ABSTRACT

FRAUTSCHI, JASON PAUL. Finite Element Simulations of Shape Memory Alloy Actuators in Adaptive Structures. (Under the direction of Stefan Seelecke.)

Shape memory alloys possess an inherent actuation capability that makes them attractive as actuators in adaptive structures, especially in applications in which their large strains, high specific work output and potential for structural integration are beneficial. However, the requisite extensive physical testing has slowed development of potential applications and highlighted the need for a simulation tool for feasibility studies. In this study, such a tool has been developed by implementing the Müller-Achenbach-Seelecke shape memory alloy model into a commercial finite element code. The material model is described with particular emphasis on its ability to predict actuatoric performance and suitability for use in the context of a displacement-based finite element framework. The interaction between the material model and the solution algorithm for the global finite element equations is thoroughly investigated with respect to the effect of solution parameters on convergence, computational cost and accuracy. Finally, simulations of several flexible structures actuated by shape memory alloys are presented as examples of the potential of the implementation to analyze practical applications. The implementation represents a versatile and novel tool for the simulation of adaptive structural components using shape memory alloy actuators.

FINITE ELEMENT SIMULATIONS OF SHAPE MEMORY ALLOY ACTUATORS IN ADAPTIVE STRUCTURES

by
JASON PAUL FRAUTSCHI

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

MECHANICAL AND AEROSPACE ENGINEERING

Raleigh

2003

APPROVED BY:

Dr. Mohammad N. Noori	Dr. Ralph C. Smith

Dr. Stefan Seelecke
Chair of Advisory Committee

PERSONAL BIOGRAPHY

Jason Paul Frautschi, born in Milwaukee, WI, obtained his Bachelor of Science degree in mechanical engineering from Temple University in Philadelphia in 2000. There, after receiving the Ennis W. Cosby faculty scholarship, he received first honors in the School of Engineering, graduating *summa cum laude* in three years. Afterwards, he worked as a Project Engineer with a fledgling consulting company before meeting Dr. Seelecke and electing to return to school at North Carolina State University in pursuit of a Master of Science degree.

ACKNOWLEDGEMENTS

This research was supported by the National Science Foundation through the grant DMI-0134464.

Special thanks are due to each member of my advisory committee: Dr. Noori, for his steadfast confidence in my success at NCSU; Dr. Smith for his motivating enthusiasm and guidance; and most of all, to Dr. Seelecke, who from the moment I arrived on campus acted as a perpetual source of inspiration through his meticulous approach to science.

Thanks also to Dr. Seelecke's other graduate students, Olaf Heintze and Jinghua Zhong, who have provided humor and camaraderie in the office.

I thank my extended family, whose curiosity in my studies and concern for my well-being have buoyed my spirits and maximized the enjoyment I have received from my work here at NCSU.

Above all I owe thanks to my wife, Erika, who has supported me already through one career transition and two degrees. Without her I surely would not be who and where I am today.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

1. INTRODUCTION

The properties of shape memory alloys (SMA) make them viable as actuators in many applications, particularly those that are weight critical, that are used in 'clean-room' environments, or that require a high force and high stroke under severe space restrictions. The so-called 'shape memory' effect, resulting from solid-solid phase transformations in the crystal lattice structure, lends these materials an inherent actuation capability combining high strains with an exceptional specific work output. Figure 1 compares SMA actuator performance with a selection of other actuation methods and materials. Shape memory alloys lead the list in maximum actuation stress and rival even hydraulics in specific work output, approaching 10^8 J/m^3. This is the highest specific work output amongst all known smart materials.

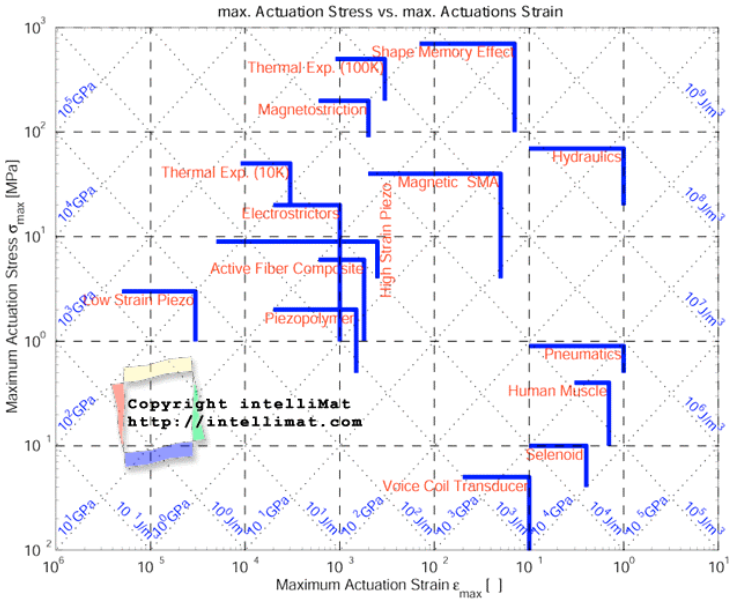

Figure 1: Comparison of actuation performance [1]

Their inherent actuation capability translates into yet another attribute, the ability to be directly integrated into a structure. This introduces the possibility of reducing or eliminating force transmission mechanisms or the piping and pumping infrastructure typically associated with the more common actuation methods, such as hydraulic or pneumatic systems. This distinction hints

at a number of potential fields of application for SMA actuators. For example, eliminating gear systems also obviates the need for lubricating oils that may not be compatible with ‘clean room’ environments, as well as removing the effects of gear lash and the resulting positioning inconsistency associated with these transmission devices. Similarly, reduction of the infrastructure needed for actuation mechanisms generates obvious benefits in weight- or space-restricted applications. For example, not only does eliminating the piping and pumping systems typical of more traditional actuation systems benefit performance through weight savings, but it also potentially liberates space in a structure, allowing modification to the structure’s configuration or providing extra housing space for additional systems. Further, the loss of piping systems precludes the need for routing cutouts in the structure, permitting structural optimization without the stress concentrations caused by these cutouts.

As a result of these attributes, potential applications for shape memory alloy actuators span the range from micro-scale positioning devices to large-scale structural actuators, in concepts as far-reaching as NASA’s ‘morphing airplane’ (Figure 2) to those already under experimental evaluation, such as the flexible wing trailing edge shown in Figure 3.

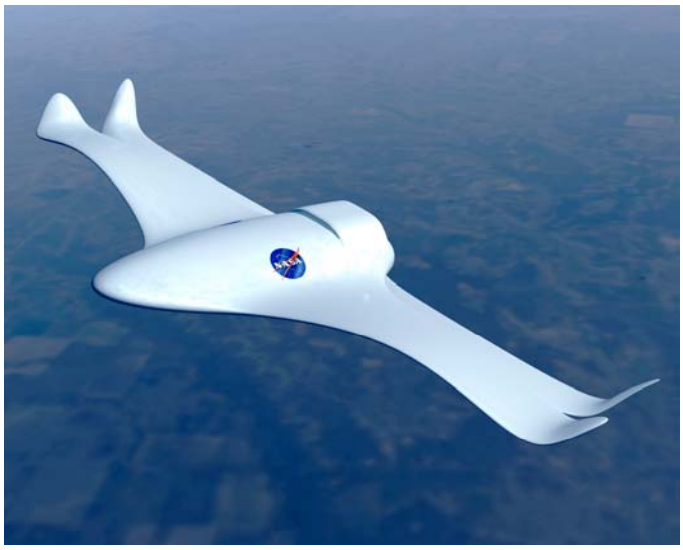

Figure 2: ‘Morphing Airplane' [2]

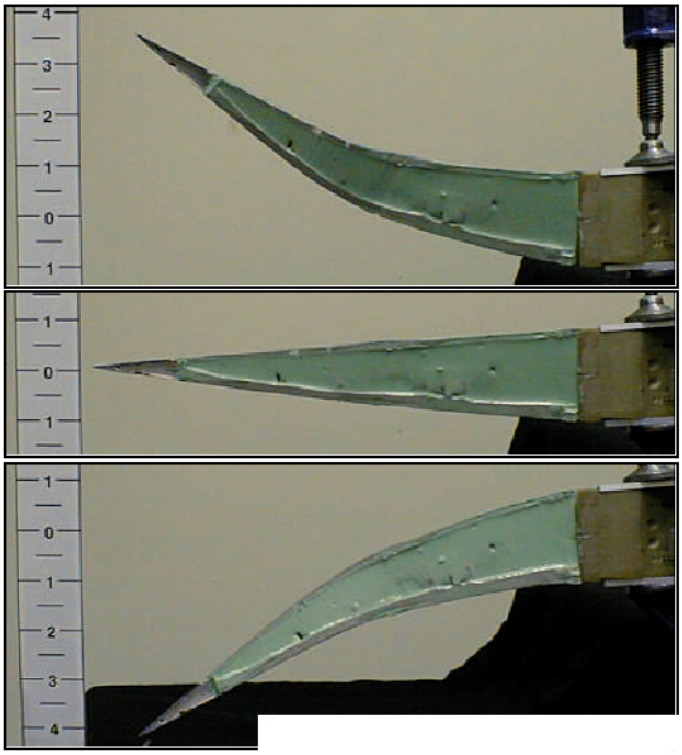

Figure 3: Flexible trailing edge of a wing actuated by SMA wires [3]

In fact, much of the development in the field of adaptive structures has been motivated by applications in aerospace engineering, where the properties of SMA actuators are particularly well-suited. These applications involve quasi-static shape control, such as the in-flight optimization of aircraft wings for improved fuel consumption and performance. Within this arena, the aircraft industry has pursued several research projects under so-called smart wing programs organized by the Defense Advanced Research Projects Agency (DARPA), resulting in breakthroughs in the technology that show promise for further development [4,5,6].

The fact that most of the concepts involving SMA actuators have been developed experimentally highlights the absence of a useful simulation tool for feasibility studies. Performance simulations offer a number of well known benefits as a first step in concept development. Namely, they have the potential to illuminate shortcomings in actuator design at a point in concept development where changes are easily and inexpensively accommodated, which can greatly reduce both the amount of time needed for performance testing and the number of

built design iterations needed to realize an effective mechanism. Additionally, the simulations can yield insight into the behavior of a structure that may be overlooked in physical experimentation, for example where comprehensive instrumentation is cost-prohibitive or otherwise unfeasible.

For structural simulations, the most widely used tool is the finite element (FE) method. This numerical method makes analysis of arbitrary structures possible, even those having irregular shapes, known material inhomogeneities, or other features that preclude the use of classical, analytical methods. Of course, for useful, physically representative simulations, developing a characteristic material model and correctly integrating the chosen simulation solution algorithm yields the greatest possibility for meaningful simulation results. This is particularly true for shape memory alloys, which show highly nonlinear and hysteretic behavior.

Though many models for the behavior of SMAs have been developed over the past two decades, the number of implementations into a finite element framework is very small, see, e.g., [7,8]. When it comes to actuator applications, the number of suitable SMA models further diminishes since most require direct prescription of the temperature. However, realistically, from a controls perspective the actuator temperature per se is a quantity that is both difficult to measure and impossible to instantaneously control.

More suitable would be a model that includes an energy balance for determining temperature and material behavior following from a prescribed electric current through the actuator, since the electric current is the essential control quantity for most SMA actuators. An example of such a model is Seelecke's improved Müller-Achenbach SMA model [9]. To the best of the author's knowledge, there has been only one publication of a SMA actuator model including an energy balance implemented into a FE code [10]. However this was an implementation based on a dimensionless SMA model using a research FE code. Consequently, it reproduced the material behavior only qualitatively and had only very basic structural elements available to which to couple the SMA actuator element.

The objective of this work is to implement a SMA actuator model capable of reproducing material behavior both qualitatively and quantitatively into the commercial FE code ANSYS for use in a wide range of feasibility studies [11]. For this it is necessary to fully understand both the

SMA material model and the FE solution process in order to guarantee that the relevant information is processed by the model and correctly translated to the global FE level. Further, in order to ensure that the material behavior characterized by the SMA model is accurately reproduced, the implementation requires substantiation through a comparison of simulation results with a reference solution generated by a SMA model version outside the FE framework. Additionally, the influence of the FE solution parameters must be studied with particular attention to their impact on the simulation behavior. Finally, to illustrate the potential of the implementation to simulate practical examples of adaptive structures with SMA actuators, several potential applications for SMA actuators are presented.

2. SHAPE MEMORY ALLOY BEHAVIOR

Shape memory alloys possess highly nonlinear and hysteretic constitutive behaviors showing a strong dependence on strain, temperature, and strain rate. These distinguishing features lend these materials an inherent actuation capability, coupling high strains with a very high work output. In this chapter, the characteristic behavior of shape memory alloy is detailed and correlated to a description of the material's Gibbs free energy, followed by an introduction to the model used in the FE implementation.

The macroscopic behavior of SMA is caused by solid-solid phase transitions in the material's crystal lattice structure, also referred to as martensitic transformations. In the uni-dimensional case, material behavior follows free energy minimization through diffusionless transitions between the cubic austenite phase (A) and/or two martensite variants (M_+ , M_-), which may be viewed as sheared lattice cells. These phase changes may result from either mechanical or thermal loading, or both, and lead to the dominant behavioral features of SMAs, including the so-called 'shape memory' effect. There are two characteristic states in which SMAs are found: 'quasi-plastic' and 'pseudo-elastic.' Interaction between these two conditions leads to the shape memory effect, and also lends SMA its actuation capability.

Shape memory alloys exhibit 'quasi-plastic' behavior when the martensite variants are the stable phases under no load, typically at low temperatures. In such a condition, stress application initially produces a linear elastic response, up to a characteristic, temperature-dependent phase transition stress. Loading above the transition stress causes substantial deformation coinciding with a phase transformation from one variant to the other (e.g. $M_- \rightarrow M_+$). However, unlike a purely plastic material, after the phase change is completed, further loading results in a resumption of the linear elastic behavior. Because the phase transition is not a spontaneous process, subsequent unloading leaves remnant deformation. Combining this remnant deformation with the distinctive elastic-yielding-elastic behavior explains the term 'quasi-

plastic,' since the material does not behave plastically in the pure sense, instead regaining its elastic nature after yielding.

SMAs behave in a 'pseudo-elastic' or super-elastic manner at higher temperatures, at which austenite is the stable phase under no load. Like the quasi-plastic case, initial loading produces a linear elastic response, up to a characteristic, temperature-dependent austenite-to-martensite (A → M) phase transition stress. Exceeding this critical stress causes a phase transformation from austenite to a martensite variant, accompanied by substantial straining. Like the quasi-plastic case, after the phase change is completed, further loading results in a resumption of the linear elastic behavior. When the stress is reduced below a lower, martensite-to-austenite (M → A) transition stress, the material transforms back to austenite, i.e. the M → A process occurs spontaneously. Complete unloading returns the sample to its original shape as in an elastic material, tracing a hysteretic path. Total strains may be on the order of 10 %. The very high strains achievable, combined with the complete recovery to the original configuration along a hysteretic path, explains the terms 'pseudo-elastic' and super-elastic.

The 'shape memory effect' results from the interplay of these two material behaviors, as illustrated in Figure 4. Starting in a quasi-plastic state (a twinned-martensite phase composition), with variants shown in green (representing M_+) and blue (representing M_-), initial loading causes existing M_+ layers to stretch along their original orientation, but begins to deform the M_- layers through shearing. Loading over the transition stress causes the M_- layers to lose stability and 'flip' into the M_+ orientation. Once this has occurred, unloading leaves the material in a state of remnant deformation since the reverse transformation does not occur spontaneously. After this plastic-like deformation in the quasi-plastic regime, the original shape may be recovered by heating the material up to the transition temperature, causing a phase transformation, whereupon the lattice cell structure shifts from the sheared martensite variant (M_+ in this case) into body centered cubic austenite cells, causing the sample to return to its original shape, as if by 'memory.' Loading at this elevated temperature would result in a pseudo-elastic behavior. Upon cessation of heating, as the material returns to ambient temperature, austenite becomes unstable and transforms to twinned martensite with no macroscopic shearing occurring. This is due to the

simultaneous formation of the two variants of martensite, which mutually cancel their strain contributions.

