         }
355         s->append("\n");
356     }
357
358 }
359
360
361 char* TSMASState::GetAsString(char *res){
362     string s;
363     char buf[30];
364
365     s.append("xi1 = ");
366     s.append(gcvt(xi1, 14, buf));
367     s.append("\nxi2 = ");
368     s.append(gcvt(xi2, 14, buf));
369     s.append("\nxi3 = ");
370     s.append(gcvt(xi3, 14, buf));
371
372     write_tnsr(&s, strain, "strain");
373     s.append("\nT = ");
374     s.append(gcvt(T, 14, buf));
375
376     write_tnsr(&s, stress, "stress");
377     write_tnsr(&s, strain_in, "strain_in");
378
379     if (res != NULL){
380         if (strlen(res) < s.length()){
381             res = new char[s.length() + 1];
382         }
383     }
384     else
385         res = new char[s.length() + 1];
386     strcpy(res, s.c_str());
387     return res;
388 }
389
390 void TSMASState::SetAsString(const char *str){
391     char* tmp = new char[strlen(str) + 1];
392
393     tmp = strcpy(tmp, str);
394
395     xi1 = ReadDouble(tmp, "xi1", '\n', 0.0);
396     xi2 = ReadDouble(tmp, "xi2", '\n', 0.0);
397     xi3 = ReadDouble(tmp, "xi3", '\n', 0.0);
398
399     T = ReadDouble(tmp, "T", '\n', 0.0);
400     ReadTensor2(strain, tmp, "strain");
401     ReadTensor2(stress, tmp, "stress");
402     ReadTensor2(strain_in, tmp, "strain_in");
403 }
404
405
406
407 void TSMASState::Assign(TInelasticState* src){
408     TSMASState *sma_src;
409
410     if (this == src){
411         return;
412     }
413
414     TInelasticState::Assign(src);
415
416     T = *src->Temperature();
417     assign_tensor(&strain, src->Strain());
418     assign_tensor(&stress, src->Stress());

```

```

419  /*
420  if (src->DStressDStrain()){
421  if (!dstress_dstrain) dstress_dstrain = new tensor4[1];
422  assign_tensor4(dstress_dstrain, src->DStressDStrain());
423  }
424  else{
425  if (!dstress_dstrain) delete dstress_dstrain;
426  dstress_dstrain = NULL;
427  }
428  */
429  sma_src = (dynamic_cast<TSMASState *>(src));
430  assign_tensor(&strain_in, &sma_src->strain_in);
431
432  xi1 = sma_src->xi1;
433  xi2 = sma_src->xi2;
434  xi3 = sma_src->xi3;
435  }
436
437  tensor4* TSMASState::EffctvC(tensor4* res){
438  TSMAMaterial *smat = (TSMAMaterial*)GetMaterial();
439
440  double xi = c1() + c2();
441  double EffE = 1.0/(xi/smat->EM + (1.0-xi)/smat->EA);
442  double EffNu = xi*smat->nuM + (1.0-xi)*smat->nuA;
443
444  if (xi <= 0.0){
445  assign_tensor4(res, &smat->CA);
446  }
447  else if (xi >= 1.0){
448  assign_tensor4(res, &smat->CM);
449  }
450  else{
451  IsotropicStiffness(res, EffE, EffNu);
452  }
453
454  return res;
455  }
456
457  tensor4* TSMASState::EffctvS(tensor4* res){
458  TSMAMaterial *smat = (TSMAMaterial*)GetMaterial();
459
460  double xi = c1() + c2();
461  double EffE = 1.0/(xi/smat->EM + (1.0-xi)/smat->EA);
462  double EffNu = xi*smat->nuM + (1.0-xi)*smat->nuA;
463
464  if (xi <= 0.0){
465  assign_tensor4(res, &smat->SA);
466  }
467  else if (xi >= 1.0){
468  assign_tensor4(res, &smat->SM);
469  }
470  else{
471  IsotropicCompliance(res, EffE, EffNu);
472  }
473
474  return res;
475  }
476
477  tensor* TSMASState::EffctvA(tensor* res){
478  TSMAMaterial *smat = (TSMAMaterial*)GetMaterial();
479
480  double xi = c1() + c2();
481
482  if (xi <= 0.0){
483  assign_tensor(res, &smat->AlphaA);
484  }
485  else if (xi >= 1.0){
486  assign_tensor(res, &smat->AlphaM);
487  }
488  else{
489  interpolate(res, &smat->AlphaA, &smat->AlphaM, xi);
490  }
491  return res;
492  }
493  }
494
495  tensor4* TSMASState::DStressDStrain(){
496  if (dstress_dstrain){
497  return dstress_dstrain;
498  }
499  else{
500  dstress_dstrain = new tensor4[1];
501  EffctvC(dstress_dstrain);
502  return (dstress_dstrain);
503  }
504  };
505

```

```

506
507 tensor* TSMASState::InelasticStrain(tensor* res, int i){
508
509     if (res){
510         assign_tensor(res, &strain_in);
511         return res;
512     }
513     else return &strain_in;
514 }
515
516 double* TSMASState::InternalVar(int i){
517     switch (i){
518         case 1: return &xi1;
519         case 2: return &xi2;
520         case 3: return &xi3;
521         case 4:{
522             dbl_tag = c1();
523             return &dbl_tag;
524         }
525         case 5:{
526             dbl_tag = c2();
527             return &dbl_tag;
528         }
529         case 6:{
530             dbl_tag = c3();
531             return &dbl_tag;
532         }
533         default: cerr << "TSMASState::InternalVar - This method expects int i in [1..3]";
534                 return NULL;
535     }
536 }
537
538 tensorPtr TSMASState::J2FlowRule(tensorPtr res){
539     TSMAMaterial* smat = (TSMAMaterial*)fmtrl;
540
541     if (!res) {
542         res = new tensor[1];
543         zero_tensor(res);
544     }
545
546     if (norm(&stress) > 1e-16){
547         dev(res, &stress);
548         return scalar_mul(res, res, sqrt(3.0/2.0)*smat->H/norm(res));
549     }
550     {
551         zero_tensor(res);
552         return res;
553     }
554 }
555
556
557 tensor4* TSMASState::dJ2_dstress(tensor4Ptr res){
558
559     tensor s, tmp;
560     tensor4 tmp2;
561     double n_s;
562
563     identity4(*res);
564
565     identity2(tmp);
566     prod22(&tmp2, &tmp, &tmp);
567     add_smul4(res, *res, tmp2, -1.0/3.0);
568
569     dev(&s, &stress);
570     prod22(&tmp2, &s, &s);
571     n_s = norm(&s);
572     add_smul4(res, *res, tmp2, -1.0/SQR(n_s));
573     scalar_mul4(res, *res, sqrt(3.0/2.0)*SMAT(fmtrl)->H/n_s);
574
575     // Symmetrize22(*res);
576     return res;
577 }
578
579
580 tensor* TSMASState::Lambda2_reverse(tensor *res){
581     if (!res) {
582         res = new tensor[1];
583         zero_tensor(res);
584     }
585
586     zero_tensor(res);
587     if (norm(&strain_in) > 1e-16){
588         assign_tensor(res, &strain_in); //dev(res, &strain_in);
589         scalar_mul(res, res, sqrt(3.0/2.0)*SMAT(fmtrl)->H/norm(res));
590         if (c2() < 0.0)
591             negate_tensor(res);
592     }

```

```

593     else{
594         return Lambda2_forward(res);
595     }
596     return res;
597 }
598
599 /*
600 double TSMASState::dPhi2_reverse_q(){
601     return SMAT(fmtrl)->rho*SMAT(fmtrl)->bA;
602 };*/
603
604 /*****
605 /*
606 /*      Indexed transformation surface and derivatives.
607 /*
608 /*****
609
610 int PT2xi(int pt_indx){
611     switch (pt_indx){
612     case PT_SA_FORWARD: return XI_SA;
613     case PT_SA_REVERSE: return XI_SA;
614     case PT_SIM_FORWARD: return XI_SIM;
615     case PT_SIM_REVERSE: return XI_SIM;
616     case PT_REORIENT: return XI_REORIENT;
617     default: cerr << "PT2xi: Inappropriate index";
618             return 0;
619     }
620 }
621
622 double TSMASState::Phi(int pt_indx){
623     switch (pt_indx){
624     case PT_SIM_FORWARD: return Phi2_forward();
625     case PT_SIM_REVERSE: return Phi2_reverse();
626     case PT_SA_FORWARD: return Phi1_forward();
627     case PT_SA_REVERSE: return Phi1_reverse();
628     case PT_REORIENT: return Phi3();
629     default: cerr << "TSMASState::Phi - Inappropriate index";
630             return 0.0;
631     }
632 }
633
634 tensor* TSMASState::dPhi_dstress(tensor* res, int pt_indx){
635     switch (pt_indx){
636     case PT_SIM_FORWARD: return dPhi2_forward_dstress(res);
637     case PT_SIM_REVERSE: return dPhi2_reverse_dstress(res);
638     case PT_SA_FORWARD: return dPhi1_forward_dstress(res);
639     case PT_SA_REVERSE: return dPhi1_reverse_dstress(res);
640     case PT_REORIENT: return dPhi3_dstress(res);
641     default: cerr << "TSMASState::dPhi_dstress - Inappropriate index";
642             return NULL;
643     }
644 }
645
646 double TSMASState::dPhi_dxi(int pt_indx, int xi_indx)
647 {
648     switch (pt_indx){
649     case PT_SIM_FORWARD:
650         switch (xi_indx){
651         case 1: return dPhi2_forward_dxi1();
652         case 2: return dPhi2_forward_dxi2();
653         case 3: return dPhi2_forward_dxi3();
654         }
655     case PT_SIM_REVERSE:
656         switch (xi_indx){
657         case 1: return dPhi2_reverse_dxi1();
658         case 2: return dPhi2_reverse_dxi2();
659         case 3: return dPhi2_reverse_dxi3();
660         }
661     case PT_SA_FORWARD:
662         switch (xi_indx){
663         case 1: return dPhi1_forward_dxi1();
664         case 2: return dPhi1_forward_dxi2();
665         case 3: return dPhi1_forward_dxi3();
666         }
667     case PT_SA_REVERSE:
668         switch (xi_indx){
669         case 1: return dPhi1_reverse_dxi1();
670         case 2: return dPhi1_reverse_dxi2();
671         case 3: return dPhi1_reverse_dxi3();
672         }
673     case PT_REORIENT: ;
674         switch (xi_indx){
675         case 1: return dPhi3_dxi1();
676         case 2: return dPhi3_dxi2();
677         case 3: return dPhi3_dxi3();
678         }
679     default: cerr << "TSMASState::dPhi_dxi - Inappropriate index";

```