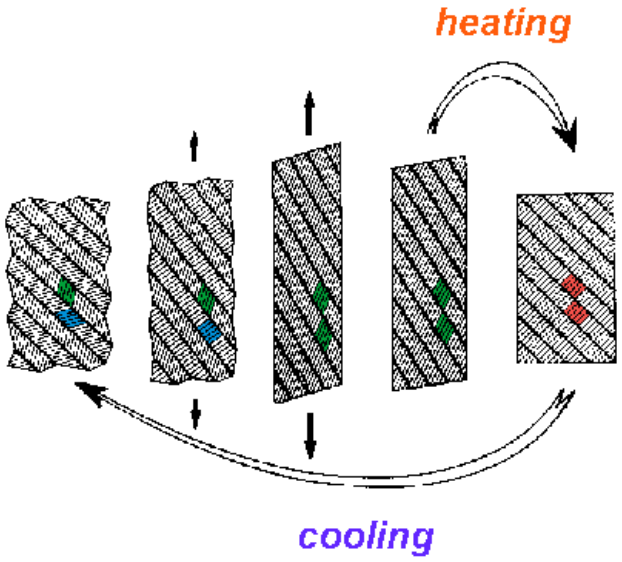

Figure 4: Shape memory effect [12]

Because of the difference in stress levels that induce transformation from one phase to another (or from one martensite variant to the other), there is a very pronounced hysteresis loop during cyclical loading where phase changes occur. Such hysteresis loops dissipate energy, producing a damping effect in vibration [13]. Thus shape memory alloys used as passive elements in a structure act as both non-linear springs and passive dampers, provided strain levels are sufficiently high to induce phase transformations. In applications where the SMA is used as an active component in an adaptive structure, and where the driving frequency is significantly smaller that the structural natural frequency, the hysteresis acts as an impediment in the ability of conventional control methods to follow a set track and necessitates the use of a model-based control scheme for optimal operation, particularly at higher frequencies.

An introduction to the shape memory effect can be found in, among others, [14,15].

2.1. SMA model

For this implementation the model representing the constitutive stress-strain relation of the SMA actuator is Seelecke's extension of the Müller-Achenbach model [16]. This model describes phase transformations as thermally activated processes governed by principles from statistical thermodynamics, and employs an ansatz free energy density function. A system of first-order, ordinary differential equations (ODEs) representing evolution laws for the phase fractions and temperature constitute the model.

Phase changes resulting from mechanical or thermal loading are predicted according to the fundamental quantity of interest, the Gibbs free energy, which is assumed to be described by a multi-parabolic function dependent on strain, temperature, and applied stress. Local minimum energy locations, or "wells," for each phase are provided, as well as energy barriers between wells. Phase transitions occur when the barriers between phase-specific free energy minimums are eliminated by either mechanical or thermal loading.

In the following, after the Gibbs free energy is described, the model is introduced through a brief summary of its equations and an examination of representative plots characterizing its behavior.

2.1.1. Free energy description

Since the Gibbs free energy is the fundamental quantity in the model, it is useful to understand its form in order to comprehend the machinations of the model. Characterization of SMA behavior is attained by means of a free energy description having a hypothesized multi-parabolic shape representing the free energy density as a function of strain and temperature. At equilibrium under no load, the micro-structural phase composition may be predicted at a temperature via the Helmholtz free energy density function; phase changes resulting from mechanical loading may be predicted by the Gibbs free energy density function.

This Helmholtz free energy density function, ψ, is assumed to be composed of five parabolas, one for each of the three material phases (A, M_+ , M_-) and two for the energy barriers between austenite and the two martensite phases (shown as C_+ , C_- in Figure 5) .

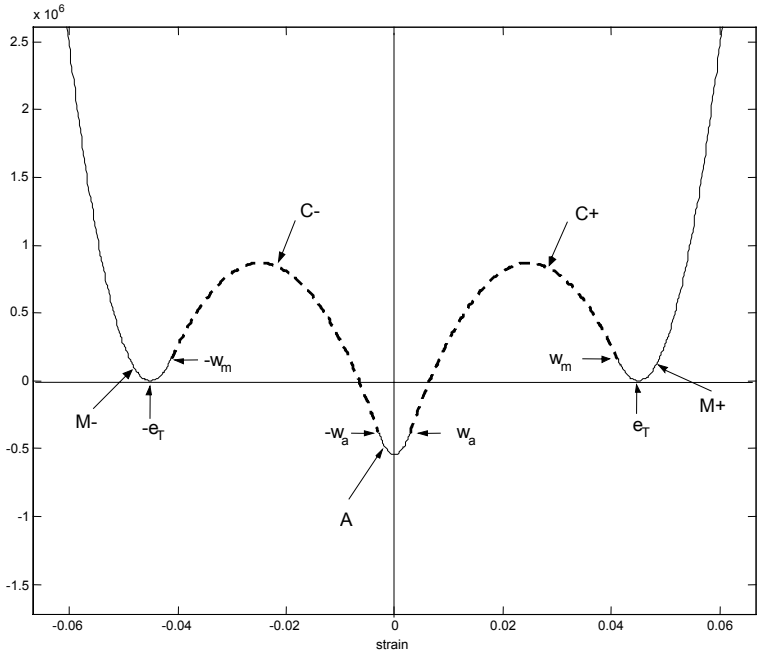

Figure 5: Helmholtz free energy density curve

Identification of some particular points of interest in the ψ - strain curve facilitates generation of the Helmholtz free energy density function. These points include the phase minimum energy locations and the intersections of the phase and barrier parabolae as follows (refer to Figure 5):

- ε_T and $-\varepsilon_T$ are defined to be the strains corresponding to locations of the local minimum energy wells for the sheared M_+ and M_- phases, respectively. The austenitic minimum energy well occurs at zero strain.
- strains corresponding to inflection points between the austenite phase and its energy barriers are defined as w_a and $-w_a$.
- strains corresponding to inflection points between the two martensitic phases and their respective energy barriers are defined as w_m and $-w_m$.

Note that the strain ε_T is constant for a material, whereas w_A and w_M depend on the relative positions of the austenite and martensite energy minima, as well as the curvature of the parabolae.

With the supposition that each phase and energy barrier may be depicted by a parabola, then each individual parabola could be described by the standard quadratic formulation:

$$\psi_i = a_i(T) * \varepsilon^2 + b_i(T) * \varepsilon + c_i(T)$$

where i = {1:5} corresponding to {A, M_+ , M_- , C_+ , C_- }, respectively, the three phases and two connecting parabolae. For generality, the coefficients are permitted to vary with temperature *T*.

Coefficient Determination

By assigning arbitrary nominal values to the minima of the three phases, i.e. $\beta_1(T)$ for austenite, $\beta_2(T)$ and $\beta_3(T)$ for martensites M_+ and M_- , equations for the individual phase parabolae and barrier parabolae can be determined. These nominal values, combined with assumed continuity and smoothness between the parabolae, dictate the following required conditions:

$$\Psi_1(0,T) = \beta_1(T) \tag{2.1.1.1}$$

$$\left.\frac{\partial \Psi_1}{\partial \varepsilon}\right|_{(0,T)} = 0 \tag{2.1.1.2}$$

$$\Psi_2(\varepsilon_T,T) = \beta_2(T) \tag{2.1.1.3}$$

$$\left.\frac{\partial \Psi_2}{\partial \varepsilon}\right|_{(\varepsilon_T,T)} = 0 \tag{2.1.1.4}$$

$$\Psi_3(-\varepsilon_T,T) = \beta_3(T) \tag{2.1.1.5}$$

$$\left.\frac{\partial \Psi_3}{\partial \varepsilon}\right|_{(-\varepsilon_T,T)} = 0 \tag{2.1.1.6}$$

$$\Psi_4(w_A,T) = \Psi_1(w_A,T) \tag{2.1.1.7}$$

$$\left.\frac{\partial \Psi_4}{\partial \varepsilon}\right|_{(w_A,T)} = \left.\frac{\partial \Psi_1}{\partial \varepsilon}\right|_{(w_A,T)} \tag{2.1.1.8}$$

$$\Psi_4(w_M, T) = \Psi_2(w_M, T) \tag{2.1.1.9}$$

$$\left.\frac{\partial \Psi_4}{\partial \varepsilon}\right|_{(w_M, T)} = \left.\frac{\partial \Psi_2}{\partial \varepsilon}\right|_{(w_M, T)} \tag{2.1.1.10}$$

$$\Psi_5(-w_A, T) = \Psi_1(-w_A, T) \tag{2.1.1.11}$$

$$\left.\frac{\partial \Psi_5}{\partial \varepsilon}\right|_{(-w_A, T)} = \left.\frac{\partial \Psi_1}{\partial \varepsilon}\right|_{(-w_A, T)} \tag{2.1.1.12}$$

$$\Psi_5(-w_M, T) = \Psi_3(-w_M, T) \tag{2.1.1.13}$$

$$\left.\frac{\partial \Psi_5}{\partial \varepsilon}\right|_{(-w_M, T)} = \left.\frac{\partial \Psi_3}{\partial \varepsilon}\right|_{(-w_M, T)} \tag{2.1.1.14}$$

Evaluation of the above requirements yields equations for the coefficients of the parabolae, as follows:

From 2.1.1.1: $c_1(T) = \beta_1(T)$

From 2.1.1.2: $b_1(T) = 0$

From 2.1.1.3: $a_2(T)\ \varepsilon_T^{\ 2} + b_2(T)\ \varepsilon_T + c_2(T) = \beta_2(T)$

From 2.1.1.4: $2\ a_2(T)\ \varepsilon_T + b_2(T) = 0$

From 2.1.1.5: $a_3(T)\ (-\varepsilon_T)^2 + b_3(T)\ (-\varepsilon_T) + c_3(T) = \beta_3(T)$

From 2.1.1.6: $2\ a_3(T)\ (-\varepsilon_T) + b_3(T) = 0$

From 2.1.1.7: $a_4(T)\ w_A^{\ 2} + b_4(T)\ w_A + c_4(T) = a_1(T)\ w_A^{\ 2} + b_1(T)\ w_A + c_1(T)$

From 2.1.1.8: $2\ a_4(T)\ w_A + b_4(T) = 2\ a_1(T)\ w_A + b_1(T)$

From 2.1.1.9: $a_4(T)\ w_M^{\ 2} + b_4(T)\ w_M + c_4(T) = a_2(T)\ w_M^{\ 2} + b_2(T)\ w_M + c_2(T)$

From 2.1.1.10: $2\ a_4(T)\ w_M + b_4(T) = 2\ a_2(T)\ w_M + b_2(T)$

From 2.1.1.11: $a_5(T)\ (-w_A)^2 + b_5(T)\ (-w_A) + c_5(T) = a_1(T)\ (-w_A)^2 + b_1(T)\ (-w_A) + c_1(T)$

From 2.1.1.12: $2\ a_5(T)\ (-w_A) + b_5(T) = 2\ a_1(T)\ (-w_A) + b_1(T)$

From 2.1.1.13: $a_5(T)\ (-w_M)^2 + b_5(T)\ (-w_M) + c_5(T) = a_3(T)\ (-w_M)^2 + b_3(T)\ (-w_M) + c_3(T)$

From 2.1.1.14: $2\ a_5(T)\ (-w_M) + b_5(T) = 2a_3(T)\ (-w_M) + b_3(T)$

Since there are only fourteen equations available for the solution of the fifteen parabola coefficients, not all of the coefficients can be explicitly determined. Moreover, introduction of the strain values ε_T, w_A, and w_M, and the phase minima β_1, β_2, and β_3 adds to the list of unknowns. Consequently, other equations or relationships are required for a full description of the free energy curve. However, the coefficients may be solved as functions of $a_1(T)$, $a_2(T)$, $a_3(T)$, ε_T, w_A, w_M, β_1, β_2, and β_3, as shown below:

$$a_4(T) = (a_1(T)\, w_A + a_2(T)\, (\varepsilon_T - w_M)) / (w_A - w_M)$$

$$a_5(T) = (a_1(T)\, w_A + a_3(T)\, (\varepsilon_T - w_M)) / (w_A - w_M)$$

$$b_1(T) = 0$$

$$b_2(T) = -\, 2\, a_2(T)\, \varepsilon_T$$

$$b_3(T) = \; 2\, a_3(T)\, \varepsilon_T$$

$$b_4(T) = -\, 2\, w_A\, (a_1(T)\, w_M + a_2(T)\, (\varepsilon_T - w_M)) / (w_A - w_M)$$

$$b_5(T) = \; 2\, w_A\, (a_1(T)\, w_M + a_2(T)\, (\varepsilon_T - w_M)) / (w_A - w_M)$$

$$c_1(T) = \beta_1(T)$$

$$c_2(T) = a_2(T)\, \varepsilon_T^2 + \beta_2(T)$$

$$c_3(T) = a_3(T)\, \varepsilon_T^2 + \beta_3(T)$$

$$c_4(T) = w_A^2\, (a_1(T)\, w_M + a_2(T)\, (\varepsilon_T - w_M)) / (w_A - w_M) + \beta_1(T)$$

$$c_5(T) = w_A^2\, (a_1(T)\, w_M + a_3(T)\, (\varepsilon_T - w_M)) / (w_A - w_M) + \beta_1(T)$$

$$c_4(T) = w_M^2\, (a_1(T)\, w_A + a_2(T)\, (\varepsilon_T - w_A)) / (w_A - w_M) + a_2(T)\, \varepsilon_T^2 + \beta_2(T)$$

$$c_5(T) = w_M^2\, (a_1(T)\, w_A + a_3(T)\, (\varepsilon_T - w_A)) / (w_A - w_M) + a_3(T)\, \varepsilon_T^2 + \beta_3(T)$$

Correlation Between Coefficients and Material Properties

Further relationships must be established for physically meaningful expressions for the remaining coefficients and unknown values. Certain thermodynamic considerations indicate that most of these variables can be found from experimental data. For example, the Gibbs Equation for a 1-D solid is as follows:

$$Tds = du - \sigma d\varepsilon \ , \ or$$
$$\sigma d\varepsilon + Tds = du \tag{2.1.1.15}$$

where *s* is entropy, *u* is internal energy. By subtracting $d(Ts)$ from both sides of the equation, a slightly modified form results, noting that $d(Ts) = Tds + sdT$:

$$\sigma d\varepsilon + Tds - d(Ts) = du - d(Ts), \ or$$
$$\sigma d\varepsilon - sdT = d(u - Ts) \tag{2.1.1.16}$$

Then, since by definition, $\Psi = u - Ts$, the differential of the Helmholtz free energy equation emerges:

$$d\Psi = \sigma d\varepsilon - sdT \tag{2.1.1.17}$$

From this, since $d\Psi = \left.\frac{\partial \Psi}{\partial T}\right|_{\varepsilon} dT + \left.\frac{\partial \Psi}{\partial \varepsilon}\right|_{T} d\varepsilon$, then

$$\left.\frac{\partial \Psi}{\partial T}\right|_{\varepsilon=constant} = -s \text{ , and} \tag{2.1.1.18}$$

$$\left.\frac{\partial \Psi}{\partial \varepsilon}\right|_{T=constant} = \sigma \tag{2.1.1.19}$$

The latter equation above motivates the use of an experimentally obtained stress-strain diagram to aid in the development of the free energy description. The stress-strain diagram yields the following useful information:

- E_A = 2 $a_1(T)$ = modulus of the austenite phase (by noting that $\left.\frac{\partial \sigma}{\partial \varepsilon}\right|_A = E_A = \left.\frac{\partial^2 \Psi}{\partial \varepsilon^2}\right|_A$)
- similarly, E_M = 2 $a_2(T)$ = 2 $a_3(T)$ = modulus of the martensite phases
- ε_T = the strain level of a hypothetical, purely martensitic (M_+) material at vanishing stress, which occurs at the minimum of the M_+ parabola
- w_A = the strain at which the phase transition from austenite to martensite begins
- w_M = the strain at which the phase transition from martensite to austenite begins

- there is a loading stress at which a phase transition from austenite to martensite (+) begins (say, 'σ_A'), corresponding to w_A, where the material strains at a constant stress
- there is an unloading stress at which a phase transition from martensite (M_+) to austenite begins (say, 'σ_M'), corresponding to w_M, where the material strains at a constant stress
- the diagram is symmetric

All coefficients previously developed can be written in terms of these material properties and constants. Further, stress-strain diagrams from experiments conducted at various temperatures reveal that the difference between σ_A and σ_M remains constant (independent of temperature), i.e.

$$\sigma_A - \sigma_M \text{ = some constant} \equiv \Delta\sigma \tag{2.1.1.20}$$

As defined, w_A and w_M are intrinsically related to σ_A and σ_M; w_A is the strain at which phase transition (A → M_+) begins and thus the point at which the stress σ_A is achieved, and similarly w_M marks the onset of the M_+ → A transition at σ_M. Consequently, w_A and w_M can be represented by σ_A and $\Delta\sigma$ as follows:

$$\begin{aligned} w_A(T) &= \frac{\sigma_A(T)}{E_A} \\ w_M(T) &= \left(\frac{\sigma_A(T) - \Delta\sigma}{E_M} \right) + \varepsilon_T \end{aligned} \tag{2.1.1.21}$$

Relative Minimum Energies of the Phases

The two redundant expressions for $c_4(T)$ and $c_5(T)$ render relationships between $\beta_1(T)$ and both $\beta_2(T)$ and $\beta_3(T)$. For example,

$$\beta_1 + w_A(T)^2 \, (a_1(T)\, w_M(T) + a_2(T)\, (\varepsilon_T - w_M(T))) \,/\, (w_A(T) - w_M(T)) \quad =$$

$$\beta_2 + a_2(T)\, \varepsilon_T^2 + w_M(T)^2 \, (a_1(T)\, w_A(T) + a_2(T)\, (\varepsilon_T - w_A(T))) \,/\, (w_A(T) - w_M(T))$$

From the above, the magnitude of the difference in minimum energies of the austenite (A) and martensite (M_+ or M_-) emerges.

The crystallographic symmetry of the material implies a symmetric stress-strain diagram. This in turn suggests that the martensitic phase minimum energies β_2 and β_3 are equal and might justifiably be denoted as martensitic, i.e. "β_M." Similarly, β_1 might be justifiably denoted with an austenite subscript, "β_A." Hence, considering the austenite and martensite (+) phases and introducing $\Delta\beta \equiv \beta_A - \beta_M$:

$$\Delta\beta = \beta_A - \beta_M = - a_1(T)\, w_A(T)\, w_M(T) + a_2(T)\, (\varepsilon_T - w_A(T))\, (\varepsilon_T - w_M(T)) \qquad (2.1.1.22)$$

$\Delta\beta$ can also be expressed in terms of the material properties and constants found from the stress-strain diagram as follows:

$$\Delta\beta = 1/2\, (\sigma_A^2\, (1/E_A - 1/E_M) - \sigma_A\, (2\varepsilon_T + \Delta\sigma\, (1/E_A - 1/E_M)) + \Delta\sigma\, \varepsilon_T) \qquad (2.1.1.23)$$

Moreover, thermodynamic considerations motivate yet another expression for the β_i, which are sometimes also referred to as chemical free energies:

$$\beta_i = u_i - s_i T + c(T - T_{reference}) + cT \ln\left(\frac{T}{T_{reference}}\right), \quad \text{i = 1,2,3} \qquad (2.1.1.24)$$

where

$s_1 = s_{(austenite)} = s_A$,

$s_{2,3} = s_{(martensite)} = s_M$,

$u_1 = u_{(austenite)} = u_A$, and

$u_{2,3} = u_{(martensite)} = u_M$

With the above relation and the assumption that the specific heat constant c is the same for both the austenite and martensite phases, $\Delta\beta$ can be rewritten as

$$\Delta\beta = (u_A - u_M) - (s_A - s_M)\, T$$

or, introducing $\Delta u \equiv u_A - u_M$ and $\Delta s \equiv s_A - s_M$, simply

$$\Delta\beta = \Delta u - \Delta s\, T \qquad (2.1.1.25)$$

Internal Energy, Entropy, Loading Phase Transformation Stress, $\sigma_A(T)$

The above expression denotes a linear dependence on temperature in the relative minimum energies of austenite and martensite; thus by determining $\Delta\beta$ at two distinct

temperatures (i.e. some upper temperature, T_U, and some lower temperature, T_L) through measurement of σ_A and σ_M (from the stress-strain diagram), Δu and Δs can be determined, i.e.

$$\Delta\beta(T_U) = \Delta u - \Delta s T_U = \frac{1}{2}\left((\sigma_A(T_U))^2 \left(\frac{1}{E_A} - \frac{1}{E_M} \right) - \sigma_A(T_U) \left(2\varepsilon_T + \Delta\sigma \left(\frac{1}{E_A} - \frac{1}{E_M} \right) \right) + \Delta\sigma\varepsilon_T \right) \quad (2.1.1.26)$$

and

$$\Delta\beta(T_L) = \Delta u - \Delta s T_L = \frac{1}{2}\left((\sigma_A(T_L))^2 \left(\frac{1}{E_A} - \frac{1}{E_M} \right) - \sigma_A(T_L) \left(2\varepsilon_T + \Delta\sigma \left(\frac{1}{E_A} - \frac{1}{E_M} \right) \right) + \Delta\sigma\varepsilon_T \right) \quad (2.1.1.27)$$

Then,

$$\Delta s = \frac{-1}{2(T_U - T_L)} \left(\begin{array}{r} \left((\sigma_A(T_U))^2 - (\sigma_A(T_L))^2 \right) \left(\frac{1}{E_A} - \frac{1}{E_M} \right) - \ldots \\ \ldots (\sigma_A(T_U) - \sigma_A(T_L)) \left(2\varepsilon_T + \Delta\sigma \left(\frac{1}{E_A} - \frac{1}{E_M} \right) \right) \end{array} \right) \quad (2.1.1.28)$$

and (from either of the temperatures at which $\Delta\beta$ is determined),

$$\Delta u = \Delta s T_U + \frac{1}{2}\left((\sigma_A(T_U))^2 \left(\frac{1}{E_A} - \frac{1}{E_M} \right) - \sigma_A(T_U) \left(2\varepsilon_T + \Delta\sigma \left(\frac{1}{E_A} - \frac{1}{E_M} \right) \right) + \Delta\sigma\varepsilon_T \right) \quad (2.1.1.29)$$

Interjecting these Δs and Δu values into the $\Delta\beta$ equation, $\sigma_A(T)$ can be found for any temperature by using the standard quadratic formula, i.e. from

$$\sigma_A(T) = \quad (2.1.1.30)$$

$$\frac{2\varepsilon_T + \Delta\sigma \left(\frac{1}{E_A} - \frac{1}{E_M} \right) - \sqrt{\left(2\varepsilon_T + \Delta\sigma \left(\frac{1}{E_A} - \frac{1}{E_M} \right) \right)^2 - 4\left(\frac{1}{E_A} - \frac{1}{E_M} \right) (\Delta\sigma\varepsilon_T - 2(\Delta u - \Delta s T))}}{2\left(\frac{1}{E_A} - \frac{1}{E_M} \right)}$$

The choice of the negative radical is justified by the fact that the positive radical yields a negative yield stress value for all temperatures, a result that is physically meaningless.

From the above equations and any stress-strain diagram, individual phase values of entropy and internal energy cannot be determined, and consequently values for β_A and β_M cannot be determined. However, since in the assessment of relative equilibrium energy minima, the absolute values of β_A and β_M are irrelevant *vis-a-vis* the difference between them, arbitrarily prescribing some value of entropy and internal energy for martensite and finding an offset from this in austenite is sufficient. For example, by setting s_M and u_M equal to zero, then

$s_A = \Delta s$ and

$u_A = \Delta u$

Following from above,

$$\beta_A = \Delta u - \Delta s T - c(T - T_{Reference}) - cT\ln(T/T_{Reference}) \quad (2.1.1.31)$$

and

$$\beta_M = - c(T - T_{Reference}) - cT\ln(T/T_{Reference}) \quad (2.1.1.32)$$

Resulting Free Energy Descriptions

With the coefficients and material parameters developed above, a variation of the Helmholtz free energy, defined as

$$\hat{\Psi} \equiv \psi - c(T - T_{Reference}) - cT\ln(T/T_{Reference}) \quad (2.1.1.33)$$

can be plotted as a function of strain and temperature. The definition of $\hat{\Psi}$ effectively locks the minima of the martensitic phase parabolas to the abscissa and permits the austenitic energy minimum to float in relation to the zero position. The plot of $\hat{\Psi}$ shows both the relative magnitude of the austenitic energy minimum and the magnitude of the energy barrier(s) between phases.

Similarly, a modified form of the Gibbs free energy can be plotted as follows:

$$\hat{G}(\sigma, \varepsilon, T) = G(\hat{\Psi}, \sigma) = \hat{\Psi}(\varepsilon, T) - \sigma\varepsilon \quad (2.1.1.34)$$

where σ is the applied mechanical stress. Examination of the $\hat{G}$ function presents a rough means of predicting the onset of phase transition under loading.

Qualitatively, phase transitions can be understood by examination of the Gibbs free energy function. A plot representing a typical NiTi material in the unloaded state at 293 K, at which all phases are stable, appears in Figure 6. The austenite phase sits in the zero strain well, while the martensite variants M_- and M_+ occupy the left- and right-side wells, respectively. No phase transitions would occur under the shown condition since the barriers separating the three phase wells obstruct the free exchange of particles.

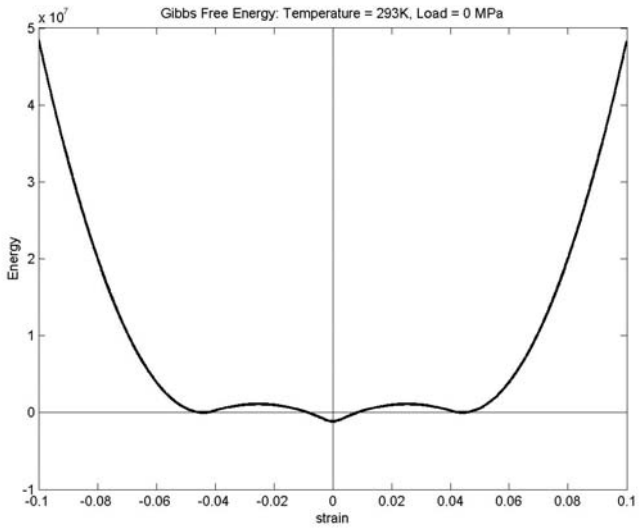

Figure 6 : Gibbs free energy density function @ 293 K, no load

Figure 7 shows the same material at a lower temperature, at which austenite is not a phase present in equilibrium. In lowering the temperature from that of the first plot to that of the second, the austenitic minimum energy well disappears, rendering austenite unstable and causing it to transform to either of the martensite variants, without preference.

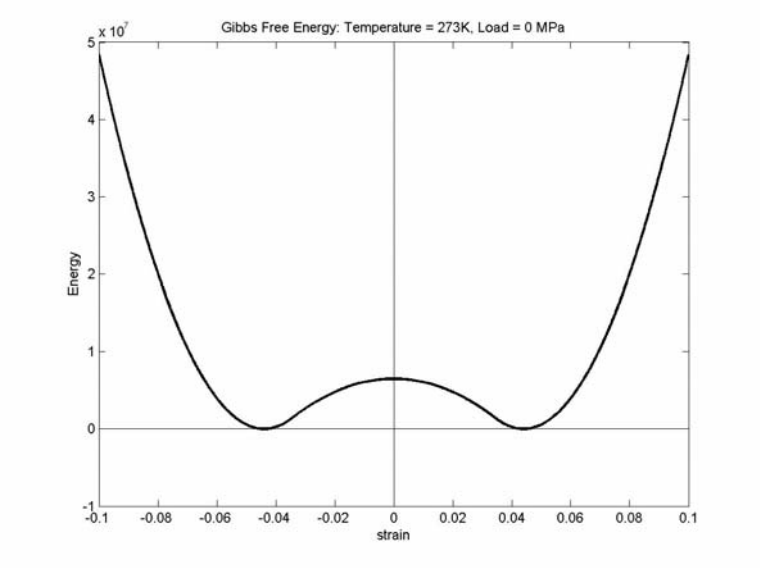

Figure 7 : Gibbs free energy density function @ 273 K, no load

The above plots show qualitatively how phase changes result from thermal loading.

Under mechanical loading, the Gibbs free energy function inclines with the product of stress and strain, as shown in Figure 8. The lack of a barrier at the M_- phase minimum, indicated by the arrow, indicates a phase instability due to the accessibility of the preferable, lower energy value at the M_+ well. In this case, the applied stress is sufficient to cause a phase change from one martensite variant to the other ($M_- \rightarrow M_+$).

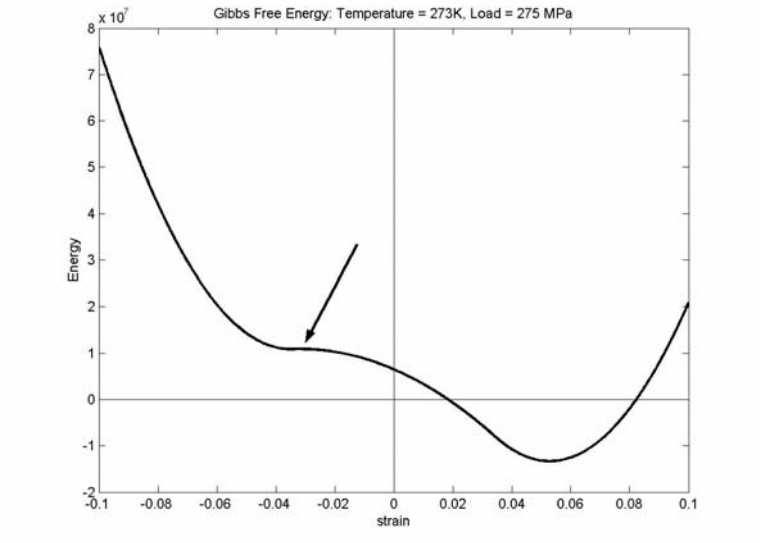

Figure 8 : Gibbs free energy density function @ 273 K, loaded

The next plot, Figure 9, shows the same material at a higher temperature under the same load. This temperature corresponds to that used to obtain Figure 6. Because of the absence of an energy barrier at the M_+ well, all martensite previously formed under loading will transform into austenite.

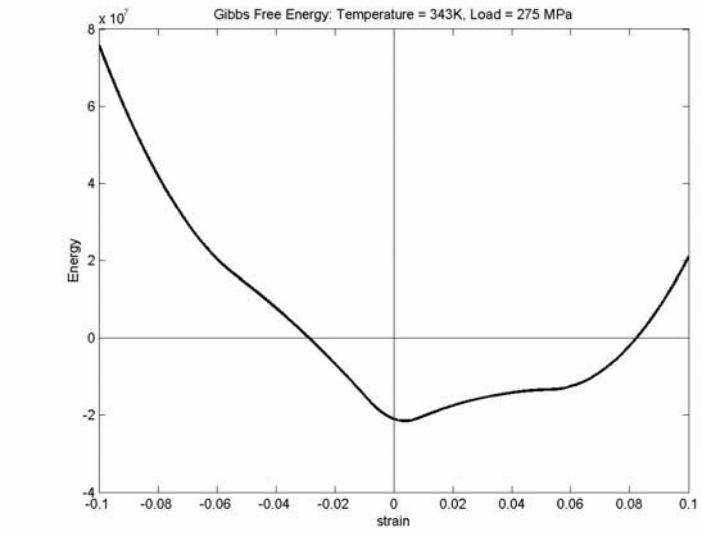

Figure 9 : Gibbs free energy density function @ 343 K, loaded

The last two figures demonstrate how phase changes result from the combination of both thermal and mechanical loading.