```

680         return NULL;
681     }
682 }
683
684 tensor* TSMASState::Lambda(tensor *res, int pt_indx){
685     switch (pt_indx){
686     case PT_SA_FORWARD: return Lambda1_forward(res);
687     case PT_SA_REVERSE: return Lambda1_reverse(res);
688     case PT_SIM_FORWARD: return Lambda2_forward(res);
689     case PT_SIM_REVERSE: return Lambda2_reverse(res);
690     case PT_REORIENT: return Lambda3(res);
691     default: cerr << "TSMASState::Lambda - Inappropriate index";
692             return NULL;
693     }
694 }
695
696 tensor4* TSMASState::dLambda_dstress(tensor4 *res, int pt_indx){
697     switch (pt_indx){
698     case PT_SA_FORWARD: return dLambda1_forward_dstress(res);
699     case PT_SA_REVERSE: return dLambda1_reverse_dstress(res);
700     case PT_SIM_FORWARD: return dLambda2_forward_dstress(res);
701     case PT_SIM_REVERSE: return dLambda2_reverse_dstress(res);
702     case PT_REORIENT: return dLambda3_dstress(res);
703     default: cerr << "TSMASState::dLambda_dstress - Inappropriate index";
704             return NULL;
705     }
706 }
707
708 tensor* TSMASState::dxi_dstress(tensor* res, int pt_indx){
709     dPhi_dstress(res, pt_indx);
710     return scalar_mul(res, res,
711         -1.0/dPhi_dxi(pt_indx, PT2xi(pt_indx)));
712 }
713
714 }
715
716 tensor4* TSMASState::DStressDStrain_Alq(tensor4* res, TInelasticState *old_state){
717     tensor X, L, tmp2;
718     tensor4 tmp4;
719     int xi_indx, pt_indx;
720     TSMASState *ostate = (dynamic_cast<TSMASState *>(old_state));
721     TSMAMaterial* smat = (TSMAMaterial*)fmatrl;
722
723     pt_indx = PT_NONE;
724
725     if (xi1 > ostate->xi1 + EPS){
726         pt_indx = PT_SA_FORWARD;
727     } else
728     if (xi1 < ostate->xi1 - EPS){
729         pt_indx = PT_SA_REVERSE;
730     }
731
732     if (xi2 > ostate->xi2 + EPS){
733         pt_indx = PT_SIM_FORWARD;
734     } else
735     if (xi2 < ostate->xi2 - EPS){
736         pt_indx = PT_SIM_REVERSE;
737     }
738
739     if (xi3 > ostate->xi3 + EPS){
740         pt_indx = PT_REORIENT;
741     }
742     xi_indx = PT2xi(pt_indx);
743
744     if (xi_indx != XI_REORIENT){
745         contract42(&tmp2, &smat->dS, &stress);
746         add_smul(&tmp2, tmp2, smat->dAlpha, T-smat->T0);
747     }
748     else
749         zero_tensor(&tmp2);
750     Lambda(&L, pt_indx);
751     add(&tmp2, &tmp2, &L);
752     dxi_dstress(&X, pt_indx);
753     prod22((tensor4Ptr)res, &tmp2, &X);
754     EffctvS(&tmp4);
755     add4(res[0], res[0], tmp4);
756     dLambda_dstress(&tmp4, pt_indx);
757     add_smul4(res, res[0], tmp4,
758         GetInternalVar(xi_indx) - ostate->GetInternalVar(xi_indx));
759     invert4(res);
760     return res;
761 }
762 }
763
764 tensor4* TSMASState::DStressDStrain_Alq2(tensor4* res, TInelasticState *old_state){
765
766     tensor X1, X2, L, tmp1, tmp2;

```



```

767     tensor4 tmp4;
768     int xi1_indx, pt1_indx;
769     int xi2_indx, pt2_indx;
770     TSMASState *ostate = (dynamic_cast<TSMASState *>(old_state));
771     TSMAMaterial* smat = (TSMAMaterial*)fmtrl;
772
773     pt1_indx = PT_NONE;
774     pt2_indx = PT_NONE;
775
776     if (xi1 > ostate->xi1 + EPS){
777         pt1_indx = PT_SA_FORWARD;
778     } else
779     if (xi1 < ostate->xi1 - EPS){
780         pt1_indx = PT_SA_REVERSE;
781     }
782
783     if (xi2 > ostate->xi2 + EPS){
784         pt2_indx = PT_SIM_FORWARD;
785     } else
786     if (xi2 < ostate->xi2 - EPS){
787         pt2_indx = PT_SIM_REVERSE;
788     }
789     if (pt1_indx == PT_NONE){
790         pt1_indx = pt2_indx;
791         pt2_indx = PT_NONE;
792     }
793
794     if (fabs(xi3 - ostate->xi3) > EPS){
795         pt2_indx = PT_REORIENT;
796     }
797     if (pt1_indx == PT_NONE){
798         pt1_indx = pt2_indx;
799         pt2_indx = PT_NONE;
800     }
801
802     xi1_indx = PT2xi(pt1_indx);
803     xi2_indx = PT2xi(pt2_indx);
804
805     if ((xi1_indx != XI_REORIENT) || (xi2_indx != XI_REORIENT)){
806         contract42(&tmp1, &smat->dS, &stress);
807         add_smul(&tmp1, tmp1, smat->dAlpha, T-smat->T0);
808     }
809     else
810         zero_tensor(&tmp1);
811     if (xi2_indx != XI_REORIENT){
812         Lambda(&L, pt2_indx);
813         add(&tmp2, &tmp1, &L);
814     }
815     else
816         Lambda(&tmp2, pt2_indx);
817
818     if (xi1_indx != XI_REORIENT){
819         Lambda(&L, pt1_indx);
820         add(&tmp1, &tmp1, &L);
821     }
822     else
823         Lambda(&tmp1, pt1_indx);
824     dxi_dstress(&X1, pt1_indx);
825     dxi_dstress(&X2, pt2_indx);
826     prod22(res, &tmp1, &X1);
827     prod22(&tmp4, &tmp2, &X2);
828     add4(res[0], res[0], tmp4);
829     EffctvS(&tmp4);
830     add4(res[0], res[0], tmp4);
831
832     dLambda_dstress(&tmp4, pt1_indx);
833     add_smul4(res, res[0], tmp4,
834             GetInternalVar(xi1_indx) - ostate->GetInternalVar(xi1_indx));
835     dLambda_dstress(&tmp4, pt2_indx);
836     add_smul4(res, res[0], tmp4,
837             GetInternalVar(xi2_indx) - ostate->GetInternalVar(xi2_indx));
838
839     invert4(res);
840     return res;
841 }
842
843
844 /*****
845 /*
846 /*      A->M^t transformation surface and derivatives.
847 /*
848 /*****
849
850 double TSMASState::Phi1_forward()
851 {
852     return phi1f(SMAT(fmtrl), &stress, T, c1());
853 };

```

```

854
855 // Derivatives
856 tensor* TSMASState::dPhi1_forward_dstress(tensor* res)
857 {
858     return dPdstress(SMAT(fmtrl), res, &stress, T);
859 // TSMAMaterial* smat = (TSMAMaterial*)fmtrl;
860 //
861 // contract42(res, @(&smat->dS), @stress);
862 // return add_smul(res, *res, smat->dAlpha, T-smat->T0);
863
864 };
865
866 double TSMASState::dPhi1_forward_dxi1()
867 {
868     TSMAMaterial* smat = (TSMAMaterial*)fmtrl;
869     return -smat->rho*smat->b1M;
870 };
871
872 double TSMASState::dPhi1_forward_dxi2()
873 {
874     return 0.0;
875 };
876
877
878 double TSMASState::dPhi1_forward_dxi3()
879 {
880     TSMAMaterial* smat = (TSMAMaterial*)fmtrl;
881     return smat->rho*smat->b1M;
882 };
883
884 /*****
885 /*
886 /*      M^-t->A transformation surface and derivatives.
887 /*
888 /*
889 /*****
890 double TSMASState::Phi1_reverse()
891 {
892     return phi1r(SMAT(fmtrl), &stress, T, c1());
893 };
894
895 tensor* TSMASState::dPhi1_reverse_dstress(tensor* res)
896 {
897     dPdstress(SMAT(fmtrl), res, &stress, T);
898     negate_tensor(res);
899     return res;
900 };
901
902 double TSMASState::dPhi1_reverse_dxi1()
903 {
904     TSMAMaterial* smat = (TSMAMaterial*)fmtrl;
905     return smat->rho*smat->b1A;
906 };
907
908 double TSMASState::dPhi1_reverse_dxi2()
909 {
910     return 0.0;
911 };
912
913
914 double TSMASState::dPhi1_reverse_dxi3()
915 {
916     TSMAMaterial* smat = (TSMAMaterial*)fmtrl;
917     return -smat->rho*smat->b1A;
918 };
919
920
921 /*****
922 /*
923 /*      A->M^-d transformation surface and derivatives.
924 /*
925 /*
926 /*****
927 double TSMASState::Phi2_forward(){
928     TSMAMaterial* smat = (TSMAMaterial*)fmtrl;
929     double res, phi_3;
930     tensor Lambda, tmp;
931
932     res = phi2f(SMAT(fmtrl), &stress, T, Lambda2_forward(&Lambda), c2());
933
934 //
935 // if (res > 0.0){
936 //     phi_3 = sqrt(3.0/2.0)*smat->H*J2(@stress) +
937 //     SMAT(fmtrl)->rho*(SMAT(fmtrl)->b1M)-SMAT(fmtrl)->Y3;
938 //     phi_3 = sqrt(3.0/2.0)*smat->H*J2(@stress) -SMAT(fmtrl)->Y3;
939 //     if (phi_3 < 0.0){
940 //         cout < "XXXXXXXXXX";

```

```

941     return phi_3;
942   }
943 }
944 */
945 return res;
946 }
947
948
949 tensorPtr TSMASState::dPhi2_forward_dstress(tensorPtr res){
950   tensor tmp;
951
952   Lambda2_forward(res);
953   //add(res, res, contract42(@tmp, @SMAT(fmtrl)->dS, @stress));
954   //add_smul(res, *res, SMAT(fmtrl)->dAlpha, T - SMAT(fmtrl)->TO);
955   add(res, res, dPdstress(SMAT(fmtrl), &tmp, &stress, T));
956   return res;
957 }
958
959
960 double TSMASState::dPhi2_forward_dxi1()
961 {
962   return 0.0;
963 };
964
965 double TSMASState::dPhi2_forward_dxi2()
966 {
967   TSMAMaterial* smat = (TSMAMaterial*)fmtrl;
968
969   return -smat->rho*smat->b2M;
970 };
971
972 double TSMASState::dPhi2_forward_dxi3()
973 {
974   TSMAMaterial* smat = (TSMAMaterial*)fmtrl;
975
976   return -smat->rho*smat->b2M;
977 };
978
979 /*
980 tensorPtr TSMASState::dPhi2_forward_dstress(tensorPtr res){
981   tensor tmp;
982
983   Lambda2_forward(res);
984   add(res, res, contract42(@tmp, @SMAT(fmtrl)->dS, @stress));
985   add_smul(res, *res, SMAT(fmtrl)->dAlpha, T - SMAT(fmtrl)->TO);
986   return res;
987 }
988 */
989
990 /*****
991 */
992 /*      M^d->A transformation surface and derivatives.
993 */
994 /*****
995
996 double TSMASState::Phi2_reverse(){
997   tensor Lambda;
998   return phi2r(SMAT(fmtrl), &stress, T, Lambda2_reverse(&Lambda), c2());
999 }
1000
1001 tensorPtr TSMASState::dPhi2_reverse_dstress(tensorPtr res){
1002   tensor tmp;
1003
1004   Lambda2_reverse(res);
1005   add(res, res, dPdstress(SMAT(fmtrl), &tmp, &stress, T));
1006   //add(res, res, contract42(@tmp, @SMAT(fmtrl)->dS, @stress));
1007   //add_smul(res, *res, SMAT(fmtrl)->dAlpha, T - SMAT(fmtrl)->TO);
1008   negate_tensor(res);
1009   return res;
1010 }
1011
1012 double TSMASState::dPhi2_reverse_dxi1(){
1013   return 0.0;
1014 };
1015
1016 double TSMASState::dPhi2_reverse_dxi2(){
1017   return SMAT(fmtrl)->rho*SMAT(fmtrl)->b2A;
1018 };
1019
1020 double TSMASState::dPhi2_reverse_dxi3(){
1021   return SMAT(fmtrl)->rho*SMAT(fmtrl)->b2A;
1022 }
1023
1024 /*****
1025 */
1026 /*      M^t->M^d transformation surface and derivatives.
1027 */

```