2.1.2. Model equations

However, a description of phase transformations based purely on the shape of the free energy function is physically inadequate to describe the material behavior. According to the theory of thermally activated processes, lattice layers in a particular phase within a shape memory alloy material are assumed to fluctuate around their minimum energy positions because of their intrinsic thermal activity. Such fluctuations may trigger phase transformations at a lower stress level than that predicted by the Gibbs energy density function. Consequently the Müller-

Achenbach-Seelecke SMA model instead ascertains the probability that a material particle will be within a certain range of an energy well in order to predict the onset of a phase transformation according to principles from statistical thermodynamics. The transition probability for $M_+ \rightarrow A$, for example, is as follows:

$$p^{A+} = \sqrt{\frac{kT}{2\pi\tilde{m}}} \frac{\exp\left(-\frac{G(\hat{\varepsilon},\sigma,T)}{kT}\right)}{\int \exp\left(-\frac{G(\varepsilon,\sigma,T)}{kT}\right) d\varepsilon} \tag{2.1.2.1}$$

where,

$G(\sigma,\varepsilon,T)$ is the Gibbs Free Energy function;

k is the Boltzmann constant

$\hat{\varepsilon}$ is the strain that marks the onset of instability from Austenite to M+ ;

$\tilde{m} = m \cdot l^2$; where

m is the mass of a lattice layer;

$l = \sqrt[3]{V_L}$; where

V_L is a representative volume.

Limits of integration follow the stable strain range for austenite. These transition probabilities are used in defining evolution laws for phase fractions, as follows:

$$\begin{aligned} \dot{x}^+ &= -x^+ p^{+A} + x^A p^{A+} \\ \dot{x}^- &= -x^- p^{-A} + x^A p^{A-} \end{aligned} \tag{2.1.2.2}$$

where

$\dot{x}^+$ is the time rate of change of the fraction of the material in the M_+ phase;

x^+ is the fraction of the material in the M_+ phase;

p^{+A} is the transition probability for a phase transformation from M_+ A (as previously defined); and so on.

The right hand sides of the ordinary differential equations (ODE) are the weighted sums of loss- and gain-probabilities, where weighting is determined by the respective extant phase fractions. The third phase, austenite, is not represented in the phase evolution equations since there is the additional constraint that the sum of the phase fractions must be unity:

$$x^{+} + x^{-} + x^{A} = 1 \tag{2.1.2.3}$$

Similarly, the strain in a shape memory alloy may be written as the weighted sum of the strains in the individual phases present, as follows:

$$\varepsilon = x^{A}\left\langle \varepsilon^{A} \right\rangle + x^{+}\left\langle \varepsilon^{+} \right\rangle + x^{-}\left\langle \varepsilon^{-} \right\rangle \tag{2.1.2.4}$$

where

ε is strain;

x^{A} is the percent of the material in the austenite phase;

$\left\langle \varepsilon^{A} \right\rangle$ is the average strain of the austenite phase particles;

x^{+} is the percent of the material in the martensite (+) phase; and so on.

The above average strain values are determined using a standard, statistical formulation as follows:

$$\left\langle \varepsilon^{phase} \right\rangle = \int \varepsilon \cdot f(\varepsilon) d\varepsilon \tag{2.1.2.5}$$

where $f(\varepsilon)$ marks the probability of finding a layer with a strain 'ε'. In fact this is a probability density. Limits of integration for the above equation follow the range of stable states for the particular phase, i.e. the regions of strain where the second derivative of the Helmholtz free energy density function is positive.

The Boltzmann approach to the probability density function is used. This approach forms a probability based on the ratio of the material's potential energy with its mean kinetic energy from thermal excitation in the following way:

$$f(\varepsilon) = C \cdot \exp\left(-\frac{G(\sigma, \varepsilon, T)}{kT} \right) \tag{2.1.2.6}$$

where

$G(\sigma, \varepsilon, T)$ is the Gibbs free energy function;

k is the Boltzmann constant; and

C is a constant determined from the requirement that the integral of (2.1.2.6) be unity, i.e.

$$\int f(\varepsilon) d\varepsilon = 1 \tag{2.1.2.7}$$

From this, then,

$$C = \frac{1}{\int \exp\left(\frac{G(\sigma, \varepsilon, T)}{kT} \right) d\varepsilon} \tag{2.1.2.8}$$

Limits of integration again follow the range of stable strains. Thus, the probability $f(\varepsilon)$ may be written as

$$f(\varepsilon) = \frac{\exp\left(-\frac{G(\sigma, \varepsilon, T)}{kT} \right)}{\int \exp\left(\frac{G(\sigma, \varepsilon, T)}{kT} \right) d\varepsilon} \tag{2.1.2.9}$$

And, similarly, the appropriate value for the average strain of a phase $\left\langle \varepsilon^{phase} \right\rangle$ may be written as

$$\left\langle \varepsilon^{phase} \right\rangle = \frac{\int \varepsilon \exp\left(-\frac{G(\sigma, \varepsilon, T)}{kT} \right) d\varepsilon}{\int \exp\left(\frac{G(\sigma, \varepsilon, T)}{kT} \right) d\varepsilon} \tag{2.1.2.10}$$

Reiterating, for each phase, the average strain involves a ratio of integrals evaluated over the stable strain range for that phase. For example, the average strain of the austenitic phase is:

$$\left\langle \varepsilon^{A} \right\rangle = \frac{\int_{-w_a(T)}^{w_a(T)} \varepsilon \exp\left(-\frac{G(\sigma,\varepsilon,T)}{k_B T} \right) d\varepsilon}{\int_{-w_a(T)}^{w_a(T)} \exp\left(-\frac{G(\sigma,\varepsilon,T)}{k_B T} \right) d\varepsilon} \qquad (2.1.2.11)$$

The result of the above formulation is that, given an applied stress, overall strain can be calculated provided that the material phase fractions are known. The Gibbs free energy, itself a function of strain, temperature, and applied stress, is the fundamental quantity of these integral expressions. Thus, implicit in (2.1.2.4) is the material stress-strain relationship.

Finally, the model contains a balance of energy equation for determination of temperature changes in the alloy (see Equation (2.1.2.12)). Such changes may result from heat transfer with the environment, through direct joule heating, and through latent heats absorbed and released during the phase transformations:

$$mc\dot{T} = k\left(T - T_E(t)\right) + j(t) - \dot{x}^{+}H^{+}(\sigma) - \dot{x}^{-}H^{-}(\sigma) \qquad (2.1.2.12)$$

This equation makes this model uniquely suitable for actuator simulation, since it draws a direct link between joule heating, which is the control quantity for most SMA actuators, and the material strain in (2.1.2.4), which is representative of the actuator stroke.

Equations (2.1.2.2) and (2.1.2.12) embody the system of ordinary differential equations that constitute the SMA model. Because this ODE system exhibits a numerically stiff behavior, an unconditionally stable implicit Runge-Kutta scheme with an adaptive time step size is used for integration. The subroutine *RADAU5* was chosen for its efficiency and robustness. More details of the model can be found in [9,17].

This model forms the basis of the FE element. For the implementation, the material data are those of a typical NiTi wire, the most widely used actuator material.

2.1.3. Model behavior

Results of displacement controlled simulations of the SMA model at various temperatures appear in the figures below. Displacement is prescribed in the manner shown in Figure 10, except in the quasi-plastic case. Starting from an un-stretched condition, the simulated material is drawn out to some maximum extent and then permitted to return to a zero load condition. To effectively remove the self-heating and self-cooling phenomena associated with the release and absorption of latent heats, the simulation strain rate is very low, allowing ample time for temperature equilibration. This produces a nearly isothermal response. Further, the maximum displacement is chosen to guarantee a full phase transition to M_+ at full extension for the temperatures chosen.

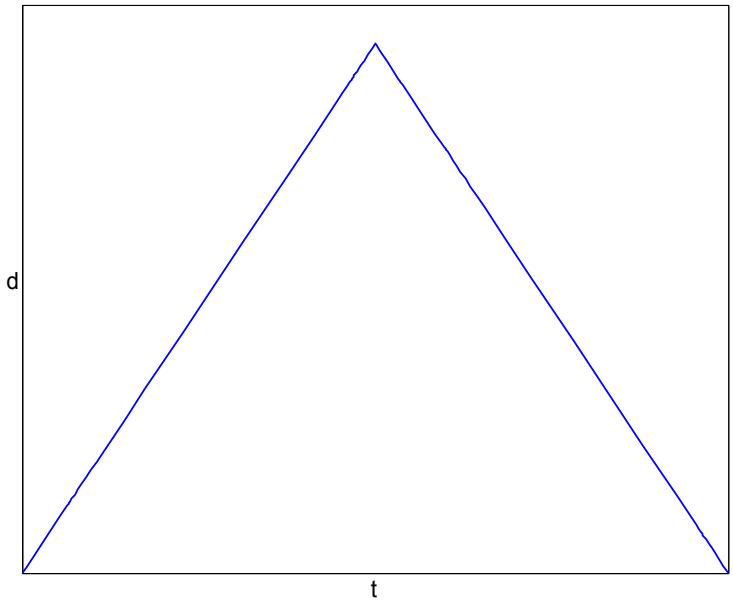

Figure 10: Prescribed displacement vs. time for model behavior description

At a low temperature, the quasi-plastic material behavior is intrinsically captured by the model. Figure 11 shows the simulated load-deformation result in the tensile quadrant only. The initial phase composition is assumed to be equal proportions of the two martensite variants.

Clearly, there is a remnant deformation as the material returns to a zero load condition, which dictates an early termination of the displacement curve of Figure 10. As shown in the figure, the initially linearly elastic material response is followed by a large displacement coinciding with a single load value, and then regains its elastic behavior on further displacement. The zero slope region of the plot corresponds to the phase transition where M_- lattice layers flip to the M_+ orientation. Upon unloading, these newly formed M_+ layers do not retransform to their original orientation, but instead find equilibrium in this configuration, leading to the remnant deformation. That the slope of the curve starting from the origin is the same as the slope along which the material returns to a zero load condition indicates that the initial phase composition was purely martensitic.

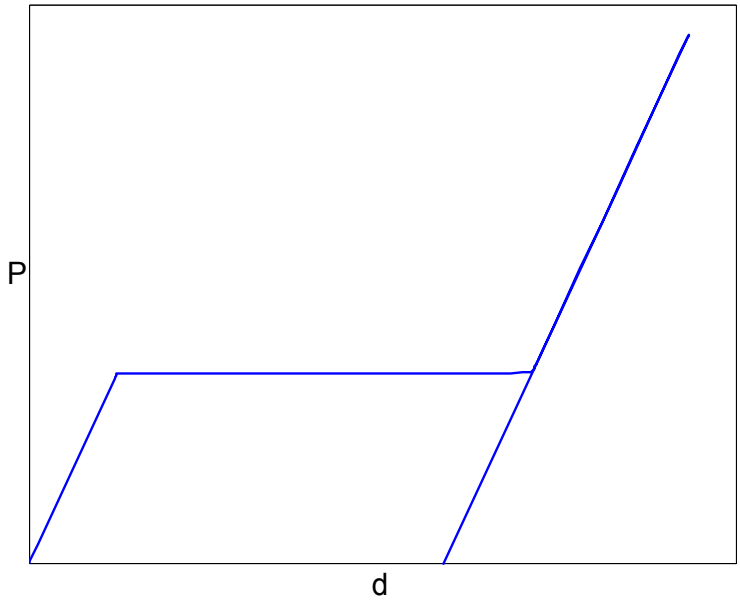

Figure 11: SMA model stress-strain curve, 273 K

Figure 12 shows the load-deformation curve at a higher temperature, where the material shows a pseudo-elastic response. Austenite is the stable phase under no load. Here again the initial response is linearly elastic, though the slope of the elastic region is much steeper than that observed at the lower temperature. This reproduces experimentally observed behavior and

indicates that a purely austenitic phase composition is significantly stiffer than a purely martensitic composition. Note that the load at which the A → M transformation occurs is higher than that shown for the quasi-plastic case. This demonstrates the model's ability to capture the temperature dependence of the transition stress. Once the A → M phase transition is complete, further deformation traces the same linear path as the lower temperature. However, as the deformation returns to zero, the material spontaneously undergoes a retransformation to austenite at a low load level. At the end of the simulation, the material has returned to its original, un-stretched and unloaded condition, at the same time regaining a purely austenitic phase composition. The hysteresis loop circumscribed by the prescribed deformation is quite clear.

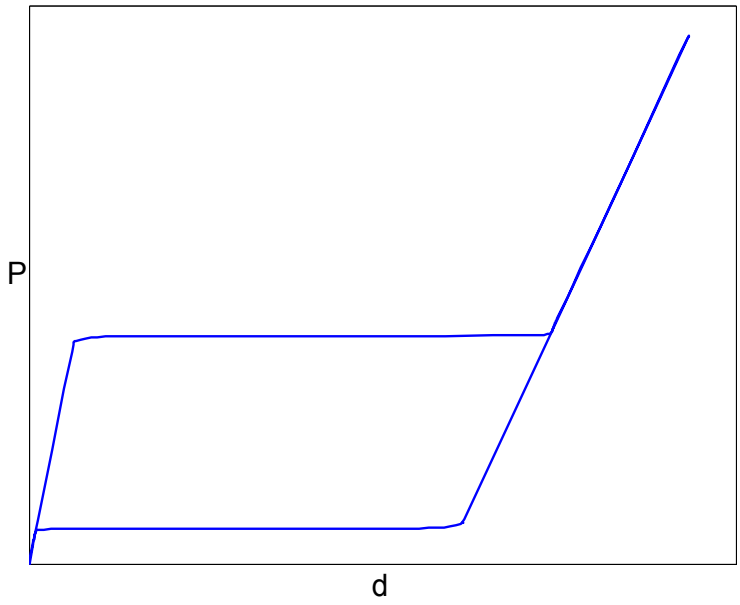

Figure 12: SMA model stress-strain curve, 313 K

The next plot shows the load-deformation curve of a simulation at yet a higher temperature. The salient differences in this case are the increases in the transition stresses. Both the A → M and the M → A phase transitions occur at substantially higher load levels than either of the two previous cases. However, the width of the hysteresis loop remains the same as the lower temperature, reproducing the experimentally observed behavior.

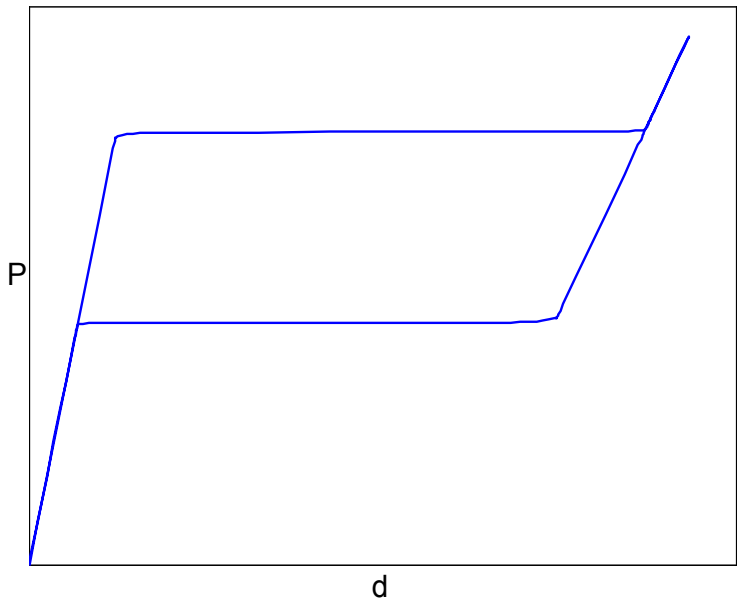

Figure 13: SMA model stress-strain curve, 353 K

These three simulations clearly demonstrate that the model is capable of reproducing the temperature-dependent nature of the material behavior, as well as the hysteresis loops produced during cyclic loading where phase transformations occur. It is the model's intrinsic ability to capture the temperature dependence of the material response, as well as its ability to calculate that response from a prescription of deformation, that makes it particularly suitable for use in representing a SMA actuator in a deformation-based finite element environment.

3. FINITE ELEMENT METHOD

By reducing the structural continuum into a finite number of discrete material elements for evaluation, the finite element method permits analysis of arbitrary structures, even those having irregular shapes, known material inhomogeneities, or other features that preclude the use of classical, analytical methods. The versatility of the method makes it the most widely used structural analysis tool.