```

1028 /*****
1029
1030 double TSMASState::Phi3()
1031 {
1032     TSMAMaterial* smat = (TSMAMaterial*)fmtrl;
1033     tensor tmp;
1034     double hard = 0;
1035
1036     hard += (contract(Lambda3(&tmp), &stress) - smat->Y3);
1037     hard -= smat->rho*smat->b3*c2();
1038     return hard;
1039 };
1040
1041 tensorPtr TSMASState::dPhi3_dstress(tensorPtr res){
1042     return Lambda3(res);
1043 };
1044
1045 double TSMASState::dPhi3_dx11()
1046 {
1047     return 0.0;
1048 };
1049
1050 double TSMASState::dPhi3_dx12()
1051 {
1052     TSMAMaterial* smat = (TSMAMaterial*)fmtrl;
1053
1054     return -smat->rho*smat->b3;
1055 };
1056
1057 double TSMASState::dPhi3_dx13()
1058 {
1059     TSMAMaterial* smat = (TSMAMaterial*)fmtrl;
1060     return -smat->rho*smat->b3;
1061 };
1062
1063
1064 /*****
1065 /*
1066 /*      General purpose routines
1067 /*
1068 /*****
1069
1070
1071 double TSMASState::min_xi1(){
1072     TSMAMaterial *smat = (TSMAMaterial *)fmtrl;
1073
1074     return max(xi3 - smat->c01, smat->c03 - xi2 - 1.0);
1075 };
1076
1077
1078 double TSMASState::min_xi2(){
1079     TSMAMaterial *smat = (TSMAMaterial *)fmtrl;
1080
1081     return max(-xi3 - smat->c02, smat->c03 - xi1 - 1.0);
1082 };
1083
1084 double TSMASState::min_xi3(){
1085     TSMAMaterial *smat = (TSMAMaterial *)fmtrl;
1086
1087     return max(-xi2 - smat->c02, smat->c01 + xi1 - 1.0);
1088 };
1089
1090
1091 double TSMASState::max_xi1(){
1092     TSMAMaterial *smat = (TSMAMaterial *)fmtrl;
1093
1094     return min(smat->c03 - xi2, 1.0 - smat->c01 + xi3);
1095 };
1096
1097 double TSMASState::max_xi2(){
1098     TSMAMaterial *smat = (TSMAMaterial *)fmtrl;
1099
1100     return min(smat->c03 - xi1, 1.0 - smat->c02 - xi3);
1101 };
1102
1103 double TSMASState::max_xi3(){
1104     TSMAMaterial *smat = (TSMAMaterial *)fmtrl;
1105
1106     return min(smat->c01 + xi1, 1.0 - smat->c02 - xi2);
1107 };
1108
1109 TInelasticState *TSMASolver::LinearLoad(TInelasticState *res, tensor* new_strain,
1110 double new_T, TInelasticState *old_state){
1111
1112     TSMAMaterial* mtrl = (TSMAMaterial*)old_state->GetMaterial();
1113     tensor e_strain, thrml_strain;
1114     tensor4 C;

```

```

1115     res->Assign(old_state);
1116
1117     assign_tensor(res->Strain(), new_strain);
1118     *(res->Temperature()) = new_T;
1119
1120     contract42(res->Stress(), res->EffctvC(&C), res->ElasticStrain(&e_strain));
1121
1122     if (SMASSTATE(res)->dstress_dstrain != NULL){
1123         res->EffctvC(SMASSTATE(res)->dstress_dstrain);
1124     }
1125 }
1126
1127     return res;
1128 }
1129
1130 /*
1131 int TSMASolver::LoadDirection2(tensor* new_strain, double new_T, TSMASState *old_state){
1132     TSMASState trial;
1133     double phi_new, phi_old;
1134     int res1 = 0, res2 = 0;
1135
1136     trial.Assign(old_state);
1137     LinearLoad(&trial, new_strain, new_T, old_state);
1138
1139     if (old_state->is_A2Md_possible()) {
1140         phi_new = trial.Phi2_forward();
1141         phi_old = old_state->Phi2_forward();
1142
1143         if (phi_new > phi_old + EPS){
1144             if (phi_new <= 0.0) res1 = 0;
1145             else res1 = 1;
1146         }
1147     }
1148
1149     if (old_state->is_Md2A_possible() ) {
1150         phi_new = trial.Phi2_reverse();
1151         phi_old = old_state->Phi2_reverse();
1152
1153         if (phi_new > phi_old + EPS){
1154             if (phi_new <= 0.0) res2 = 0;
1155             else res2 = -1;
1156         }
1157     }
1158
1159     assert(!((res1 == 1) && (res2 == -1)));
1160
1161     if (res1) return res1;
1162     if (res2) return res2;
1163     return 0;
1164 };*/
1165
1166 TinelasticState *TSMASolver::SimoCPP(TSMASState *res, tensor* new_strain,
1167 double new_T, TSMASState* old_state, int pt_indx){
1168     TSMAMaterial *smat = SMAT(old_state->GetMaterial());
1169
1170     double phi_0, phi_k, dxi_k, M22, xi_n, xi_new;
1171     tensor M12, M21, dstress, F_k, tmp;
1172     tensor4 M11inv;
1173     int i_cnt = 0;
1174     int xi_indx = PT2xi(pt_indx);
1175
1176     bool flag;
1177
1178     flag = res == old_state;
1179     if (flag){
1180         old_state = new TSMASState;
1181         old_state->Assign(res);
1182     }
1183
1184     assign_tensor(&res->strain, new_strain);
1185     res->T = new_T;
1186
1187     // Check the yield surface.
1188     phi_0 = phi_k = res->Phi(pt_indx);
1189     if (phi_k > PHI_EPS)
1190     {
1191         zero_tensor(&F_k);
1192         xi_n = old_state->GetInternalVar(xi_indx);
1193         //while ( ((fabs(phi_k/phi_0) > TOL) || (fabs(phi_k) > PHI_EPS)) || (norm(&F_k) > TOL))
1194         while ( (fabs(phi_k) > PHI_EPS) || (norm(&F_k) > TOL))
1195         {
1196             negate_tensor(res->HookeResidual(&F_k));
1197             phi_k = -phi_k;
1198         }
1199     }
1200
1201

```

```

1202     res->dHookeResidual_dstress(&M11inv, pt_indx, old_state);
1203     invert4(&M11inv);
1204     res->dHookeResidual_dPT(&M12, pt_indx);
1205     res->dPhi_dstress(&M21, pt_indx);
1206     M22 = res->dPhi_dxi(pt_indx, xi_indx);
1207
1208     dxi_k = (phi_k - contract(&M21, contract42(&tmp, &M11inv, &F_k)))/
1209             (M22 - contract(&M21, contract42(&tmp, &M11inv, &M12)));
1210     add_smul(&tmp, F_k, M12, -dxi_k);
1211     contract42(&dstress, &M11inv, &tmp);
1212
1213     add(res->Stress(), res->Stress(), &dstress);
1214     xi_new = res->GetInternalVar(xi_indx) + dxi_k;
1215     add_smul(&res->strain_in, old_state->strain_in,
1216             *(res->Lambda(&tmp, pt_indx)), xi_new - xi_n);
1217     res->SetInternalVar(xi_indx, xi_new);
1218     i_cnt++;
1219     phi_k = res->Phi(pt_indx);
1220
1221     if (i_cnt > MAX_ITER_COUNT){
1222         double T_int;
1223         TSMASState intrm = TSMASState();
1224         interpolate(&tmp, old_state->Strain(), new_strain, 0.5);
1225         T_int = (new_T + *(old_state->Temperature()))/2.0;
1226         LinearLoad(&intrm, &tmp, T_int, old_state);
1227         if (intrm.Phi(pt_indx) > PHI_EPS){
1228             SimoCPP(&intrm, &tmp, T_int, old_state, pt_indx);
1229             if ((pt_indx == PT_SIM_REVERSE) && (intrm.xi2 < intrm.min_xi2())){
1230                 RestoreConsistency_Md2A(&intrm, new_strain, new_T, old_state);
1231                 if (flag) delete old_state;
1232                 return res;
1233             }
1234         }
1235         LinearLoad(res, new_strain, new_T, &intrm);
1236         SimoCPP(res, new_strain, new_T, &intrm, pt_indx);
1237         break;
1238     }
1239 }
1240 }
1241 if (res->dstress_dstrain){
1242     res->DStressDStrain_Alg(res->dstress_dstrain, old_state);
1243 }
1244 if (flag) delete old_state;
1245 return res;
1246 }
1247
1248 TInelasticState *TSMASolver::SimoCPP2(TSMASState *res, tensor* new_strain,
1249 double new_T, TSMASState* old_state, int pt_indx1, int pt_indx2){
1250
1251     TSMAMaterial *smat = SMAT(old_state->GetMaterial());
1252
1253     double phi1, phi2, dxi1, dxi2, xii_n, xi2_n, xii_new, xi2_new;
1254     double M22, M23, M32, M33, detM;
1255     double phi10, phi20, F0;
1256     tensor M12, M13, M21, M31;
1257     tensor dstress, F, tmp, tmp2, tmp3;
1258     tensor4 M11inv;
1259
1260     int i_cnt = 0;
1261     int xii_indx, xi2_indx;
1262
1263     xii_indx = PT2xi(pt_indx1);
1264     xi2_indx = PT2xi(pt_indx2);
1265
1266     bool flag;
1267
1268     flag = res == old_state;
1269     if (flag){
1270         old_state = new TSMASState;
1271         old_state->Assign(res);
1272     }
1273
1274     assign_tensor(&res->strain, new_strain);
1275     res->T = new_T;
1276
1277     phi10 = phi1 = res->Phi(pt_indx1);
1278     phi20 = phi2 = res->Phi(pt_indx2);
1279     if ((phi10 > PHI_EPS) || (phi20 > PHI_EPS)){
1280         if (fabs(phi10) <= PHI_EPS){
1281             phi10 = 1.0;
1282         }
1283         if (fabs(phi20) <= PHI_EPS){
1284             phi20 = 1.0;
1285         }
1286         zero_tensor(&F);
1287         xii_n = old_state->GetInternalVar(xii_indx);
1288         xi2_n = old_state->GetInternalVar(xi2_indx);

```

```

1289 //while ( (fabs(phi1/phi10) > TOL) || (fabs(phi2/phi20) > TOL) || (norm(&F) > TOL))
1290 while ( (fabs(phi1) > PHI_EPS) || (fabs(phi2) > PHI_EPS) || (norm(&F) > TOL))
1291 {
1292     // right hand side. Don't optimize for clarity.
1293     negate_tensor(res->HookeResidual(&F));
1294     phi1 = -phi1;
1295     phi2 = -phi2;
1296
1297     // F derivatives
1298     res->dHookeResidual_dstress2(&M11inv, pt_indx1, pt_indx2, old_state);
1299     invert4(&M11inv);
1300     res->dHookeResidual_dPT(&M12, pt_indx1);
1301     res->dHookeResidual_dPT(&M13, pt_indx2);
1302     // Phi1 derivatives
1303     res->dPhi_dstress(&M21, pt_indx1);
1304     M22 = res->dPhi_dxi(pt_indx1, xi1_indx);
1305     M23 = res->dPhi_dxi(pt_indx1, xi2_indx);
1306     // Phi2 derivatives
1307     res->dPhi_dstress(&M31, pt_indx2);
1308     M32 = res->dPhi_dxi(pt_indx2, xi1_indx);
1309     M33 = res->dPhi_dxi(pt_indx2, xi2_indx);
1310
1311     // Eliminating the dstress from the equations form the matrix
1312     contract42(&tmp2, &M11inv, &M12);
1313     contract42(&tmp3, &M11inv, &M13);
1314     M22 -= contract(&M21, &tmp2);
1315     M32 -= contract(&M31, &tmp2);
1316     M23 -= contract(&M21, &tmp3);
1317     M33 -= contract(&M31, &tmp3);
1318     // ... and right-hand side
1319     contract42(&tmp, &M11inv, &F);
1320     phi1 -= contract(&M21, &tmp);
1321     phi2 -= contract(&M31, &tmp);
1322
1323     // Solving the reduced system for dxi1 and dxi2
1324     detM = M22*M33-M32*M23;
1325     dxi1 = (phi1*M33-phi2*M23)/detM;
1326     dxi2 = (phi2*M22-phi1*M32)/detM;
1327     // Back-substituting to find the stress
1328     add_smul(&tmp, F, M12, -dxi1);
1329     add_smul(&tmp, tmp, M13, -dxi2);
1330     contract42(&dstress, &M11inv, &tmp);
1331
1332     // Updating the state variables
1333     add(res->Stress(), res->Stress(), &dstress);
1334     xi1_new = res->GetInternalVar(xi1_indx) + dxi1;
1335     xi2_new = res->GetInternalVar(xi2_indx) + dxi2;
1336     add_smul(&res->strain_in, old_state->strain_in,
1337             *(res->Lambda(&tmp, pt_indx1)), xi1_new - xi1_n);
1338     add_smul(&res->strain_in, res->strain_in,
1339             *(res->Lambda(&tmp, pt_indx2)), xi2_new - xi2_n);
1340     res->SetInternalVar(xi1_indx, xi1_new);
1341     res->SetInternalVar(xi2_indx, xi2_new);
1342     i_cnt++;
1343
1344     phi1 = res->Phi(pt_indx1);
1345     phi2 = res->Phi(pt_indx2);
1346
1347     if (i_cnt > MAX_ITER_COUNT){
1348         double T_int;
1349         TSMASState intrm = TSMASState();
1350
1351         interpolate(&tmp, old_state->Strain(), new_strain, 0.5);
1352         T_int = (new_T + *(old_state->Temperature()))/2.0;
1353         LinearLoad(&intrm, &tmp, T_int, old_state);
1354         if ((intrm.Phi(pt_indx1) > PHI_EPS) || (intrm.Phi(pt_indx2) > PHI_EPS)){
1355             SimoCPP2(&intrm, &tmp, T_int, old_state, pt_indx1, pt_indx2);
1356         }
1357         if (intrm.c3() < 0.0)
1358         {
1359             res->Assign(&intrm);
1360         }
1361         else
1362         {
1363             LinearLoad(res, new_strain, new_T, &intrm);
1364             SimoCPP2(res, new_strain, new_T, &intrm, pt_indx1, pt_indx2);
1365         }
1366         break;
1367     }
1368 }
1369 }
1370 }
1371 if (res->dstress_dstrain){
1372     res->DStressDStrainAlg2(res->dstress_dstrain, old_state);
1373 }
1374 if (flag) delete old_state;
1375 return res;

```

```

1376 }
1377
1378
1379 TInelasticState *TSMASolver::SimoCPP_forward(TSMASState *res, tensor* new_strain,
1380 double new_T, TSMASState* old_state){
1381
1382     TSMAMaterial *smat = SMAT(old_state->GetMaterial());
1383
1384     double phi_0, phi_k, dxi_k, T_int, c;
1385     tensor R, r, v, tmp1, tmp2, dStress, dStrainT, e_strain;
1386     tensor4 tmp, *E;
1387     int i_cnt = 0;
1388     bool flag;
1389
1390     flag = res == old_state;
1391     if (flag){
1392         old_state = new TSMASState;
1393         old_state->Assign(res);
1394         LinearLoad(res, new_strain, new_T, old_state);
1395     }
1396
1397     // Elastic prediction.
1398     // Check the yield surface.
1399     phi_0 = phi_k = res->Phi2_forward();
1400     E = res->EffctvC(res->DStressDStrain());
1401     if (phi_k >= 0.0)
1402     {
1403         zero_tensor(&R);
1404         //while ( (fabs(phi_k/phi_0) > TOL) || (norm(&R) > TOL))
1405         while ( (fabs(phi_k) > PHI_EPS) || (norm(&R) > TOL))
1406         {
1407             res->Lambda2_forward(&r);
1408             res->dPhi2_forward_dstress(&v);
1409
1410             sub(&R, &(old_state->strain_in), &(res->strain_in));
1411             add_smul(&R, R, r, res->xi2 - old_state->xi2);
1412
1413             add_smul4(E,
1414                 *(res->EffctvS(&tmp)), *(res->dLambda2_forward_dstress(E)), res->xi2 - old_state->xi2);
1415             invert4(E);
1416
1417             // Compute delta xi
1418             c = contract(&v, contract42(&tmp2, E, &v)) - res->dPhi2_forward_dxi2();
1419             dxi_k = (phi_k - contract(&v, contract42(&tmp1, E, &R)))/c;
1420             // Compute delta stress
1421             add_smul(&dStress, tmp1, tmp2, dxi_k);
1422             negate_tensor(&dStress);
1423
1424             contract42(&tmp1, &(smat->dS), &(res->stress));
1425             add_smul(&tmp1, tmp1, smat->dAlpha, new_T - smat->T0);
1426             // Compute delta e^t. Note that tmp still holds EffectiveS
1427             contract42(&dStrainT, &tmp, &dStress);
1428             add_smul(&dStrainT, dStrainT, tmp1, dxi_k);
1429             negate_tensor(&dStrainT);
1430             // Updating xi2
1431             res->xi2 += dxi_k;
1432             // Updating e^t
1433             add(&(res->strain_in), &(res->strain_in), &dStrainT);
1434             //add(&(res->stress), &(res->stress), &dStress);
1435
1436             contract42(&(res->stress), res->EffctvC(&tmp), res->ElasticStrain(&e_strain));
1437             phi_k = res->Phi2_forward();
1438
1439             if (++i_cnt > MAX_ITER_COUNT){
1440                 interpolate(&tmp1, old_state->Strain(), new_strain, 0.5);
1441                 T_int = (new_T + *(old_state->Temperature()))/2.0;
1442                 LinearLoad(res, &tmp1, T_int, old_state);
1443                 SimoCPP_forward(res, &tmp1, T_int, old_state);
1444                 return SimoCPP_forward(res, new_strain, new_T, res);
1445             }
1446         }
1447         // Tangent
1448         add_smul4(E,
1449             *(res->EffctvS(&tmp)), *(res->dLambda2_forward_dstress(E)), res->xi2-old_state->xi2);
1450         invert4(E);
1451         res->dPhi2_forward_dstress(&v);
1452         contract42(&tmp1, E, &v);
1453         prod22(&tmp, &tmp1, &tmp1);
1454         c = contract(&v, contract42(&tmp2, E, &v)) - res->dPhi2_forward_dxi2();
1455         add_smul4(E, *E, tmp, -1.0/c);
1456
1457         // tensor4 E1;
1458         // res->DStressDStrain_Alg(&E1, old_state);
1459         res->DStressDStrain_Alg(E, old_state);
1460         // sub4(&E1, &E1, E);
1461
1462         if (!cnflag){

```



```

1463         cnflag = true;
1464         NumericalJacobian(*E, res, old_state);
1465         cnflag = false;
1466     }
1467 }
1468 if (flag) delete old_state;
1469 return res;
1470 }
1471
1472
1473 TInelasticState *TSMASolver::SimoCPP_reverse(TSMASState *res, tensor* new_strain,
1474 double new_T, TSMASState* old_state){
1475
1476     TSMAMaterial *smat = SMAT(old_state->GetMaterial());
1477
1478     double phi_0, phi_k, dxi_k;
1479     tensor R, r, v, tmp1, tmp2, dStress, dStrainT, e_strain;
1480     tensor4 tmp, Tangent;
1481     int i_cnt = 0;
1482     bool flag;
1483
1484     flag = res == old_state;
1485     if (flag){
1486         old_state = new TSMASState;
1487         old_state->Assign(res);
1488     }
1489
1490     // Elastic prediction.
1491     // Check the yield surface.
1492     phi_0 = phi_k = res->Phi2_reverse();
1493     //Tangent = res->EffctvC(res->DStressDStrain());
1494     phi_0 = 1.0;
1495     if (phi_k > 0.0)
1496     {
1497         zero_tensor(&R);
1498         while ( (fabs(phi_k/phi_0) > PHI_EPS) || (norm(&R) > TOL))
1499         {
1500             res->Lambda2_reverse(&r);
1501             res->dPhi2_reverse_dstress(&v);
1502
1503             sub(&R, &old_state->strain_in, &res->strain_in);
1504             add_smul(&R, R, r, res->xi2 - old_state->xi2);
1505
1506             res->EffctvC(&Tangent);
1507
1508             // delta xi
1509             dxi_k = (phi_k - contract(&v, contract42(&tmp1, &Tangent, &R)))/
1510                 (-contract(&v, contract42(&tmp2, &Tangent, &v)) - res->dPhi2_reverse_dxi2());
1511             // delta stress
1512             add_smul(&dStress, tmp1, tmp2, -dxi_k);
1513             negate_tensor(&dStress);
1514
1515             contract42(&tmp1, &(smat->dS), &res->stress);
1516             add_smul(&tmp1, tmp1, smat->dAlpha, new_T - smat->T0);
1517             contract42(&dStrainT, res->EffctvS(&tmp), &dStress);
1518             add_smul(&dStrainT, dStrainT, tmp1, dxi_k);
1519             negate_tensor(&dStrainT);
1520
1521             res->xi2 += dxi_k;
1522             add(&res->strain_in, &res->strain_in, &dStrainT);
1523             add(&res->stress, &res->stress, &dStress);
1524             contract42(res->Stress(), res->EffctvC(&tmp), res->ElasticStrain(&e_strain));
1525
1526             phi_k = res->Phi2_reverse();
1527
1528             if (++i_cnt > MAX_ITER_COUNT){
1529                 interpolate(&tmp1, old_state->Strain(), new_strain, 0.5);
1530                 SimoCPP_reverse(res, &tmp1, (new_T + *(old_state->Temperature()))/2.0, old_state);
1531                 return SimoCPP_reverse(res, new_strain, new_T, res);
1532             }
1533         }
1534     }
1535     if (flag) delete old_state;
1536     return res;
1537 }
1538
1539
1540 bool TSMASState::is_A2Mt_possible(){
1541     return !((fabs(c1() - 1.0) < TOL) || (fabs(c3()) < TOL));
1542 }
1543
1544 bool TSMASState::is_A2Md_possible(){
1545     return !((fabs(c2() - 1.0) < TOL) || (fabs(c3()) < TOL));
1546 }
1547
1548 bool TSMASState::is_Md2A_possible(){
1549     return !((fabs(c3() - 1.0) < TOL) || (fabs(c2()) < TOL));

```

```

1550 }
1551
1552 bool TSMASState::is_Mt2A_possible(){
1553     return !((fabs(c3() - 1.0) < TOL) || (fabs(c1()) < TOL));
1554 }
1555
1556 bool TSMASState::is_Mt2Md_possible(){
1557     return !((fabs(c2() - 1.0) < TOL) || (fabs(c1()) < TOL));
1558 }
1559
1560 bool does_A2Md_happen(TSMASState* old_state, TSMASState* trial){
1561     double phi_new, phi_old;
1562     bool res = false;
1563
1564     if (old_state->is_A2Md_possible() ) {
1565         phi_new = trial->Phi2_forward();
1566         phi_old = old_state->Phi2_forward();
1567
1568         res = ((phi_new > phi_old + PHI_EPS) && (phi_new > PHI_EPS));
1569     }
1570
1571     return res;
1572 }
1573
1574
1575 bool does_Md2A_happen(TSMASState* old_state, TSMASState* trial){
1576     double phi_new, phi_old;
1577     bool res = false;
1578
1579     if (old_state->is_Md2A_possible() ) {
1580         phi_new = trial->Phi2_reverse();
1581         phi_old = old_state->Phi2_reverse();
1582
1583         res = ((phi_new > phi_old + PHI_EPS) && (phi_new > PHI_EPS));
1584     }
1585
1586     return res;
1587 }
1588
1589 bool does_Mt2Md_happen(TSMASState* old_state, TSMASState* trial){
1590
1591     return old_state->is_Mt2Md_possible() && (trial->Phi3() > PHI_EPS);
1592 }
1593
1594
1595 bool does_A2Mt_happen(TSMASState* old_state, TSMASState* trial){
1596
1597     if (old_state->T > trial->T){
1598         return (old_state->is_A2Mt_possible()) && (trial->Phi1_forward() > PHI_EPS);
1599     }
1600     else return false;
1601 }
1602
1603
1604 /*
1605 bool does_Mt2A_happen(TSMASState* old_state, TSMASState* trial){
1606
1607     if (old_state->T < trial->T){
1608         return (old_state->is_Mt2A_possible()) && (trial->Phi1_reverse() > PHI_EPS);
1609     }
1610     else return false;
1611 } */
1612
1613 bool does_Mt2A_happen(TSMASState* old_state, TSMASState* trial){
1614     double phi_new, phi_old;
1615     bool res = false;
1616
1617     if (old_state->is_Mt2A_possible() ) {
1618         phi_new = trial->Phi1_reverse();
1619         phi_old = old_state->Phi1_reverse();
1620
1621         res = ((phi_new > phi_old + PHI_EPS) && (phi_new > PHI_EPS));
1622     }
1623
1624     return res;
1625 }
1626
1627
1628 TInelasticState *TSMASolver::OrtizPopov85(TInelasticState *res, tensor* new_strain,
1629     double new_T,
1630     TInelasticState* old_state,
1631     TYieldSurfaceProc YieldSurface, TStrainFlowProc GetR, TIntVarFlowProc GetH,
1632     TDPhiDStressProc GetDPhiDStress, TDPhiDqProc GetDPhiDq, int q_num){
1633
1634     tensor e_strain, tmp1, tmp2, v, r, old_r;
1635     tensor4 tgnt, C;
1636     double gamma, ksi, phi_0, curr_phi, h, old_h, tmp_h;

```