After discretization, the approximate displacement of the structure at specific points (nodes) is calculated and translated by appropriate differentiation into a strain field within each element. From the strain, a stress distribution over each element is found, which, when integrated over the element, yields nodal forces. At each node, externally applied forces and nodal forces must satisfy equilibrium for a converged solution.

Because stress and stiffness of shape memory alloys depend on strain state and temperature, the global FE equations are nonlinear. Moreover, the hysteretic character of the material requires the loading to be applied incrementally through a series of time steps Δt.

In this chapter, after the global finite element solution equation is presented, the process by which ANSYS solves this equation is outlined with particular emphasis on both the interaction with the SMA model and the method by which convergence is established.

3.1. Governing equations

The global finite element equation serves as the means by which the response of any arbitrary structure can be analyzed. The weak form of the balance of momentum for the quasi-static case at time t_{n+1}= $(t_n+\Delta t)$ is given by (see [18] for the derivation):

$$\int \mathbf{B}^T \boldsymbol{\sigma}(\mathbf{a}_{n+1}) dv = \mathbf{P}^{ext}(\mathbf{a}_{n+1}) \tag{3.1.1}$$

where

B is the divergence or strain-displacement matrix,

$\boldsymbol{\sigma}$ is the Cauchy stress tensor,

$\mathbf{P}^{ext}$ is the external loading vector, and

a is displacement.

Clearly, both the stress and potentially the external loading are functions of displacement **a**, making this a nonlinear equation that cannot be explicitly solved, but instead requires an iterative solution method.

The Newton-Raphson iteration scheme represents the customary means of solving nonlinear equations. This scheme can be viewed as a linearization of the Taylor expansion of the equation. Let **R** be defined in the following way:

$$\mathbf{R} = \int \mathbf{B}^T \boldsymbol{\sigma}(\mathbf{a}_{n+1}) dv - \mathbf{P}^{ext}(\mathbf{a}_{n+1}) \tag{3.1.2}$$

Here, **R** is commonly called the 'residual' or 'unbalanced' load vector [19]. With the above definition, equation (3.1.1) becomes a homogeneous equation of the form:

$$\mathbf{R} = 0 \tag{3.1.3}$$

A Taylor expansion of equation (3.1.3) at time t_{n+1}= $(t_n+\Delta t)$ for iteration $(i+1)$ yields:

$$\mathbf{R}_{n+1}^{(i)} + \frac{\partial \mathbf{R}_{n+1}^{(i)}}{\partial \mathbf{a}_{n+1}^{(i)}}\left(\delta \mathbf{a}_{n+1}^{(i+1)}\right) + h.o.t. = 0 \tag{3.1.4}$$

where

$$\delta \mathbf{a}_{n+1}^{(i+1)} = \mathbf{a}_{n+1}^{(i+1)} - \mathbf{a}_{n+1}^{(i)} \tag{3.1.5}$$

and *h.o.t.* are higher order terms.

Neglecting any potential changes in external loading with changes in **a**, and noting that by definition $\mathbf{B} = \frac{\partial \boldsymbol{\varepsilon}}{\partial \mathbf{a}}$, then:

$$\frac{\partial \mathbf{R}_{n+1}^{(i)}}{\partial \mathbf{a}_{n+1}^{(i)}} = \frac{\partial \mathbf{R}_{n+1}^{(i)}}{\partial \boldsymbol{\varepsilon}_{n+1}^{(i)}} \frac{\partial \boldsymbol{\varepsilon}_{n+1}^{(i)}}{\partial \mathbf{a}_{n+1}^{(i)}} = \frac{\partial \mathbf{R}_{n+1}^{(i)}}{\partial \boldsymbol{\varepsilon}_{n+1}^{(i)}} \mathbf{B} \approx \int \mathbf{B}^T \frac{\partial \boldsymbol{\sigma}_{n+1}^{(i)}}{\partial \boldsymbol{\varepsilon}_{n+1}^{(i)}} \mathbf{B} dv \tag{3.1.6}$$

The integral term in the above equation is defined as the tangent stiffness:

$$\int \mathbf{B}^T \left(\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}} \right)\Bigg|_{n+1}^{(i)} \mathbf{B} dv \equiv \mathbf{K}_{n+1}^{(i)} \tag{3.1.7}$$

This tangent stiffness refers to the slope of the stress-strain curve at a particular strain value. For shape memory alloys, this slope ranges from values as high as the modulus of austenite (the stiffer of the phases) to near zero during stress-induced isothermal phase transitions.

Generally, then, linearizing the Taylor expansion by neglecting the higher order terms, for the *i-th* iteration of the time-discretized problem, the balance of momentum equation reads [20]:

$$\mathbf{K}_{n+1}^{(i)}\left(\delta \mathbf{a}_{n+1}^{(i+1)}\right) = \mathbf{P}_{n+1}^{ext} - \mathbf{I}_{n+1}^{(i)} \tag{3.1.8}$$

where $\mathbf{I}_{n+1}^{(i)} \equiv \int \mathbf{B}^T \boldsymbol{\sigma}\left(\mathbf{a}_{n+1}^{(i)}\right) dv$ is the internal load vector corresponding to the internal stress state .

Equation (3.1.8) reproduces the well-known Newton-Raphson iteration scheme. The equation is iterated until some convergence criterion is achieved, typically based either on a measure of the residual force vector **R** or on the size of the displacement increment $\delta \mathbf{a}_{n+1}^{(i+1)}$.

To aid in convergence behavior, a line search is used in determining the updated displacement. This method reduces the iteration increment $\delta \mathbf{a}_{n+1}^{(i+1)}$ in cases where the full increment may cause solution instability. The updated displacement is given by:

$$\mathbf{a}_{n+1}^{(i+1)} = \mathbf{a}_{n+1}^{(i)} + s\delta\mathbf{a}_{n+1}^{(i+1)} \tag{3.1.9}$$

where $0.05 \leq s \leq 1$ is the line search parameter.

The line search parameter s is determined by finding the solution of the equation given in [11] as:

$$\delta\mathbf{u}_{n+1}^{(i+1)^T}\left(\mathbf{P}^{ext} - \mathbf{I}_{n+1}\left(s\delta\mathbf{u}_{n+1}^{(i+1)}\right)\right) = 0 \tag{3.1.10}$$

Once the line search parameter is determined, the modified iteration increment $s\delta\mathbf{a}_{n+1}^{(i+1)}$ is used in the FE solution equation (3.1.8).

From this global FE equation, it is apparent that extensive communication between the global FE code and the SMA actuator model is necessary for solution. In particular, the SMA model must supply both the tangent stiffness **K** and the internal stress state $\boldsymbol{\sigma}$ for each iteration required at time t_{n+1} to the global FE solution algorithm, and for each iteration, the global FE code must supply all the necessary data to the model, such as time and strain increments and state variables. To ensure that this communication occurs smoothly and correctly, the performance of the implementation has been extensively investigated through comparison to a reference solution.

3.2. ANSYS solution process

The general solution process used by ANSYS in the simulation of SMA actuators in adaptive structures is outlined below. In order to solve the global FE equation, ANSYS performs an iterative process, calling the SMA model several times per iteration for updating solution information until convergence is achieved.

For the first sub-step of every load step, ANSYS determines the stress state and tangent stiffness of the SMA actuators using the converged displacement solution from the final sub-step of the previous load step and a zero displacement increment:

$$\begin{aligned} \boldsymbol{\sigma}_{n+1}^{(0)} &= \boldsymbol{\sigma}\left(\mathbf{a}_{n+1}^{(0)}\right) \\ \mathbf{I}_{n+1}^{(0)} &= \mathbf{I}\left(\mathbf{a}_{n+1}^{(0)}\right) \end{aligned} \tag{3.2.1}$$

where $\mathbf{a}_{n+1}^{(0)} = \mathbf{a}_n + \delta\mathbf{a}_n$. In the case of the first sub-step of the first load step, zero displacement and increment are assumed:

$$\mathbf{a}_0^{(0)} = \mathbf{0} \tag{3.2.2}$$

The stress and stiffness values are used to assemble the stiffness matrices and to calculate the internal restoring force vector **I**.

In subsequent sub-steps, stress and stiffness values corresponding to the converged solution from the previous sub-step are retained in memory, so that the initial call to the model is not required.

Together with the boundary conditions, the stiffness matrix and restoring force vector are used for determining the first iteration for the displacement increment:

$$\delta\mathbf{a}_{n+1}^{1} = \left(\mathbf{K}_{n+1}^{(0)}\right)^{-1}\left(\mathbf{P}^{ext} - \mathbf{I}_{n+1}^{(0)}\right) \tag{3.2.3}$$

The model is called again to update the stresses based on the new strain increment, from which the residual force vector is calculated:

$$\mathbf{R}_{n+1}^{(1)} = \mathbf{P}_{n+1}^{ext} - \mathbf{I}_{n+1}^{(1)} \tag{3.2.4}$$

If the convergence criteria are met, ANSYS again calls the SMA model with the converged strain and strain increment for updated values of the stress state and tangent stiffness, and proceeds to the next sub-step. If the convergence criteria are not satisfied, then the updated stiffness and stress values are used to calculate the second displacement increment iteration, $\delta\mathbf{a}_{n+1}^{2}$, and so on, until the convergence criteria are satisfied, or until the user-specified maximum number of iterations is reached. In the event that this maximum iteration number is met, then ANSYS exits the analysis with the error that the solution did not converge.

Determination of convergence depends on the criteria specified by the user for each load step. Choices include force convergence criterion, displacement convergence criterion, or a combination of the two. Within each criterion, a convergence tolerance must be prescribed to define the extent to which the solution must meet its criteria.

The force-based convergence method compares the residual force vector with a force criterion value. Specifically, for each iteration, a reference value is calculated from the square root sum of the squares of the applied force vector and the components of the residual force vector that correspond to constrained nodes:

$$F_{ref}^{(i)} = \sqrt{\sum_{j}\left(P_{j}^{ext}\right)^{2} + \sum_{k}\left(R_{k}^{(i)}\right)^{2}} \tag{3.2.5}$$

where *j* encompasses all nodes and *k* denotes constrained nodes only. The product of this reference value and the specified convergence tolerance marks the force convergence criterion:

$$F_{criterion}^{(i)} = F_{ref}^{(i)}\ cnvtol \tag{3.2.6}$$

To this, ANSYS compares a force convergence value, which is determined from the square root sum of the squares of the components of the residual force vector corresponding to degrees of freedom:

$$F_{val}^{(i)} = \sqrt{\sum_{l}\left(R_{l}^{(i)}\right)^{2}} \tag{3.2.7}$$

where *l* denotes nodes corresponding to degrees of freedom. Convergence is achieved when the force convergence value is less than the criterion.

The force convergence method is suited to instances where the slope of the load-deformation curve is steep near the strain increment in question. In this case, small changes in the displacement correspond to relatively large changes in the force, as illustrated by the red dashed curve in Figure 14; in this way the force provides a more sensitive measure of convergence than displacement.

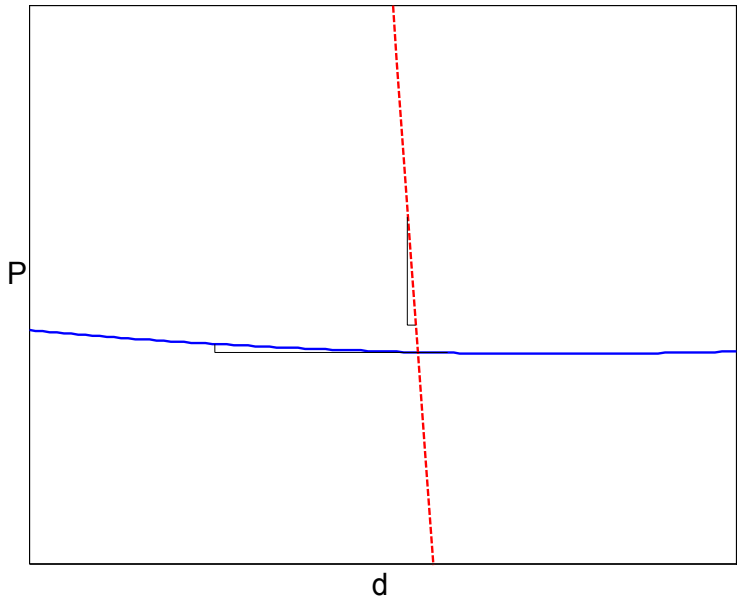

Figure 14: Convergence criteria selection: force - displacement curves

The blue curve in the figure represents an instance where the displacement based convergence method may be more suitable. Here a relatively small change in the force corresponds to a large change in the displacement, indicating a poor sensitivity in the force. Using the force convergence method in this case introduces the possibility for a potentially large error, since the range of force values satisfying convergence may permit an unacceptably large range of corresponding displacements.

The displacement-based method compares the computed displacement increment to a measure of the current displacement to determine convergence. Here, the reference value is found from the smallest of the infinitive, L1 and L2 norms of the displacement vector, and the

convergence criterion is the product of the reference value and the specified convergence tolerance:

$$u_{ref}^{(i)} = \min\left(\begin{array}{c} \max\left(\left|a_j^{(i)}\right|\right) \\ \sum_j \left|a_j^{(i)}\right| \\ \sqrt{\sum_j \left(a_j^{(i)}\right)^2} \end{array} \right)$$
$$u_{criterion}^{(i)} = u_{ref}^{(i)} \ \ cnvtol \tag{3.2.8}$$

The displacement convergence value is determined from the L2 norm of the displacement increment:

$$u_{val}^{(i)} = \sqrt{\sum_j \left(\delta a_j^{(i)}\right)^2} \tag{3.2.9}$$

When the displacement convergence value is less than the convergence criterion, then the solution is considered converged.

A combination of both force and displacement convergence criteria results in the requirement that both criteria be met for convergence, i.e.

$$F_{val}^{(i)} \le F_{criterion}^{(i)} \ \ \& \ \ u_{val}^{(i)} \le u_{criterion}^{(i)} \tag{3.2.10}$$

The solution process is illustrated in detail for a model problem in Appendices D and E, for both force driven and displacement driven cases, respectively.

To validate its performance, the implementation has been investigated with respect to the influence of user-specified solution parameters. For this, the simulation results are compared to a reference solution provided by the local-level version of the model, outside the finite element framework.

4. MODEL IMPLEMENTATION

As previously noted, it is evident from the global FE equation that solution data must be communicated back and forth between the FE level and the model or element level. In particular, element stiffness and stresses must be available to the FE code from the SMA model, which in turn requires updated values of time and strain increments, as well as current phase composition and temperature. The commercial finite element code ANSYS was chosen for the implementation of the SMA model largely for the flexibility it provides in defining the necessary links between the FE level equation and the local material constitutive law.

Specifically, this preprocessing, solution, and post-processing software package includes a family of elements, the 18x series, that are readily modified for user-defined subroutines through an element option called '*usermat*.' ANSYS is also attractive because of its extensive element library, nearly two hundred element types, which permits analysis of a wide array of systems ranging from structural, thermal, electromagnetic, and coupled-field systems. This coupled-field capability is especially attractive since it has the ability to handle multi-physics problems, like coupled heat transfer and mechanics. Both of these fields play an integral role in the behavior of SMA, as indicated by the nature of the ODE system that constitutes the SMA model. For the current implementation, however, ANSYS provides only structural results. Since heat transfer and temperature evolution are calculated at the model level, outside the FE framework, this implementation should be regarded as a preliminary step towards a multi-physics implementation.

To represent a SMA actuator, a three-dimensional link element (type LINK180) was modified via the *usermat* option to reproduce the material behavior defined in the Müller-Achenbach-Seelecke SMA model. This modification involved altering an ANSYS-supplied Fortran code, the *usermat* subroutine, that in this case serves as the intermediary between the FE level and the SMA model equations, translating the relevant input and output data into a format that can be understood at both levels. The SMA model itself required some minor revisions to

reformulate it from a *program* to a *subroutine* and to correctly interpret input from the *usermat* subroutine. Also, a macro was written that, in addition to all the modeling, solution, and post-processing steps needed in any typical analysis, specifies SMA material parameters and joule heating prescription. Details of the subroutines are discussed below.

usermat subroutine

In an effort to use the material model as a separate, self-contained core subroutine, the *usermat* subroutine functions primarily as the communication link between ANSYS and the SMA model. This practice is intended to facilitate updating the model with incremental improvements as they evolve without imposing the need to completely dissect the implementation.