```

1637     int i_cnt = 0;
1638     bool flag;
1639
1640     flag = res == old_state;
1641     if (flag){
1642         old_state = new TSMASState;
1643         old_state->Assign(res);
1644     }
1645
1646     // Elastic prediction.
1647     //LinearLoad(res, new_strain, new_T, old_state);
1648
1649     if (res->DStressDStrain())
1650         res->EffctvC(res->DStressDStrain());
1651
1652     // Check the yield surface.
1653     phi_0 = curr_phi = (res->*YieldSurface)();
1654     if (curr_phi > TOL){
1655         gamma = 0.0;
1656         (old_state->*GetR)(&old_r);
1657         (old_state->*GetH)(&old_h);
1658         contract42(&tmp1, old_state->EffctvC(&C), &old_r);
1659         while (fabs(curr_phi/phi_0) > TOL)
1660             {
1661                 (res->*GetR)(&r);
1662                 (res->*GetH)(&h);
1663
1664                 interpolate(&r, &old_r, &r, falpha);
1665                 h = (1.0 - falpha)*old_h + falpha*h;
1666
1667                 assign_tensor(&tmp2, &tmp1);
1668                 tmp_h = h;
1669
1670                 (res->*GetDPhiDStress)(&v);
1671                 ksi = (res->*GetDPhiDq)();
1672
1673                 contract42(&tmp1, res->EffctvC(&C), &r);
1674                 gamma = (gamma*(ksi*tmp_h - contract(&v, &tmp2))-curr_phi)/(ksi*h-contract(&v, &tmp1));
1675                 // gamma = gamma-curr_phi/(ksi*h-contract(&v, &tmp1));
1676                 add_smul((res->InelasticStrain(NULL, q_num)),
1677                     *(old_state->InelasticStrain(NULL, q_num)), r, gamma);
1678                 *(res->InternalVar(q_num)) = *(old_state->InternalVar(q_num)) + gamma*h;
1679                 res->ElasticStrain(&e_strain);
1680                 contract42(res->Stress(), res->EffctvC(&C), &e_strain);
1681
1682                 curr_phi = (res->*YieldSurface)();
1683                 if (++i_cnt > MAX_ITER_COUNT){
1684                     //cerr << "TSMASolver::OrtizPopov85 : diverging, giving up.";
1685                     interpolate(&tmp1, old_state->Strain(), new_strain, 0.5);
1686                     OrtizPopov85(res, &tmp1, (new_T + *(old_state->Temperature()))/2.0, old_state, YieldSurface, GetR,
1687                         GetH, GetDPhiDStress, GetDPhiDq, q_num);
1688                     return OrtizPopov85(res, new_strain, new_T, res, YieldSurface, GetR, GetH, GetDPhiDStress, GetDPhiDq,
1689                         q_num);
1690                 }
1691             }
1692     }
1693     if (res->DStressDStrain()){
1694         res->EffctvC(&C);
1695         contract42(&tmp1, &C, (res->*GetR)(&r));
1696         contract42(&tmp2, &C, (res->*GetDPhiDStress)(&r));
1697         ksi = contract(&r, &tmp1) - h*(res->*GetDPhiDq)();
1698         scalar_mul(&tmp2, &tmp2, 1.0/ksi);
1699         prod22(&tgtnt, &tmp1, &tmp2);
1700         sub4(res->DStressDStrain(), res->DStressDStrain(), &tgtnt);
1701     }
1702     if (flag) delete old_state;
1703     return res;
1704 }
1705
1706 void TSMASState::restore_phi3(){
1707     TSMAMaterial *smat = (TSMAMaterial*)GetMaterial();
1708     double new_c1, A, sH;
1709     tensor tmp;
1710
1711     Lambda3(&tmp);
1712     sH = contract(&tmp, &stress);
1713     A = smat->Y3-sH+smat->rho*smat->b3*xi3;
1714     new_c1 = -c2()*(A+smat->rho*smat->b2M)/(A-smat->rho*smat->b1M);
1715     if (new_c1 < EPS) new_c1 = 0.0;
1716     if (new_c1 > 1.0) new_c1 = 1.0;
1717     xi1 = new_c1 + xi3 - smat->c01;
1718 }
1719
1720 TSMASState *TSMASolver::SelfAccom_forward(TSMASState *res, tensor* new_strain,
1721     double new_T, TSMASState* old_state){

```

```

1722 TSMAMaterial* smat = (TSMAMaterial*)res->GetMaterial();
1723 int i_cnt = 0;
1724 double phi_0, phi_k;
1725
1726 SimoCPP(res, new_strain, new_T, old_state, PT_SA_FORWARD);
1727 /* phi_0 = phi_k = res->Phi1_forward();
1728 if (phi_0 > PHI_EPS){
1729     trial.Assign(old_state);
1730     do{
1731         res->xi1 = res->xi1 - phi_k/res->dPhi1_forward_dxi1();
1732         trial.xi1 = res->xi1;
1733         LinearLoad(res, new_strain, new_T, &trial);
1734         i_cnt++;
1735         if (i_cnt > MAX_ITER_COUNT)
1736             throw ESMAError("SelfAccom_forward: failed to converge");
1737         phi_k = res->Phi1_forward();
1738     } while (fabs(phi_k/phi_0) > TOL);
1739 }
1740 */
1741 if ((res->c1() > 1.0) || (res->c3() < 0.0)){
1742     res->xi1 = min(res->xi1, min(1.0 + res->xi3 - smat->c01, smat->c03 - res->xi2));
1743     //trial.xi1 = res->xi1;
1744     //LinearLoad(res, new_strain, new_T, &trial);
1745     LinearLoad(res, new_strain, new_T, res);
1746 }
1747 if (res->xi1 < old_state->xi1){
1748     throw ESMAError("SelfAccom_forward: incorrect solution found - xi1 decreased");
1749 }
1750 return res;
1751 }
1752
1753 TSMASolver::SelfAccom_reverse(TSMASolver *nstate, tensor* new_strain,
1754 double new_T, TSMASolver *ostate){
1755 int i_cnt = 0;
1756 double tmp, phi_0, phi_k;
1757 TSMAMaterial* smat = (TSMAMaterial*)nstate->GetMaterial();
1758
1759 SimoCPP(nstate, new_strain, new_T, ostate, PT_SA_REVERSE);
1760 if (nstate->xi1 > ostate->xi1){
1761     throw ESMAError("SelfAccom_reverse: incorrect solution found - xi1 increased");
1762 }
1763 if (nstate->xi1 < nstate->min_xi1()){
1764     Complete_Mt2A(nstate, new_strain, new_T, ostate);
1765     // nstate->xi1 = nstate->min_xi1();
1766     if (nstate->Phi3() > PHI_EPS){
1767         // Along the path we must have detwinned as well
1768         LinearLoad(nstate, new_strain, new_T, ostate);
1769         SimoCPP2(nstate, new_strain, new_T, ostate, PT_SA_REVERSE, PT_REORIENT);
1770         RestoreConsistency_Mt2A_Mt2Md(nstate, new_strain, new_T, ostate);
1771     }
1772     else
1773         LinearLoad(nstate, new_strain, new_T, nstate);
1774 }
1775 else if (does_Mt2Md_happen(ostate, nstate)){
1776     // Along the path we must have detwinned as well
1777     LinearLoad(nstate, new_strain, new_T, ostate);
1778     SimoCPP2(nstate, new_strain, new_T, ostate, PT_SA_REVERSE, PT_REORIENT);
1779     if (nstate->xi1 > ostate->xi1){
1780         throw ESMAError("SelfAccom_reverse: incorrect solution found - xi1 increased");
1781     }
1782     if (nstate->xi3 < ostate->xi3){
1783         throw ESMAError("SelfAccom_reverse: incorrect solution found - xi3 decreased");
1784     }
1785     RestoreConsistency_Mt2A_Mt2Md(nstate, new_strain, new_T, ostate);
1786 }
1787 if (nstate->xi1 > ostate->xi1){
1788     throw ESMAError("SelfAccom_reverse: incorrect solution found - xi1 increased");
1789 }
1790 return nstate;
1791 }
1792
1793 tensor* TSMASolver::HookeResidual(tensor* res){
1794     tensor tmp;
1795     tensor4 S;
1796
1797     return sub(res, contract42(res, EffctvS(&S), Stress()), ElasticStrain(&tmp));
1798 }
1799
1800 tensor* TSMASolver::dHookeResidual_dPT(tensor* res, int pt_indx){
1801     TSMAMaterial *smat = (TSMAMaterial*)GetMaterial();
1802     tensor tmp;
1803
1804     if (pt_indx != PT_REORIENT){
1805         contract42(res, &smat->dS, Stress());
1806         add_smul(res, *res, smat->dAlpha, *Temperature()-smat->T0);
1807     }

```

```

1809     }
1810     else{
1811         zero_tensor(res);
1812     }
1813
1814     // add(res, res, Lambda(&tmp, pt_indx));
1815     // This would be an optimization:
1816     switch (pt_indx){
1817         case PT_SA_FORWARD: return res;
1818         case PT_SA_REVERSE: return res;
1819         case PT_SIM_FORWARD: return add(res, res, Lambda2_forward(&tmp));
1820         case PT_SIM_REVERSE: return add(res, res, Lambda2_reverse(&tmp));
1821         case PT_REORIENT: return add(res, res, Lambda3(&tmp));
1822         default: cerr << "TSMASState::dHookeResidual_dxi - Inappropriate index";
1823                 return NULL;
1824     }
1825 }
1826
1827
1828 tensor4* TSMASState::dHookeResidual_dstress(tensor4* res, int pt_indx, TSMASState *old_state){
1829     tensor4 tmp;
1830
1831     EffctvS(res);
1832
1833     if ((pt_indx == PT_SIM_FORWARD)){
1834         add_smul4(res, *res, *(dLambda2_forward_dstress(&tmp)), xi2 - old_state->xi2);
1835     }
1836     if ((pt_indx == PT_SIM_REVERSE)){
1837         add_smul4(res, *res, *(dLambda2_reverse_dstress(&tmp)), xi2 - old_state->xi2);
1838     }
1839     if ((pt_indx == PT_REORIENT)){
1840         add_smul4(res, *res, *(dLambda3_dstress(&tmp)), xi3 - old_state->xi3);
1841     }
1842     //return add_smul4(res, *res, *(dLambda_dstress(&tmp, pt_indx)), c2() - old_state->c2());
1843     return res;
1844 }
1845
1846 tensor4* TSMASState::dHookeResidual_dstress2(tensor4* res, int pt_indx1, int pt_indx2, TSMASState *old_state){
1847     tensor4 tmp;
1848
1849     EffctvS(res);
1850
1851     if ((pt_indx1 == PT_SIM_FORWARD) || (pt_indx2 == PT_SIM_FORWARD)){
1852         add_smul4(res, *res, *(dLambda2_forward_dstress(&tmp)), xi2 - old_state->xi2);
1853     }
1854     if ((pt_indx1 == PT_SIM_REVERSE) || (pt_indx2 == PT_SIM_REVERSE)){
1855         add_smul4(res, *res, *(dLambda2_reverse_dstress(&tmp)), xi2 - old_state->xi2);
1856     }
1857     if ((pt_indx1 == PT_REORIENT) || (pt_indx2 == PT_REORIENT)){
1858         add_smul4(res, *res, *(dLambda3_dstress(&tmp)), xi3 - old_state->xi3);
1859     }
1860     return res;
1861 }
1862
1863 TSMASState *TSMASolver::SIM(TSMASState *nstate, tensor* new_strain, double new_T, TSMASState* ostate){
1864
1865     /* return (TSMASState *)OrtizPopov85(res, new_strain, new_T, old_state,
1866                                         (TYieldSurfaceProc)TSMASState::Phi2_forward,
1867                                         (TStrainFlowProc)TSMASState::Lambda2_forward,
1868                                         (TIntVarFlowProc)TSMASState::get_h2,
1869                                         (TDPhiDStressProc)TSMASState::dPhi2_forward_stress,
1870                                         (TDPHiDqProc)TSMASState::dPhi2_forward_dxi2, 2);
1871     */
1872     (TSMASState *)SimoCPP_forward(nstate, new_strain, new_T, ostate);
1873     // (TSMASState *)SimoCPP(res, new_strain, new_T, old_state, PT_SIM_FORWARD);
1874     if (nstate->xi2 < ostate->xi2){
1875         throw ESMAError("TSMASolver::SI_forward failed: xi2 decreased.");
1876     }
1877     return nstate;
1878 }
1879
1880 TSMASState *TSMASolver::SI_reverse(TSMASState *nstate, tensor* new_strain, double new_T, TSMASState* ostate){
1881
1882     // (TSMASState *)SimoCPP_reverse(nstate, new_strain, new_T, ostate);
1883     (TSMASState *)SimoCPP(nstate, new_strain, new_T, ostate, PT_SIM_REVERSE);
1884     if (nstate->xi2 > ostate->xi2){
1885         throw ESMAError("TSMASolver::SI_reverse failed: xi2 increased.");
1886     }
1887     return nstate;
1888 }
1889
1890 TSMASState *TSMASolver::M2A(TSMASState *nstate, tensor* new_strain, double new_T, TSMASState* ostate)
1891 {
1892     TSMAMaterial* smat = SMAT(ostate->GetMaterial());
1893
1894     SimoCPP2(nstate, new_strain, new_T, ostate, PT_SA_REVERSE, PT_SIM_REVERSE);
1895     if (nstate->xi1 > ostate->xi1){

```