Input and output data delivered to *usermat* and expected by the global FE code are listed and briefly explained in Appendix A. The primary data include the material parameters, state variables, strain and strain increment, and time and time increment.

A second function of the *usermat* subroutine is to explicitly calculate the material tangent stiffness. This can be accomplished analytically by finding the full derivative of the stress. The expression for stress is:

$$\sigma = \frac{E_M\left(\varepsilon - \left(x_+ - x_-\right)\varepsilon_T\right)}{x_+ + x_- + \frac{E_M}{E_A}\left(1 + x_+ + x_-\right)} \tag{4.1}$$

where

σ is stress,

ε is strain,

ε_T is the zero stress strain associated with a purely M_+ phase composition

E_m is the martensitic elastic modulus

E_A is the austenitic elastic modulus

x_i indicates the composition fraction of phase (i).

Noting that the stress is a function of strain and phase fractions, and that the phases themselves depend on strain through their dependence on the Gibbs free energy, then the differential stress is as follows:

$$d\sigma = \frac{\partial \sigma}{\partial \varepsilon}\bigg|_{x_1,x_2} d\varepsilon + \frac{\partial \sigma}{\partial x_1}\bigg|_{x_2,\varepsilon} dx_1 + \frac{\partial \sigma}{\partial x_2}\bigg|_{x_1,\varepsilon} dx_2 \tag{4.2}$$

From this, it becomes apparent that the full derivative of stress with respect to strain at time $t+\Delta t$ is given by:

$$\left.\frac{d\sigma}{d\varepsilon}\right|_{t+\Delta t} = \frac{\partial \sigma}{\partial \varepsilon} + \frac{\partial \sigma}{\partial x_1}\frac{dx_1}{d\varepsilon} + \frac{\partial \sigma}{\partial x_2}\frac{dx_2}{d\varepsilon} \tag{4.3}$$

As it is, equation (4.3) is not very useful since, although the stress can easily be differentiated with respect to strain or either phase fraction, the quantities $\frac{dx_1}{d\varepsilon}$ and $\frac{dx_2}{d\varepsilon}$ are not known.

However, it is possible to formulate a more useful expression by introducing differential time, in the following manner:

$$\left.\frac{d\sigma}{d\varepsilon}\right|_{t+\Delta t} = \frac{\partial \sigma}{\partial \varepsilon} + \frac{\partial \sigma}{\partial x_1}\frac{dx_1}{dt}\frac{dt}{d\varepsilon} + \frac{\partial \sigma}{\partial x_2}\frac{dx_2}{dt}\frac{dt}{d\varepsilon} \tag{4.4}$$

Or, in a more recognizable form:

$$\left.\frac{d\sigma}{d\varepsilon}\right|_{t+\Delta t} = \frac{\partial \sigma}{\partial \varepsilon} + \frac{\partial \sigma}{\partial x_1}\dot{x}_1\frac{dt}{d\varepsilon} + \frac{\partial \sigma}{\partial x_2}\dot{x}_2\frac{dt}{d\varepsilon} \tag{4.5}$$

The quantities $\dot{x}_1$ and $\dot{x}_2$ are the familiar phase evolution equations of the SMA model which can be easily calculated knowing current phase fractions and transition probabilities. The quantity $\frac{dt}{d\varepsilon}$ can be recognized as the inverse of the strain rate. This value can be approximated through a linearization using the time and strain increments of the current iteration.

The stress partial differentials are calculated explicitly, with the following results:

$$\frac{\partial\sigma}{\partial\varepsilon} = \frac{E_M}{x_+ + x_- + \frac{E_M}{E_A}\left(1 + x_+ + x_-\right)}$$

$$\frac{\partial\sigma}{\partial x_1} = \frac{-E_m\left(1 - \frac{E_M}{E_A}\right)\left(\varepsilon - \left(x_1 - x_2\right)\varepsilon_T\right) + \varepsilon_T\left(x_+ + x_- + \frac{E_M}{E_A}\left(1 + x_+ + x_-\right)\right)}{\left(x_+ + x_- + \frac{E_M}{E_A}\left(1 + x_+ + x_-\right)\right)^2} \tag{4.6}$$

$$\frac{\partial\sigma}{\partial x_2} = \frac{-E_m\left(1 - \frac{E_M}{E_A}\right)\left(\varepsilon - \left(x_1 - x_2\right)\varepsilon_T\right) - \varepsilon_T\left(x_+ + x_- + \frac{E_M}{E_A}\left(1 + x_+ + x_-\right)\right)}{\left(x_+ + x_- + \frac{E_M}{E_A}\left(1 + x_+ + x_-\right)\right)^2}$$

with variables as previously defined. A pseudo-code of a typical *usermat* subroutine is given in Appendix B.

SMA subroutine

Changes to the core material model were held to the minimum in order to facilitate introduction of improvements without rewriting the code altogether. The primary modification involved changing the code from a *program* to a *subroutine* that can be called from the *usermat* subroutine. The following changes were incorporated for this purpose:

1. The appropriate argument list is included in the subroutine statement.
2. The material properties, joule heating magnitude and duration, and tolerance for integration of the model's ODE system are correlated with those defined in the macro, via ANSYS.
3. The change in strain over the time increment is assumed to vary linearly. Roughly, the strain at any time between the starting and ending time of a particular call to the SMA model is defined as:

$$\varepsilon = \varepsilon_0 + \frac{\Delta\varepsilon}{\Delta t}\left(t - t_0\right) \tag{4.7}$$

where:

ε is strain,

ε_0 is the strain at the start of the time increment,

$\Delta\varepsilon$ is the strain increment,

Δt is the time increment,

t is time, and

t_0 is the time at the start of the time increment.

4. Since only stepped joule heating was applied for the examples in this study, the joule heating is activated by 'heating start' and 'heating end' times specified in the macro and included in the list of material properties, i.e.:

$$j(t) = \begin{cases} 0, & t < ht_{start} \\ j_0, & ht_{start} \le t \le ht_{end} \\ 0, & t > ht_{end} \end{cases} \tag{4.8}$$

where:

$j(t)$ is the joule heating as a function of time t,

j_0 is the magnitude of the heat pulse,

ht_{start} indicates the start time of the heating pulse, and

ht_{end} indicates the end time of the heating pulse.

5. The calculation of stress is redefined to account for the difference in initial strain between the ANSYS implementation and the local-level Fortran version. In ANSYS, the initial length denotes the length of the prestretched, fully martensitic (M_+) actuator, which is assumed to be in an unstrained state; in the local-level version, the prescription of a fully M_+ phase composition has the effect of introducing a prestrain equal to ε_T. Consequently, the strains may be compared by the following relation:

$$\left(\varepsilon\right)_{ANSYS} = \left(\varepsilon + \varepsilon_T\right)_{Fortran} \tag{4.9}$$

With this in mind, the stress must be calculated by the following equation:

$$\sigma_{ANSYS} = \frac{E_M\left(\left(\varepsilon + \varepsilon_T\right) - \left(x_+ - x_-\right)\varepsilon_T\right)}{x_+ + x_- + \frac{E_M}{E_A}\left(1 + x_+ + x_-\right)} \tag{4.10}$$

with variables as previously defined. Compare to equation (4.1).

The rest of the model code is left unaltered. Modified versions of the material model can easily be implemented by making the above modifications to the local-level version and recompiling the ANSYS executable file.

MACRO

Finally, the macro read by ANSYS for the simulations plays a multifaceted role. Tables defining material parameters and initial conditions are created. More importantly, the user-defined element type is specified in this file. Additionally, the file includes all the modeling, solution, and post-processing steps needed in any typical analysis.

The material property table is constructed using the '*TB,USER*' and '*TB,DATA*' commands. *TB,USER* indicates that the table contains material parameters for the element having a user-defined material behavior. '*TB,DATA*' provides a means to locate data sequentially in the table. The material parameters specified in the macro appear below:

1. Austenitic modulus of elasticity.
2. Martensitic modulus of elasticity.
3. Remnant strain in a pure phase quasi-plastic condition.
4. Volume of a material lattice layer.
5. Relaxation parameter.
6. Lower temperature, for determination of the transition stress.
7. Upper temperature, for determination of the transition stress.
8. Measured A → M transformation stress at lower temperature.
9. Measured A → M transformation stress at upper temperature.
10. Difference in stress levels for the A → M and M → A transformations.

11. Heat transfer coefficient.
12. Material specific heat.
13. Environmental temperature.
14. Material density.
15. Actuator radius.
16. Actuator initial length.
17. Tolerances for RADAU5, the integration scheme used for solving the model's system of ODEs.
18. Heat pulse starting time.
19. Heat pulse ending time.
20. Heat pulse amplitude.

The table for initial conditions is constructed using the '*TB,STATE*' and '*TB,DATA*' commands. The initial conditions include the starting martensitic phase fractions (the fraction of austenite is implicit) and actuator temperature, which need not be the same as the environmental temperature. However, the phase composition should be compatible with the initial temperature, e.g. the material should not be prescribed in a fully austenitic condition when the temperature is below that at which austenite is stable under no load. A pseudo-code for the macro file appears in Appendix C.

For actual use, these Fortran files must be compiled into an executable file through a compatible compiler, e.g., Compaq Visual Fortran. This is achieved by running the ANSYS-supplied file '*ANSCUST.bat*,' which finds all files in the directory with either a '*.C*' or '*.F*' extension, and then links them dynamically through the compilation. Running *ANSCUST.bat* from a command window allows viewing of errors in the compilation, critical knowledge in ensuring an operational executable. Once compiled, ANSYS must be opened through this executable in order to be able to call the user-defined material.

4.1. Implementation performance evaluation

A primary goal in this study is to scrutinize the interaction of the SMA model with the global finite element equation in order to verify the implementation. Results obtained from a simulation using a local-level model, which requires direct prescription of stresses or strains, provide a reference solution to which to compare the FE simulation results.

The system selected for comparison consists of a SMA wire actuator coupled to a linearly elastic spring (see Figure 15). This combination is a basic, yet very typical case of an actuator working against a structure that is mainly characterized by its stiffness. As such, it represents many of the qualities emblematic of an adaptive structure.

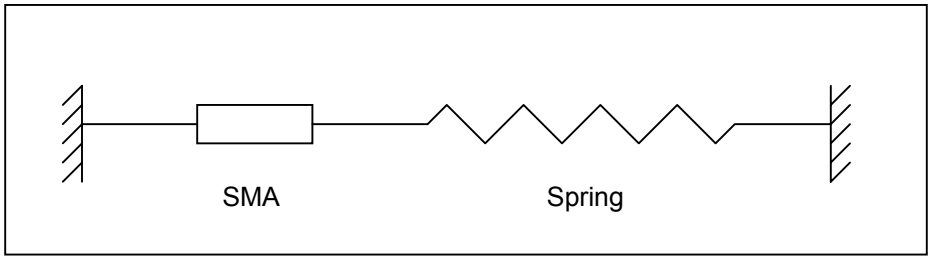

Figure 15: SMA - spring system

Actuation occurs through joule heating and convective cooling. Heating beyond the transformation temperature induces a phase transformation in the pre-stretched SMA from the initial martensite phase (M_+) to the austenite phase (A), a process accompanied by substantial contraction. While a SMA under a constant load exhibits a rather sharp transformation behavior, the increasing stress level provided by the force from the stretching spring, accompanied by a rise in the transformation temperature, extends the phase transformation over a larger range of temperatures, as shown in the temperature-strain plot Figure 16. Conversely, cooling renders the austenite phase unstable, leading to a re-transformation to martensite along a temperature path that also changes with loading. In contrast to unloaded cooling, during which self-accommodating martensite twins are likely to form, the tensile loading leads to the formation of only one variant as

the restoring force of the spring draws the actuator out to its starting length. For the simulations, the wire and spring are pre-tensioned.

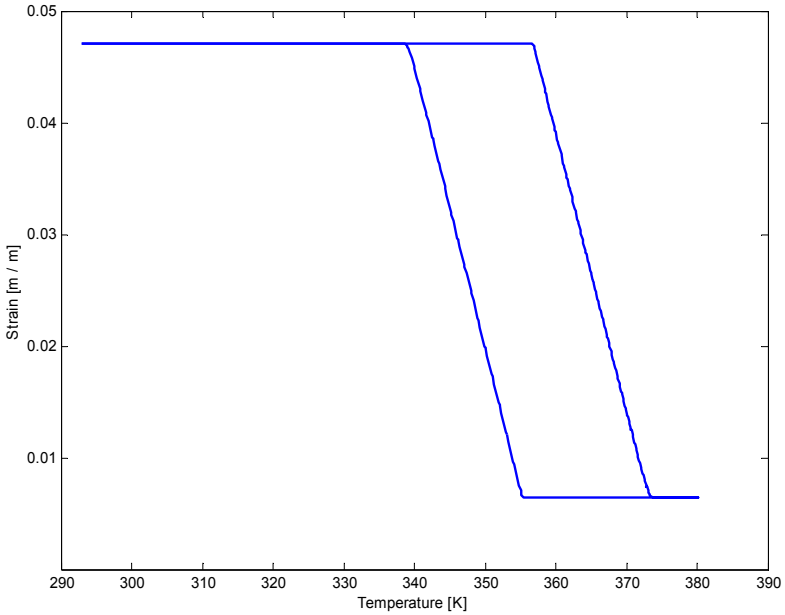

Figure 16: Typical strain-temperature plot in SMA-spring system

For the cycle, a hysteretic path is circumscribed. In the cases where the phase transformation in either direction is incomplete, partial loops within the outer loop are formed, so that the choice of paths within the strain range indicated in Figure 16 may result in any strain value. It is notable that temperature is not unique for a particular strain. As previously mentioned, this feature poses a unique challenge to conventional control schemes.

Actuation is controlled via the simple stepped joule heating application shown in Figure 17. The heating amplitude and duration are chosen to effect a complete transformation from martensite (M_+) to austenite for maximum actuator end-displacement. On the other hand, the wire temperature is closely monitored during the simulation so that heating amplitude and duration may be apportioned to ensure that it remains below a safe maximum level. Exceeding this level in a physical application can lead to degradation of performance as the material loses its shape recovery capability. The ability of the model to determine actuator temperature from the

prescribed joule heating, interaction with the environment, and latent heats absorbed or released during phase transformations makes this model particularly well-suited for representing the actuatoric qualities of SMAs in feasibility studies.

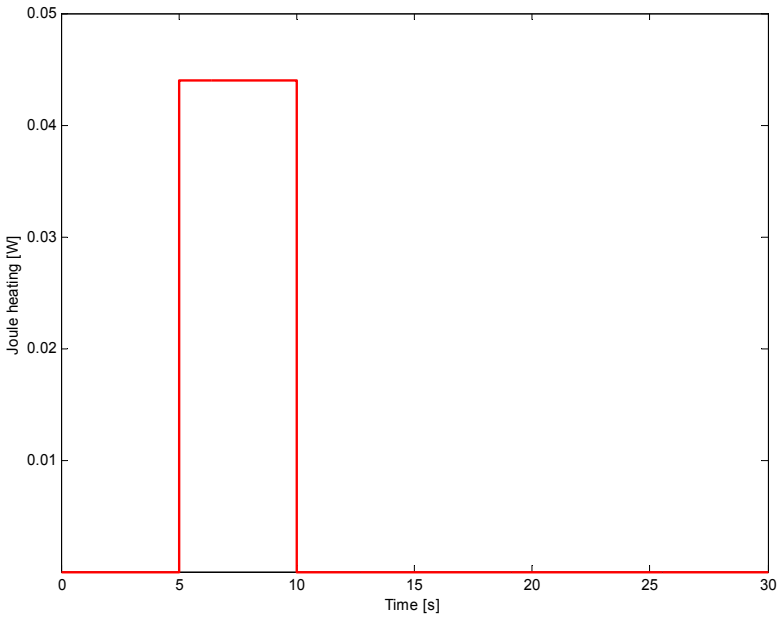

Figure 17: Heat pulse applied to the SMA actuator

Temperature, material phase, and displacement histories obtained from a simulation using the standalone implementation appear in the following figures and serve as the reference solution for the FE implementation.