```

1896     throw ESMAError("M2A: incorrect solution - xi1 has increased");
1897 }
1898 if (nstate->xi2 > ostate->xi2){
1899     nstate->Assign(ostate);
1900     LinearLoad(nstate, new_strain, new_T, ostate);
1901     SimoCPP(nstate, new_strain, new_T, ostate, PT_SA_REVERSE);
1902     //throw ESMAError("M2A: incorrect solution - xi2 has increased");
1903 }
1904 if ((nstate->c1() < 0.0) || (nstate->c3() > 1.0)){
1905     nstate->xi1 = nstate->min_xi1();
1906     nstate->xi2 = ostate->xi2;
1907     SimoCPP(nstate, new_strain, new_T, ostate, PT_SIM_REVERSE);
1908     if (nstate->xi2 > ostate->xi2){
1909         throw ESMAError("M2A: incorrect solution - xi1 set to minimum, xi2 has increased");
1910     }
1911 }
1912 return nstate;
1913 }
1914 }
1915 TSMASolver::A2M(TSMASolver *nstate, tensor *new_strain, double new_T, TSMASolver *ostate)
1916 {
1917     TSMASolver *smat = SMAT(ostate->GetMaterial());
1918
1919     SimoCPP2(nstate, new_strain, new_T, ostate, PT_SA_FORWARD, PT_SIM_FORWARD);
1920     if ((nstate->xi1 < ostate->xi1) && (nstate->xi2 < ostate->xi2)){
1921         throw ESMAError("TSMASolver::A2M - Algorithmic failure");
1922     }
1923     if (nstate->xi1 < ostate->xi1){
1924         TSMASolver prev;
1925         bool flag;
1926
1927         prev.Assign(ostate);
1928         LinearLoad(nstate, new_strain, new_T, &prev);
1929         SimoCPP(nstate, new_strain, new_T, &prev, PT_SIM_FORWARD);
1930         flag = RestoreConsistency_A2Md_xxx(nstate, new_strain, new_T, &prev);
1931         if (flag){
1932             prev.xi2 = nstate->max_xi2();
1933         }
1934         if (nstate->Phi_forward() > PHI_EPS)
1935             throw ESMAError("TSMASolver::A2M - Phi1_forrward is inconsistent");
1936         //if ((nstate->c1() > EPS) && (nstate->Phi3() > PHI_EPS))
1937         //    throw ESMAError("Error type 2.");
1938         if (flag) LinearLoad(nstate, new_strain, new_T, &prev);
1939         if ((nstate->c3() > 0.0) && (nstate->Phi_forward() > PHI_EPS))
1940             throw ESMAError("TSMASolver::A2M - Phi1_forrward is inconsistent");
1941         if ((nstate->c1() > EPS) && (nstate->Phi3() > PHI_EPS))
1942         {
1943             Detwin(nstate, new_strain, new_T, &prev);
1944             RestoreConsistency_xi3(nstate, new_strain, new_T, &prev);
1945         }
1946     }
1947 }
1948 if (nstate->xi2 < ostate->xi2){
1949     TSMASolver prev;
1950     bool flag;
1951
1952     prev.Assign(ostate);
1953     LinearLoad(nstate, new_strain, new_T, &prev);
1954     SimoCPP(nstate, new_strain, new_T, &prev, PT_SA_FORWARD);
1955     if (flag = RestoreConsistency_A2Mt(nstate, new_strain, new_T, &prev)){
1956         prev.xi1 = nstate->max_xi1();
1957     };
1958     if (nstate->Phi2_forward() > PHI_EPS)
1959         throw ESMAError("TSMASolver::A2M - Phi2_forrward is inconsistent");
1960     //if ((nstate->c1() > TOL) && (nstate->Phi3() > PHI_EPS))
1961     //    throw ESMAError("Error type 2.");
1962     if (flag) LinearLoad(nstate, new_strain, new_T, &prev);
1963     if ((nstate->c3() > 0.0) && (nstate->Phi2_forward() > PHI_EPS))
1964         throw ESMAError("TSMASolver::A2M - Phi2_forrward is inconsistent");
1965     if ((nstate->c1() > EPS) && (nstate->Phi3() > PHI_EPS))
1966     {
1967         cout << "XXX";
1968         // Detwin(nstate, new_strain, new_T, &prev);
1969         // RestoreConsistency_xi3(nstate, new_strain, new_T, &prev);
1970     };
1971 }
1972 if ((nstate->xi1 > nstate->max_xi1()) || (nstate->xi2 > nstate->max_xi2())){
1973     RestoreConsistency_A2M(nstate, new_strain, new_T, ostate);
1974 }
1975 if (nstate->xi1 < ostate->xi1){
1976     throw ESMAError("M2A: incorrect solution - xi1 has decreased");
1977 }
1978 if (nstate->xi2 < ostate->xi2){
1979     throw ESMAError("M2A: incorrect solution - xi2 has decreased");
1980 }
1981 return nstate;
1982 }

```

```

1983
1984
1985 TSMASState *TSMASolver::Mt2Md_and_Mt2A(TSMASState *res, tensor* new_strain, double new_T, TSMASState* ostate)
1986 {
1987     TSMAMaterial* smat = SMAT(ostate->GetMaterial());
1988     double lambda, intrpl_t;
1989     tensor inter_strain;
1990
1991     SimoCPP2(res, new_strain, new_T, ostate, PT_SA_REVERSE, PT_REORIENT);
1992     if (res->xi1 > ostate->xi1){
1993         throw ESMAError("Mt2Md_and_Mt2A: incorrect solution - xi1 has increased");
1994     }
1995     if (res->xi3 < ostate->xi3){
1996         throw ESMAError("Mt2Md_and_Mt2A: incorrect solution - xi3 has decreased");
1997     }
1998     // if ((res->c1() < 0.0) || (res->c3() > 1.0)){
1999     if (res->xi1 < res->min_xi1()){
2000         TSMASState* inter = new TSMASState();
2001
2002         inter->Assign(ostate);
2003         while (fabs(res->xi1 - res->min_xi1()) > TOL){
2004             lambda = (res->min_xi1() - inter->xi1)/(res->xi1-inter->xi1);
2005             interpolate(&inter_strain, &inter->strain, &res->strain, lambda);
2006             intrpl_t = inter->T*(1.0-lambda) + res->T*(lambda);
2007             LinearLoad(res, &inter_strain, intrpl_t, inter);
2008             SimoCPP2(res, &inter_strain, intrpl_t, inter, PT_SA_REVERSE, PT_REORIENT);
2009         }
2010         res->xi1 = res->min_xi1();
2011         inter->Assign(res);
2012         LinearLoad(res, new_strain, new_T, inter);
2013         if (does_Mt2Md_happen(inter, res)){
2014             SimoCPP(res, new_strain, new_T, inter, PT_REORIENT);
2015             if (res->xi3 < ostate->xi3){
2016                 throw ESMAError("Mt2Md_and_Mt2A: incorrect solution - xi1 set to minimum, xi3 has decreased");
2017             }
2018         }
2019         delete inter;
2020     }
2021     // }
2022     return res;
2023 }
2024
2025
2026
2027 TSMASState *TSMASolver::Detwin(TSMASState *res, tensor* new_strain, double new_T, TSMASState* old_state){
2028     TSMAMaterial* mtrl;
2029
2030     if (!old_state->is_Mt2Md_possible()){
2031         return (TSMASState *)LinearLoad(res, new_strain, new_T, old_state);
2032     }
2033     else{
2034         // return (TSMASState *)OrtizPopov85(res, new_strain, new_T, old_state,
2035         // (TYieldSurfaceProc)TSMASState::Phi3,
2036         // (TStrainFlowProc)TSMASState::Lambda3,
2037         // (TIntVarFlowProc)TSMASState::get_h3,
2038         // (TDPhiDStressProc)TSMASState::dPhi3_dstress,
2039         // (TDPhiDqProc)TSMASState::dPhi3_q, 3);
2040         return (TSMASState *)SimoCPP(res, new_strain, new_T, old_state, PT_REORIENT);
2041     }
2042 }
2043
2044 void print_tnsr2s(ostream *strm, char *name, tensor tnsr){
2045     (*strm) << name << ": \n";
2046     for (int i = 0; i < 3; i++){
2047         for (int j = 0; j < 3; j++){ (*strm) << tnsr[i][j] << " ";
2048             (*strm) << "\n";
2049         }
2050     }
2051     strm->flush();
2052 }
2053
2054 PInelasticState __cdecl TSMASolver::Load(PInelasticState res, tensor* new_strain,
2055 double new_T, PInelasticState old_state){
2056
2057     TSMAMaterial* smat = SMAT(old_state->GetMaterial());
2058     TSMASState *nstate, *ostate;
2059     tensor tmp, estress;
2060     double intrpl_t, lambda;
2061     bool A2Md, Md2A, A2Mt, Mt2A, Mt2Md;
2062
2063     nstate = (TSMASState*)res;
2064     ostate = new TSMASState();
2065     ostate->Assign(old_state);
2066
2067     if (!nstate) {
2068         nstate = new TSMASState(smat);
2069     }

```

```

2070     else
2071         nstate->SetMaterial(smat);
2072
2073     LinearLoad(nstate, new_strain, new_T, ostate);
2074
2075     A2Mt = does_A2Mt_happen(ostate, nstate);
2076     A2Md = does_A2Md_happen(ostate, nstate);
2077     Md2A = does_Md2A_happen(ostate, nstate);
2078     Mt2A = does_Mt2A_happen(ostate, nstate);
2079     Mt2Md = does_Mt2Md_happen(ostate, nstate);
2080
2081     if (A2Mt && A2Md){
2082     {
2083         A2M(nstate, new_strain, new_T, ostate);
2084         A2Mt = false;
2085         A2Md = false;
2086         Mt2Md = (nstate->c1() > EPS) && (nstate->Phi3() > PHI_EPS);
2087         if (Mt2Md)
2088         {
2089             LinearLoad(nstate, new_strain, new_T, ostate);
2090             SIM(nstate, new_strain, new_T, ostate);
2091             RestoreConsistency_A2Md_xxx(nstate, new_strain, new_T, ostate);
2092             if ((nstate->c1() > EPS) && (nstate->Phi3() > PHI_EPS)){
2093                 LinearLoad(nstate, new_strain, new_T, ostate);
2094                 SimoCPP2(nstate, new_strain, new_T, ostate, PT_SA_FORWARD, PT_REORIENT);
2095             }
2096
2097             if (nstate->Phi1_forward() > PHI_EPS){
2098                 LinearLoad(nstate, new_strain, new_T, ostate);
2099                 SimoCPP2(nstate, new_strain, new_T, ostate, PT_SA_FORWARD, PT_REORIENT);
2100             }
2101
2102             LinearLoad(nstate, new_strain, new_T, nstate);
2103             Mt2Md = false;
2104         }
2105     }
2106 }
2107
2108
2109 if (Mt2A && Md2A){
2110     M2A(nstate, new_strain, new_T, ostate);
2111     Mt2A = false;
2112     Md2A = false;
2113 }
2114
2115 if (Mt2A && Mt2Md){ // Since detwinning depletes c1() it also moves phi1 away. It may happen that after
2116 detwinning the stress state no longer violates phi1_reverse
2117     Detwin(nstate, new_strain, new_T, ostate);
2118     if (does_Mt2A_happen(ostate, nstate)){
2119         LinearLoad(nstate, new_strain, new_T, ostate);
2120         Mt2Md_and_Mt2A(nstate, new_strain, new_T, ostate);
2121     } //else RestoreConsistency_xi3(nstate, new_strain, new_T, ostate);
2122     Mt2A = false;
2123     Mt2Md = false;
2124 }
2125
2126 if (A2Mt){
2127     SelfAccom_forward(nstate, new_strain, new_T, ostate);
2128 }
2129
2130 if (Mt2A){
2131     SelfAccom_reverse(nstate, new_strain, new_T, ostate);
2132     //if (does_Mt2Md_happen(ostate, nstate)){
2133     // LinearLoad(nstate, new_strain, new_T, ostate);
2134     // Mt2Md_and_Mt2A(nstate, new_strain, new_T, ostate);
2135     //}
2136 }
2137
2138 if (A2Md){
2139     if (!SIM(nstate, new_strain, new_T, ostate)) return NULL;
2140     RestoreConsistency_A2Md(nstate, new_strain, new_T, ostate);
2141 }
2142
2143 if (Md2A){
2144     S1_reverse(nstate, new_strain, new_T, ostate);
2145     RestoreConsistency_Md2A(nstate, new_strain, new_T, ostate);
2146 }
2147
2148 if (Mt2Md) {
2149     Detwin(nstate, new_strain, new_T, ostate);
2150     if (does_Mt2A_happen(ostate, nstate)){
2151         Mt2Md_and_Mt2A(nstate, new_strain, new_T, ostate);
2152     }
2153     RestoreConsistency_xi3(nstate, new_strain, new_T, ostate);
2154 }
2155

```