Figure 18 shows the temperature history of the SMA wire over the single heating cycle shown above. Naturally, the wire temperature rises in response to the applied heating, up to a maximum approaching 380 K at the end of the heat pulse, and then returns to ambient temperature over time through heat transfer with the environment. These temperature changes have the effect of inducing phase transitions in the material, causing contraction of the actuator. Note, however, that the rates of heating and cooling differ from the conventional exponential behavior.

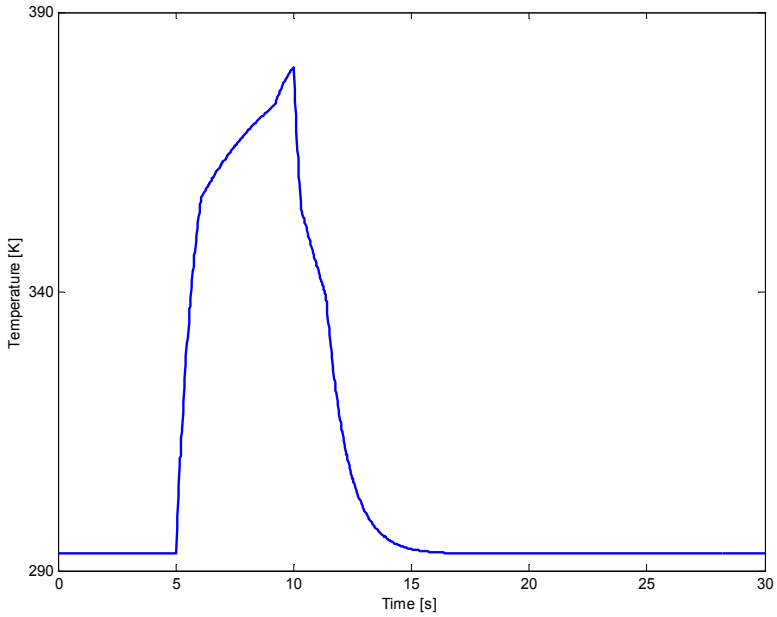

Figure 18: Resulting temperature history

The regions of reduced rates in warming and cooling correspond to the phase transitions, processes in which latent heats are absorbed or released (the phase evolution of the process is shown in Figure 19). During heating, for example, as the material transforms from martensite to austenite, latent heat is absorbed, effectively slowing the rate at which the wire temperature climbs. Once the phase composition becomes fully austenitic, the material ceases to absorb heat and the rise in wire temperature regains its exponential behavior. Similarly, during the austenite to martensite transformation in cooling, latent heat is released by the SMA, leading to the slowed rate of cooling. Clearly, the temperature intervals over which the phase transitions depend on whether the transition is M → A or A → M. These temperature ranges parallel those shown in the strain-temperature plot (Figure 16).

The displacement evolution depends entirely on the phase composition of the SMA, which is shown over time in Figure 19, below. Since the actuator is in tension throughout the simulation, only the M_+ variant and austenite interact. The times over which the transformations occur can be correlated with the intervals of slowed heating and cooling rates in Figure 18.

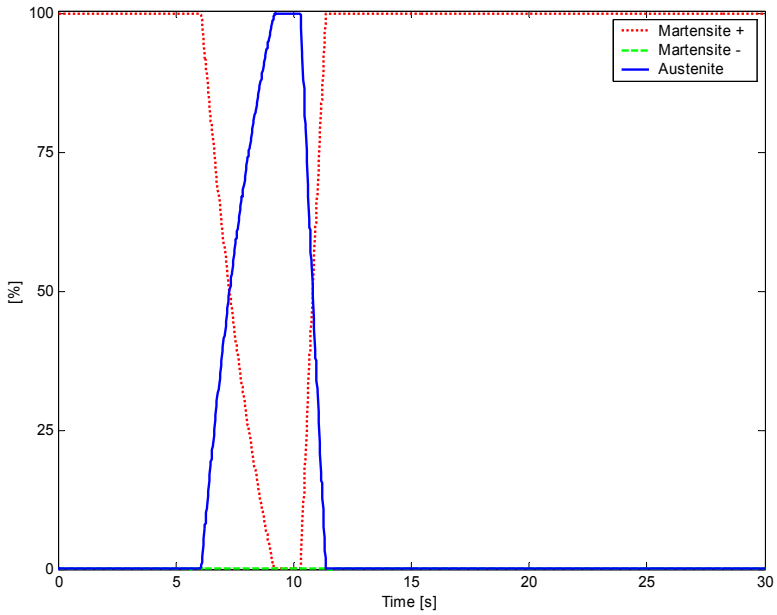

Figure 19: Phase evolution

The change in crystallographic shape that accompanies the phase transformation causes straining in the SMA wire, and consequently motion at the connection point between the wire and the spring, a convenient point of reference. This displacement is shown in Figure 20. Maximum stroke is achieved exactly at the time when the phase transition to austenite is completed. Similarly, the return of the wire to its original length signals the completion of the A → M transformation.

In the figure, zero displacement denotes the unstressed length of the purely M_+ wire. The level of pre-tension determines the offset from this zero value The high range of stroke possible with the 75 mm length of wire prescribed for the simulation is evident in the figure. At just over 3 mm, this stroke corresponds to a 4% strain, which is typical of SMA actuators.

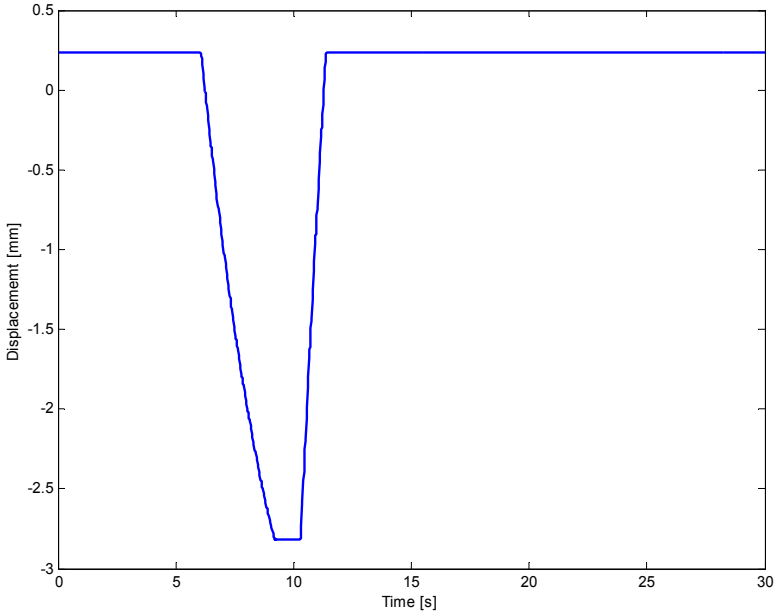

Figure 20: SMA actuator stroke vs. time

Results of a parametric study on the influence of user-prescribed solution variables appear below. This study is intended to validate the SMA model implementation into the finite element by comparing FE simulation results with the local-level model results described above. Specifically, the impact of the choice of convergence criterion, convergence tolerances, and time step size, on both the solution accuracy and computational cost are shown, as is the effect of tolerances in the implicit Runge-Kutta numerical integration scheme for the SMA model on global convergence.

A series of simulations varying the time increment establish that the number of iterations per time step tends towards unity with an increasing number of sub-steps under both displacement and force convergence criteria over a wide range of convergence tolerances. Figure 21 shows results for the case of displacement convergence criterion, Figure 22 for force convergence criterion. This trend is to be expected since, as the time step size is decreased, so too is the resulting equilibrium strain increment, with the result that the changes in the tangent stiffness and element stresses are smaller than they would be with a larger time step, all other things being equal. This in turn leads to a reduction in the number of required iterations to

achieve convergence. Comparing Figure 21 to Figure 22, The force convergence method holds a slight advantage in number of iteration per time step for a given tolerance.

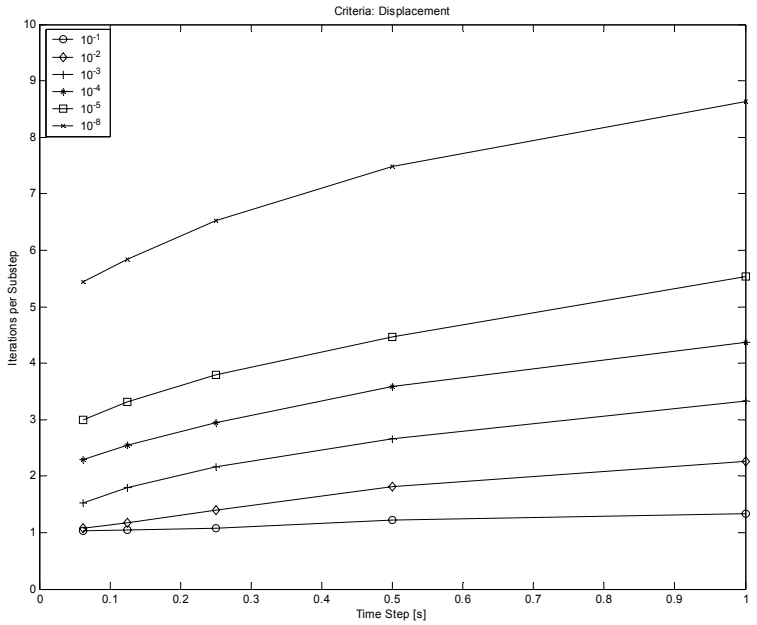

Figure 21: Iterations per time step, displacement convergence criterion

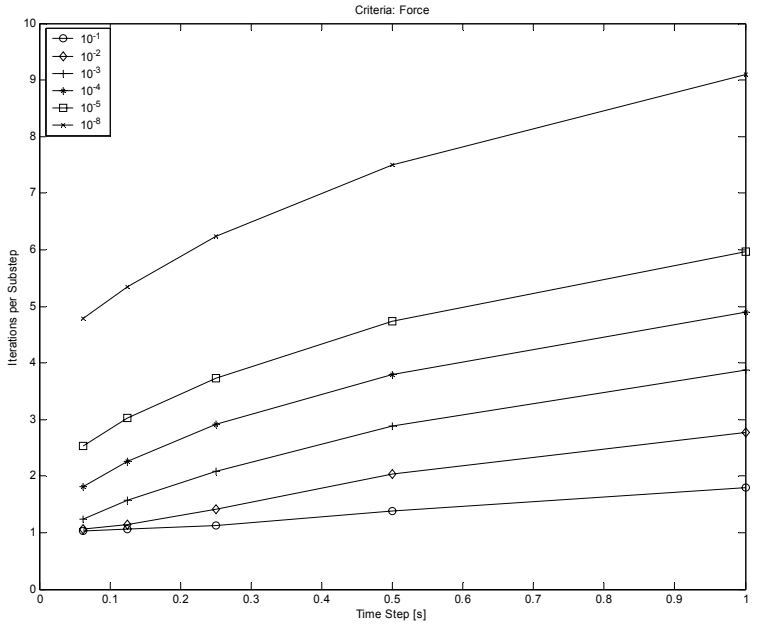

Figure 22: Iterations per time step, force convergence criterion

From one perspective, a small time step may be preferred for reducing the number of iterations per time step, which would seem to offer the possibility to reduce computational cost. However, increasing the number of time steps necessarily increases the total number of sub-steps. Though the number of iterations per time increment decreases, the product of the iterations per time step and the total number of time increments indicates an overall increase in the total number of iterations required for a simulation.

This total iteration number is closely related to computational cost, which may be measured by CPU time. Figure 23 and Figure 24 show how the computational cost varies with the same choices of time steps and convergence tolerances used in the determination of number of iterations per time increment, for displacement and force convergence criteria, respectively. In every case, a reduction in time step corresponds to an increase in CPU time. However, it should be noted that halving the time increment does not double the CPU time for the range of time steps chosen, except for the smallest time increment using the force convergence criterion.

The chosen convergence tolerance level also heavily influences the computational cost. Without exception, tightening the convergence tolerance increases the CPU time. This trend can be attributed to the reduction in the acceptable displacement or force imbalance.

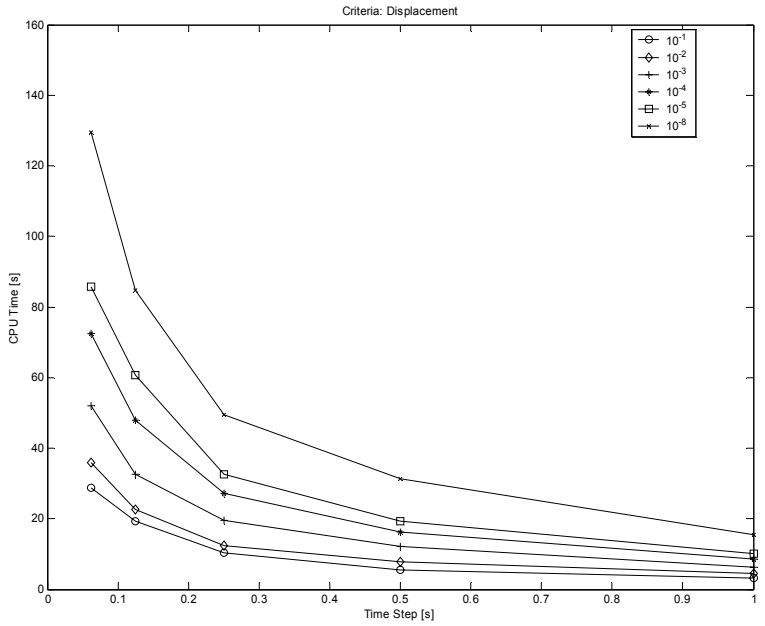

Figure 23: CPU time, displacement convergence criterion

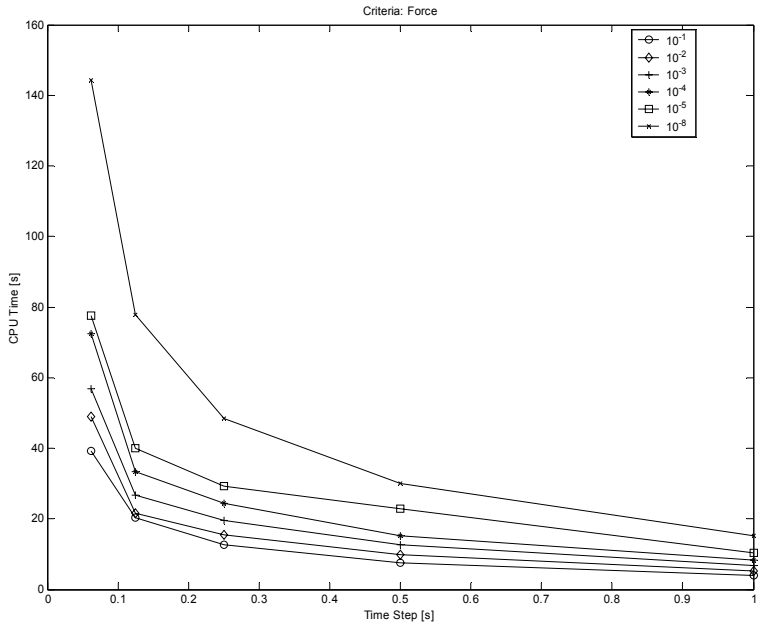

Figure 24: CPU time, force convergence criterion

Ultimately, however, in evaluating the performance of an actuator, the error in displacement is the primary quantity of interest since this gives a measure of the accuracy or exactness of the simulation. Defining a root mean square error as

$$e = \frac{1}{N}\sum_{n}\sqrt{\left(a_n - a_{n,reference}\right)^2} \tag{4.1.1}$$

where a is the displacement at the node connecting the SMA wire and the spring, $a_{reference}$ is the reference solution calculated with the local-level model version, *n* indicates the solution time, and *N* is the number of data points, the results show that the error is more sensitive to time step size than convergence tolerance. This is shown in Figure 25 and Figure 26 for both displacement and force convergence criteria, respectively. In both cases, initial tightening of the convergence tolerance reduces the error of the simulation, as does decreasing the time increment, except for the two spurious data points in Figure 25. However, tightening convergence tolerances beyond 10^{-4} does not notably improve the accuracy of the results, though, recalling the trends in computational cost, the tighter tolerances increase computational cost.