```

2156
2157
2158     if (!nstate->check_consistency(ostate)){
2159         // throw ESMABError("Inconsistent state. Look at err.txt file.");
2160         ofstream fout;
2161         char buf[400];
2162         fout.open("err.txt", ios::app);
2163         fout << "Inconsistent state: \n\n" << ostate->GetAsString(buf) << "\n\n";
2164         fout << "Error while trying to load with strain:";
2165         print_tnsr2s(&fout, "new_strain", *new_strain);
2166         fout << "\n\n and T = " << new_T;
2167         fout.close();
2168     }
2169     delete ostate;
2170
2171     // allocated_mem = TotalAllocated();
2172
2173     return nstate;
2174 }
2175
2176
2177 bool TSMASolver::RestoreConsistency_Md2A(TSMASState *nstate, tensor* new_strain,
2178     double new_T, TSMASState *ostate){
2179
2180     if (nstate->xi2 < nstate->min_xi2()) {
2181         nstate->Assign(ostate);
2182         nstate->xi2 = nstate->min_xi2();
2183         zero_tensor(&nstate->strain_in);
2184         LinearLoad(nstate, new_strain, new_T, nstate);
2185         return true;
2186     }
2187     else
2188         return false;
2189 }
2190
2191 void TSMASolver::RestoreConsistency_xi3(TSMASState *res, tensor* new_strain,
2192     double new_T, TSMASState *ostate){
2193
2194     tensor tmp;
2195     double intrpl_t, lambda;
2196     TSMASState *prev;
2197
2198     if (res->xi3 > res->max_xi3()){
2199         prev = new TSMASState();
2200         prev->Assign(ostate);
2201         prev->xi1 = res->xi1;
2202         prev->xi2 = res->xi2;
2203         while (fabs(res->xi3 - res->max_xi3()) > TOL){
2204             lambda = (res->max_xi3() - prev->xi3)/(res->xi3-prev->xi3);
2205             interpolate(&tmp, &prev->strain, &res->strain, lambda);
2206             intrpl_t = prev->T*(1.0-lambda) + res->T*(lambda);
2207             LinearLoad(res, &tmp, intrpl_t, prev);
2208             SimoCPP(res, &tmp, intrpl_t, prev, PT_REORIENT);
2209             //Detwin(res, &tmp, intrpl_t, prev);
2210         }
2211         res->xi3 = res->max_xi3();
2212         prev->Assign(res);
2213         LinearLoad(res, new_strain, new_T, prev);
2214         delete prev;
2215     }
2216 }
2217
2218
2219 bool TSMASolver::RestoreConsistency_A2Mt(TSMASState *nstate, tensor* new_strain,
2220     double new_T, TSMASState *ostate){
2221
2222     bool flag = nstate->xi1 > nstate->max_xi1();
2223
2224     if (flag)
2225     {
2226         double lambda, intrpl_t;
2227         tensor tmp;
2228         TSMASState prev;
2229
2230         prev.Assign(ostate);
2231         prev.xi2 = nstate->xi2;
2232         prev.xi3 = nstate->xi3;
2233
2234         while (fabs(nstate->xi1 - nstate->max_xi1()) > TOL){
2235             lambda = (nstate->max_xi1() - prev.xi1)/(nstate->xi1-prev.xi1);
2236             interpolate(&tmp, &prev.strain, &nstate->strain, lambda);
2237             intrpl_t = prev.T*(1.0-lambda) + nstate->T*(lambda);
2238             LinearLoad(nstate, &tmp, intrpl_t, &prev);
2239             SimoCPP(nstate, &tmp, intrpl_t, ostate, PT_SA_FORWARD);
2240         }
2241     }
2242     return flag;

```

```

2243
2244 /*
2245 TSMAMaterial *smat = (TSMAMaterial*)nstate->GetMaterial();
2246 if (nstate->xi1 > nstate->max_xi1()){
2247     nstate->xi1 = min(
2248         nstate->xi1, min(
2249             1.0 + nstate->xi3 - smat->c01,
2250             smat->c03 - nstate->xi2));
2251     LinearLoad(nstate, new_strain, new_T, nstate);
2252 }
2253 */
2254 }
2255
2256 bool TSMASolver::RestoreConsistency_A2Md_xxx(TSMASState *res, tensor* new_strain,
2257 double new_T, TSMASState *ostate){
2258
2259     bool flag = res->xi2 > res->max_xi2();
2260
2261     if (flag){
2262         tensor tmp;
2263         double intrpl_t, lambda;
2264         TSMASState *prev;
2265
2266         prev = new TSMASState();
2267         prev->Assign(ostate);
2268         prev->xi1 = res->xi1;
2269         prev->xi3 = res->xi3;
2270
2271         while (fabs(res->xi2 - res->max_xi2()) > TOL){
2272             lambda = (res->max_xi2() - prev->xi2)/(res->xi2-prev->xi2);
2273             interpolate(&tmp, &prev->strain, &res->strain, lambda);
2274             intrpl_t = prev->T*(1.0-lambda) + res->T*(lambda);
2275             LinearLoad(res, &tmp, intrpl_t, prev);
2276             SIM(res, &tmp, intrpl_t, prev);
2277         }
2278         res->xi2 = res->max_xi2();
2279         delete prev;
2280     }
2281
2282     return flag;
2283 }
2284
2285 void TSMASolver::RestoreConsistency_A2Md(TSMASState *res, tensor* new_strain,
2286 double new_T, TSMASState *ostate){
2287
2288     tensor tmp;
2289     double intrpl_t, lambda;
2290     TSMASState *prev;
2291
2292     if (res->xi2 > res->max_xi2()){
2293         prev = new TSMASState();
2294         prev->Assign(ostate);
2295         prev->xi1 = res->xi1;
2296         prev->xi3 = res->xi3;
2297
2298         while (fabs(res->xi2 - res->max_xi2()) > TOL){
2299             lambda = (res->max_xi2() - prev->xi2)/(res->xi2-prev->xi2);
2300             interpolate(&tmp, &prev->strain, &res->strain, lambda);
2301             intrpl_t = prev->T*(1.0-lambda) + res->T*(lambda);
2302             LinearLoad(res, &tmp, intrpl_t, prev);
2303             SIM(res, &tmp, intrpl_t, prev);
2304         }
2305         res->xi2 = res->max_xi2();
2306         prev->Assign(res);
2307         LinearLoad(res, new_strain, new_T, prev);
2308         delete prev;
2309     }
2310 }
2311
2312 void TSMASolver::RestoreConsistency_A2M(TSMASState *nstate, tensor* new_strain,
2313 double new_T, TSMASState *ostate){
2314
2315     tensor tmp, intl;
2316     double intrpl_t, lambda, l_min, l_max, err;
2317
2318     if ((nstate->xi1 > nstate->max_xi1()) || (nstate->xi2 > nstate->max_xi2())){
2319
2320         TSMASState prev;
2321         double fin_T;
2322         tensor fin_strain;
2323
2324         prev.Assign(ostate);
2325         prev.xi3 = nstate->xi3;
2326
2327         lambda = 1.0;
2328         l_min = 0.0;
2329         l_max = 1.0;

```

```

2330
2331   fin_T = nstate->T;
2332   assign_tensor(&fin_strain, &(nstate->strain));
2333
2334   if (nstate->c3() < 0.0){
2335     while (fabs(nstate->c3()) > TOL){
2336       lambda = lambda*(-prev.c3()/(nstate->c3()-prev.c3()));
2337       if ((lambda > l_max) || (lambda < l_min)) {
2338         lambda = 0.5*(l_min + l_max);
2339       }
2340       interpolate(&tmp, &prev.strain, &fin_strain, lambda);
2341       intrpl_t = prev.T*(1.0-lambda) + fin_T*(lambda);
2342       LinearLoad(nstate, &tmp, intrpl_t, &prev);
2343       SimoCPP2(nstate, &tmp, intrpl_t, ostate, PT_SA_FORWARD, PT_SIM_FORWARD);
2344       if (nstate->c3() < 0.0)
2345         l_max = lambda;
2346       else
2347         l_min = lambda;
2348     }
2349
2350     if (nstate->xi1 < prev.xi1){
2351       bool flag;
2352       nstate->Assign(&prev);
2353       LinearLoad(nstate, new_strain, new_T, &prev);
2354       SimoCPP(nstate, new_strain, new_T, &prev, PT_SIM_FORWARD);
2355       flag = RestoreConsistency_A2Md_xxx(nstate, new_strain, new_T, &prev);
2356       if (nstate->Phi1_forward() > PHI_EPS)
2357         throw ESMAError("TSMASolver::A2M - Phi1_forrward is inconsistent");
2358       if (flag){
2359         prev.Assign(nstate);
2360         lambda = nstate->c3();
2361         prev.xi2 = nstate->max_xi2();
2362         LinearLoad(nstate, new_strain, new_T, &prev);
2363       }
2364       if (nstate->xi1 > nstate->max_xi1() + TOL){
2365         throw ESMAError("TSMASolver::RestoreConsistency_A2M: Feature not yet implemented.");
2366       }
2367       if (nstate->xi2 > nstate->max_xi2() + TOL){
2368         throw ESMAError("TSMASolver::RestoreConsistency_A2M: Feature not yet implemented.");
2369       }
2370       return;
2371     }
2372
2373     if (nstate->xi2 < prev.xi2){
2374       bool flag;
2375       nstate->Assign(&prev);
2376       LinearLoad(nstate, new_strain, new_T, &prev);
2377       SimoCPP(nstate, new_strain, new_T, &prev, PT_SA_FORWARD);
2378       flag = RestoreConsistency_A2Mt(nstate, new_strain, new_T, &prev);
2379       if (nstate->Phi2_forward() > PHI_EPS)
2380         throw ESMAError("TSMASolver::A2M - Phi2_forrward is inconsistent");
2381       if (flag){
2382         prev.Assign(nstate);
2383         prev.xi1 = nstate->max_xi1();
2384         LinearLoad(nstate, new_strain, new_T, &prev);
2385       }
2386
2387       if (nstate->xi1 > nstate->max_xi1() + TOL){
2388         throw ESMAError("TSMASolver::RestoreConsistency_A2M: Feature not yet implemented.");
2389       }
2390       if (nstate->xi2 > nstate->max_xi2() + TOL){
2391         throw ESMAError("TSMASolver::RestoreConsistency_A2M: Feature not yet implemented.");
2392       }
2393       return;
2394     }
2395
2396     if (nstate->xi1 > nstate->max_xi1() + TOL){
2397       throw ESMAError("TSMASolver::RestoreConsistency_A2M: Feature not yet implemented.");
2398     }
2399     if (nstate->xi2 > nstate->max_xi2() + TOL){
2400       throw ESMAError("TSMASolver::RestoreConsistency_A2M: Feature not yet implemented.");
2401     }
2402
2403     // At this point there is no more austenite.
2404     prev.Assign(nstate);
2405     lambda = nstate->c3();
2406     prev.xi1 = nstate->xi1 + 0.5*lambda;
2407     prev.xi2 = nstate->xi2 + 0.5*lambda;
2408
2409     if ((nstate->c1() > EPS) && (nstate->Phi3() > PHI_EPS))
2410       throw ESMAError("Error type 23");
2411
2412     LinearLoad(nstate, new_strain, new_T, &prev);
2413
2414     if ((nstate->c1() > EPS) && (nstate->Phi3() > PHI_EPS))
2415     {

```

```

2417         Detwin(nstate, new_strain, new_T, &prev);
2418         RestoreConsistency_xi3(nstate, new_strain, new_T, &prev);
2419     }
2420 }
2421 if (nstate->xi1 > nstate->max_xi1() + TOL){
2422     throw ESMAError("TSMASolver::RestoreConsistency_A2M: Feature not yet implemented.");
2423 }
2424 if (nstate->xi2 > nstate->max_xi2() + TOL){
2425     throw ESMAError("TSMASolver::RestoreConsistency_A2M: Feature not yet implemented.");
2426 }
2427 }
2428 return;
2429
2430 // First try to restore consistency by changing xi1
2431 while (fabs(nstate->xi1 - nstate->max_xi1()) > TOL){
2432     lambda = (nstate->max_xi1() - prev.xi1)/(nstate->xi1-prev.xi1);
2433     interpolate(&tmp, &prev.strain, &nstate->strain, lambda);
2434     intrpl_t = prev.T*(1.0-lambda) + nstate->T*(lambda);
2435     LinearLoad(nstate, &tmp, intrpl_t, &prev);
2436     SimoCPP2(nstate, &tmp, intrpl_t, ostate, PT_SA_FORWARD, PT_SIM_FORWARD);
2437 }
2438 if (nstate->xi2 > nstate->max_xi2() + TOL){
2439     // Repeat the exercise for xi2
2440     throw ESMAError("Featire not yet debugged.");
2441     LinearLoad(nstate, new_strain, new_T, ostate);
2442     SimoCPP2(nstate, new_strain, new_T, ostate, PT_SA_FORWARD, PT_SIM_FORWARD);
2443     prev.Assign(ostate);
2444     prev.xi3 = nstate->xi3;
2445     while (fabs(nstate->xi2 - nstate->max_xi2()) > TOL){
2446         lambda = (nstate->max_xi2() - prev.xi2)/(nstate->xi2-prev.xi2);
2447         interpolate(&tmp, &prev.strain, &nstate->strain, lambda);
2448         intrpl_t = prev.T*(1.0-lambda) + nstate->T*(lambda);
2449         LinearLoad(nstate, &tmp, intrpl_t, &prev);
2450         SimoCPP2(nstate, &tmp, intrpl_t, ostate, PT_SA_FORWARD, PT_SIM_FORWARD);
2451     }
2452     if (nstate->xi1 > nstate->max_xi1() + TOL)
2453         throw ESMAError("TSMASolver::RestoreConsistency_A2M_xi1: Failed to restore consistency.");
2454     nstate->xi2 = nstate->max_xi2();
2455     prev.Assign(nstate);
2456     LinearLoad(nstate, new_strain, new_T, &prev);
2457     SimoCPP(nstate, new_strain, new_T, &prev, PT_SA_FORWARD);
2458     RestoreConsistency_A2Mt(nstate, new_strain, new_T, &prev);
2459 }
2460 else{
2461     nstate->xi1 = nstate->max_xi1();
2462     prev.Assign(nstate);
2463     LinearLoad(nstate, new_strain, new_T, &prev);
2464     SimoCPP(nstate, new_strain, new_T, &prev, PT_SIM_FORWARD);
2465     RestoreConsistency_A2Md(nstate, new_strain, new_T, &prev);
2466 }
2467 }
2468 }
2469
2470 void TSMASolver::Complete_Mt2A(TSMASState *nstate, tensor* new_strain,
2471 double new_T, TSMASState *ostate){
2472
2473     tensor tmp;
2474     double intrpl_t, lambda;
2475
2476     if (nstate->xi1 < nstate->min_xi1()){
2477         TSMASState prev = TSMASState();
2478         TSMASState test = TSMASState();
2479         test.Assign(nstate);
2480         test.xi1 = test.min_xi1();
2481         prev.Assign(ostate);
2482         prev.xi2 = nstate->xi2;
2483         prev.xi3 = nstate->xi3;
2484
2485         while (fabs(nstate->xi1 - nstate->min_xi1()) > TOL){
2486             lambda = (nstate->min_xi1() - prev.xi1)/(nstate->xi1-prev.xi1);
2487             interpolate(&tmp, &prev.strain, &nstate->strain, lambda);
2488             intrpl_t = prev.T*(1.0-lambda) + nstate->T*(lambda);
2489             LinearLoad(nstate, &tmp, intrpl_t, &prev);
2490             SimoCPP(nstate, &tmp, intrpl_t, ostate, PT_SA_REVERSE);
2491         }
2492         nstate->xi1 = nstate->min_xi1();
2493         // if (test.Phi3() * nstate->Phi3() < 0.0){
2494         //     cout << test.Phi3() << nstate->Phi3();
2495         // }
2496     }
2497 }
2498
2499 void TSMASolver::RestoreConsistency_Mt2A_Mt2Md(TSMASState *nstate, tensor* new_strain,
2500 double new_T, TSMASState *ostate){
2501
2502     tensor tmp;
2503     double intrpl_t, lambda, err;

```

```

2504 TSMASState *prev;
2505
2506 if ((nstate->xi1 < nstate->min_xi1()) || (nstate->xi3 > nstate->max_xi3())){
2507     prev = new TSMASState();
2508     prev->Assign(ostate);
2509     prev->xi2 = nstate->xi2;
2510
2511     // First try to restore consistency by changing xi1
2512     while (fabs(nstate->xi1 - nstate->min_xi1()) > TOL){
2513         lambda = (nstate->min_xi1() - prev->xi1)/(nstate->xi1-prev->xi1);
2514         if (lambda < 0){
2515             lambda = 0.5;
2516         }
2517         interpolate(&tmp, &prev->strain, &nstate->strain, lambda);
2518         intrpl_t = prev->T*(1.0-lambda) + nstate->T*(lambda);
2519         LinearLoad(nstate, &tmp, intrpl_t, prev);
2520         SimoCPP2(nstate, &tmp, intrpl_t, ostate, PT_SA_REVERSE, PT_REORIENT);
2521     }
2522     if (nstate->xi3 > nstate->max_xi3() + TOL){
2523         // Repeat the exercise for xi3
2524         throw ESMAError("Feature not yet debugged.");
2525         LinearLoad(nstate, new_strain, new_T, ostate);
2526         SimoCPP2(nstate, new_strain, new_T, ostate, PT_SA_REVERSE, PT_REORIENT);
2527         prev->Assign(ostate);
2528         prev->xi2 = nstate->xi2;
2529         while (fabs(nstate->xi3 - nstate->max_xi3()) > TOL){
2530             lambda = (nstate->max_xi3() - prev->xi3)/(nstate->xi3-prev->xi3);
2531             interpolate(&tmp, &prev->strain, &nstate->strain, lambda);
2532             intrpl_t = prev->T*(1.0-lambda) + nstate->T*(lambda);
2533             LinearLoad(nstate, &tmp, intrpl_t, prev);
2534             SimoCPP2(nstate, &tmp, intrpl_t, ostate, PT_SA_REVERSE, PT_REORIENT);
2535         }
2536         if (nstate->xi1 < nstate->min_xi1() - TOL)
2537             throw ESMAError("TSMASolver::RestoreConsistency_Mt2A_Mt2Md: Failed to restore consistency.");
2538         nstate->xi3 = nstate->max_xi3();
2539         prev->Assign(nstate);
2540         LinearLoad(nstate, new_strain, new_T, prev);
2541         if (does_Mt2A_happen(prev, nstate)){
2542             SimoCPP(nstate, new_strain, new_T, prev, PT_SA_REVERSE);
2543             RestoreConsistency_xi3(nstate, new_strain, new_T, prev);
2544         }
2545     }
2546     else{
2547         nstate->xi1 = nstate->min_xi1();
2548         prev->Assign(nstate);
2549         LinearLoad(nstate, new_strain, new_T, prev);
2550         if (does_Mt2Md_happen(prev, nstate)){
2551             SimoCPP(nstate, new_strain, new_T, prev, PT_REORIENT);
2552             RestoreConsistency_A2Md(nstate, new_strain, new_T, prev);
2553         }
2554     }
2555     delete prev;
2556 }
2557 }
2558
2559
2560 bool TSMASState::check_consistency(TSMASState *ostate){
2561     bool res;
2562     tensor tmp, e_strain;
2563     tensor4 C;
2564
2565     res = (-EPS < c1()) && (c1() < 1.0 + EPS)
2566         && (-EPS < c2()) && (c2() < 1.0 + EPS)
2567         && (-EPS < c3()) && (c3() < 1.0 + EPS);
2568
2569     res = res && (fabs(c1() + c2() + c3() - 1.0) < EPS);
2570
2571     if (!res) return res;
2572
2573     if (is_A2Mt_possible() && does_A2Mt_happen(ostate, this)){
2574         throw ESMAError("Inconsistent state. A2Mt violated");
2575     };
2576     if (is_A2Md_possible() && does_A2Md_happen(ostate, this)){
2577         throw ESMAError("Inconsistent state. A2Md violated");
2578     };
2579     if (is_Md2A_possible() && does_Md2A_happen(ostate, this)){
2580         throw ESMAError("Inconsistent state. Md2A violated");
2581     };
2582     if (is_Mt2A_possible() && does_Mt2A_happen(ostate, this)){
2583         throw ESMAError("Inconsistent state. Mt2A violated");
2584     };
2585     if (is_Mt2Md_possible() && does_Mt2Md_happen(ostate, this)){
2586         does_Mt2Md_happen(ostate, this);
2587         cout << "Inconsistent state. Mt2Md violated";
2588         throw ESMAError("Inconsistent state. Mt2Md violated");
2589     };
2590 }

```

```
2591
2592     contract42(&tmp, EffctvC(&C), ElasticStrain(&e_strain));
2593     sub(&tmp, &tmp, &stress);
2594     return (norm(&tmp) < 1.0e-3);
2595
2596 }
2597
2598 tensor4Ptr TMSASolver::NumericalJacobian(tensor4 L, TMSAState *nstate, TMSAState *ostate)
2599 {
2600     TMSAState eps_state;
2601     double dstrain = 1.0e-8;
2602     tensor strain;
2603
2604     for (int k = 0; k < 3; k++){
2605         for (int l = 0; l < 3; l++){
2606             eps_state.Assign(ostate);
2607             assign_tensor(&strain, nstate->Strain());
2608             strain[k][l] += dstrain;
2609             //if (k != l) strain[l][k] += dstrain;
2610             Load(&eps_state, &strain, nstate->T, ostate);
2611             for (int i = 0; i < 3; i++){
2612                 for (int j = 0; j < 3; j++){
2613                     {
2614                         L[i][j][k][l] = (eps_state.stress[i][j] - nstate->stress[i][j])/dstrain;
2615                     }
2616                 }
2617             }
2618             return (tensor4Ptr)&L;
2619         }
2620     }
```

VITA

Petar Angelov Popov (also spelled Peter Angelov Popov), son of Galina Nikoltcheva and Angel Popov, was born on June 16, 1977 in Sofia, Bulgaria. He finished Sofia Mathematics High School "Paisij Hilendarski" in May, 1995. He then attended the Faculty of Mathematics and Informatics at the Sofia University "St. Kliment Ohridski" and successfully graduated with a major in Applied Mathematics, specialization Mechanics in October, 1999. Since January, 2000, he began Ph.D. studies at the Department of Aerospace Engineering, Texas A&M University, College Station, Texas. Petar Popov can be reached at the following permanent address:

22 Carnegie Str,
Sofia 1000,
BULGARIA