For a better interpretation of the results, the error values shown in the figures may be compared to the maximum end displacement of the SMA actuator, 3.05 mm. Normalizing the error by this displacement, an averaged error of 10^{-4} mm represents a 3.28% deviation from the reference solution, and an averaged error of 10^{-6} represents a 0.0328% deviation. Of course, the acceptable error is dictated by the specific application under investigation.

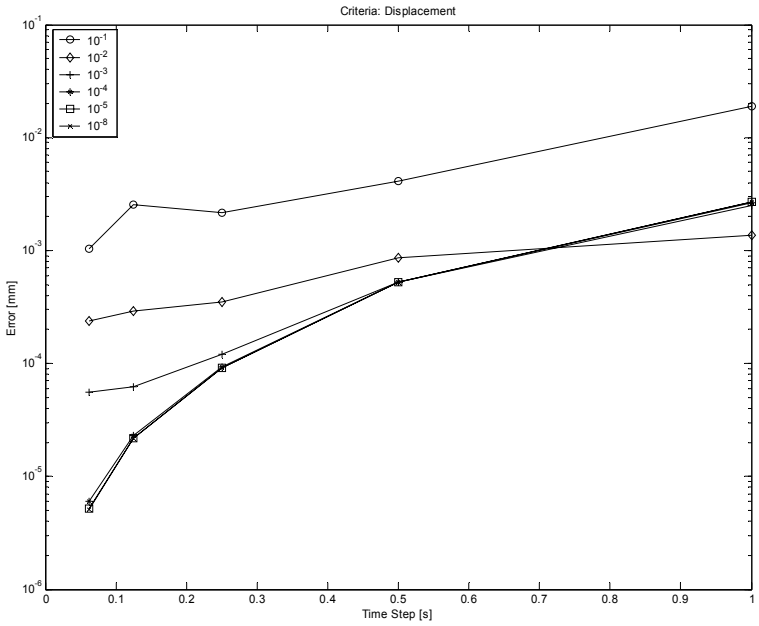

Figure 25: Averaged displacement error, displacement convergence criterion

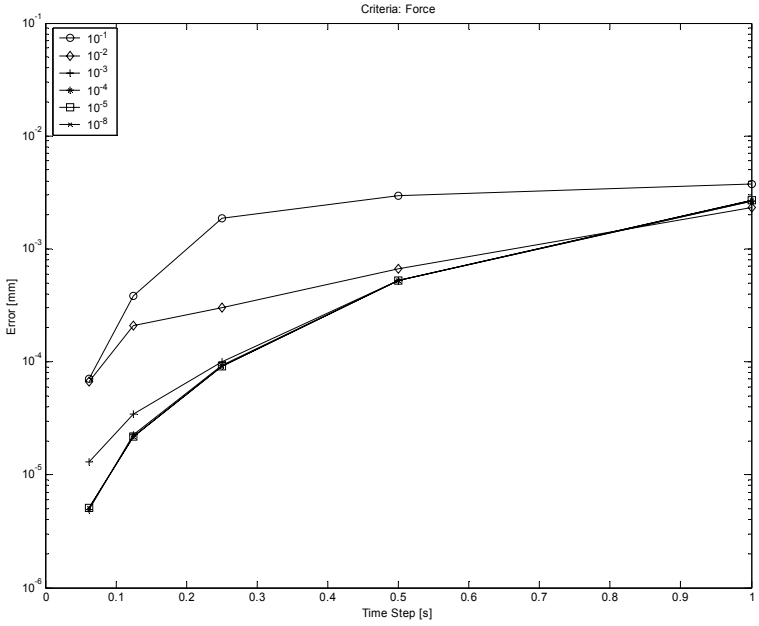

Figure 26: Averaged displacement error, force convergence criterion

The importance of the time increment may be understood by recognizing that path dependent behavior is resolved only as fine as the time step. In a material exhibiting strongly non-linear and hysteretic behavior, such as shape memory alloy, too large a time step exposes the possibility of skipping over important path dependencies. This is because global FE equilibrium is only required at the end of the time step, not throughout the increment. Due to the material path dependencies, the solution at the end of a time increment is directly linked to the straining process throughout the increment. As previously mentioned, for this implementation, strain is assumed to vary linearly through a time increment at the model or element level. A large time step size introduces a possibly large error resulting from this assumption. On the other hand, a very high number of sub-steps increases accuracy, at the cost of computational time. A compromise between CPU time and displacement error is desirable for this reason.

Figure 27 and Figure 28 show the displacement error curves superimposed on the CPU time curves for the two convergence criteria. These plots permit an assessment of the relationship between accuracy and cost; as such they might justifiably be called efficiency plots. For a particular acceptable level of error, the combination of solution parameters that minimizes computational cost can be determined from the plots. For example, if a 3.28% deviation is acceptable, then an average error of 10^{-4} is required. A glance at the efficiency plots indicates that CPU time can be minimized by choosing the force convergence criterion with a tolerance of 10^{-3}, using a time step of 0.25 seconds. If, however, an average error of $2x10^{-5}$ is required, then the plots show that a force convergence tolerance of 10^{-4} and a 0.125 second time step minimize the computational cost.

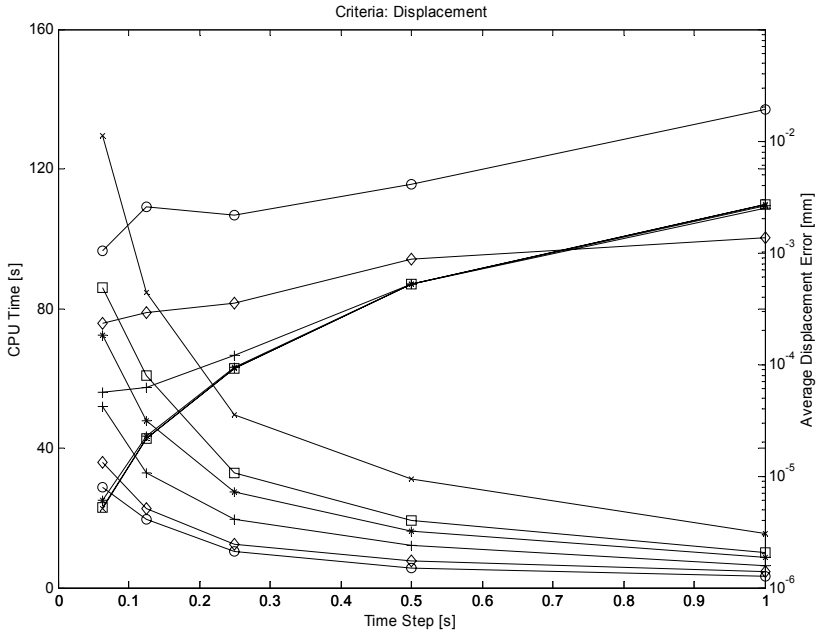

Figure 27: Efficiency plot, displacement convergence criterion

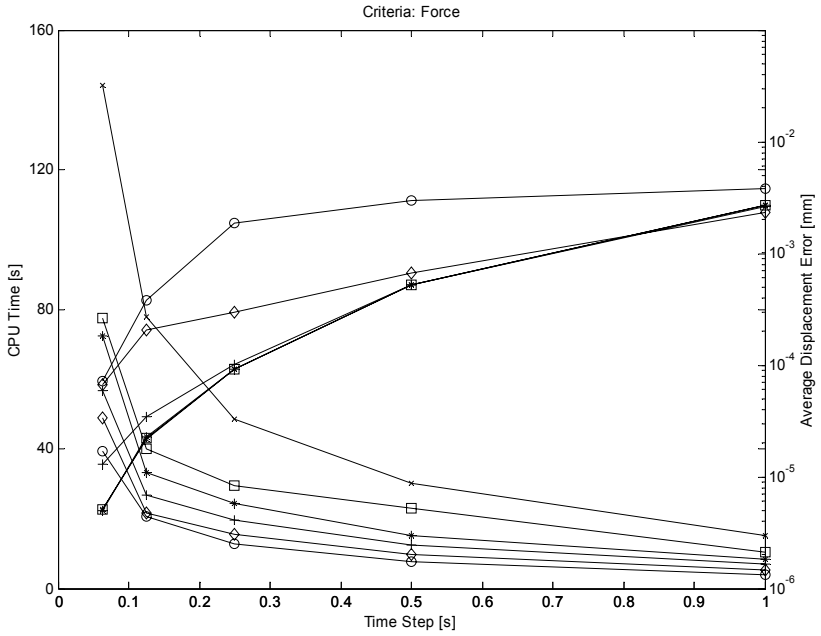

Figure 28: Efficiency plot, force convergence criterion

As a final comment it is worthwhile noting the effect of changing the tolerance levels in the implicit numerical integrator called by the SMA model. Figure 29, which shows results for a time increment of 0.25 seconds and a force convergence tolerance of 10^{-4}, illustrates the nearly 'on-off switch' influence that this tolerance has on global convergence behavior. Though the onset of instability at the FE level is very sudden, once below this level convergence occurs with little or no variation in the number of iterations required at the FE level. Similar behavior is seen with other time increments and convergence tolerances, though the critical tolerance value becomes smaller with the time increment and tighter convergence tolerance. Consequently, the tolerance for the implicit integral solver was set at a suitably low figure. For the simulations this tolerance was set at 10^{-8} for all time increments.

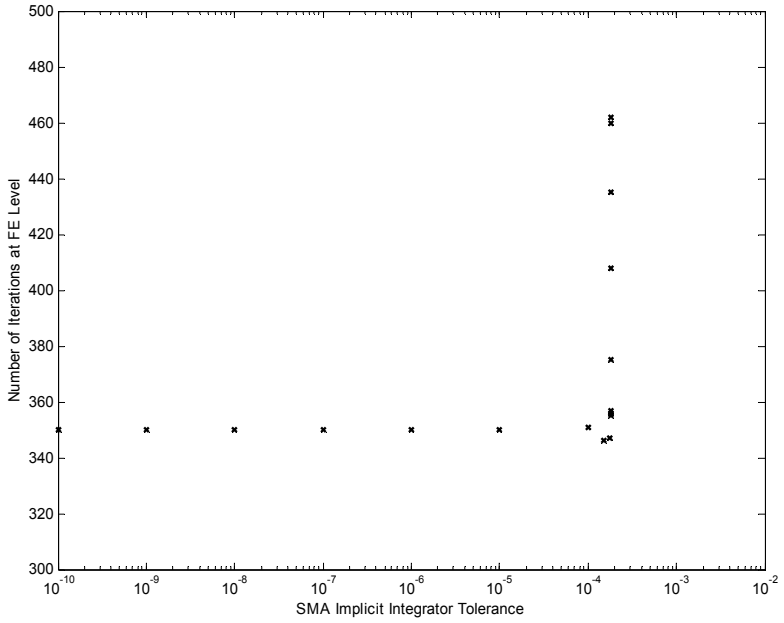

Figure 29: Influence of tolerance on Runge-Kutta integration scheme

5. OTHER APPLICATIONS

5.1. Adaptive beam

To attempt to simulate an adaptive structure with more than one SMA actuator, an adaptive beam actuated by two SMA actuators was modeled. This simulation approximately replicates a lab experiment carried out in [21], though exact material properties and dimensions are unknown. The system is comprised of two SMA wires mounted to stiff rectangular standoff stubs, which are in turn mounted to the flexible beam (see Figure 30).

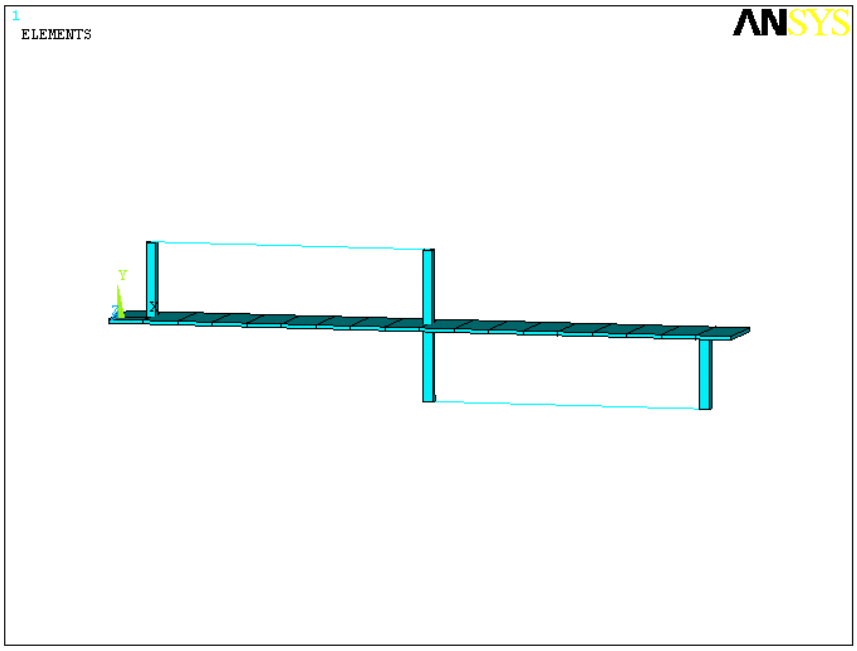

Figure 30: Beam discretization

Material properties and dimensions are shown in the table, below. Here, the flexible beam is represented by a typical thermoplastic material, and the SMA stubs are composed of aluminum. The SMA actuators, at 50 micrometers diameter, represent nearly the thinnest commercially available wire size.

Table 1: Adaptive beam material properties and dimensions

Flexible Beam		Units
Youngs modulus	2.756	GPa
Length	168.75	mm
Width	20	mm
Height	1	mm
Standoff Stubs		
Youngs modulus	70	GPa
Length	10	mm
Width	2.5	mm
Height	2.5	mm
SMA Actuators		
Length	75	mm
Diameter	0.05	mm

Joule heating is applied asynchronously, in the manner depicted in Figure 31. As shown, the right-hand actuator is heated at a relatively low power over a seven second interval, while the left wire is heated at a significantly higher magnitude during the interval over which the right wire is heated, but for only one and a quarter seconds.

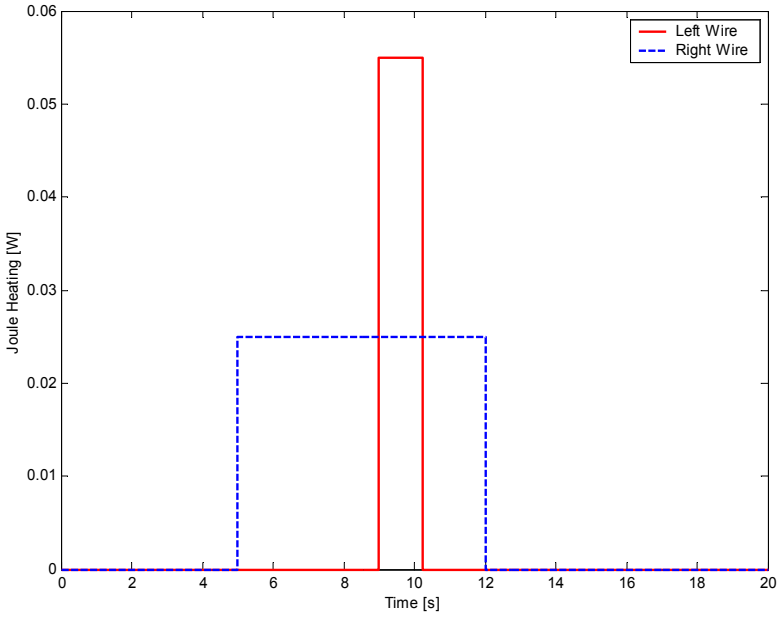

Figure 31: Beam simulation, joule heating prescription

In the simulations, the beam endpoints are constrained to horizontal movement, and the left endpoint is additionally prohibited from moving up or down. Convergence was based on the force criterion, with a tolerance of 10^{-4}, and a time increment of 0.25 seconds was used.

The resulting displacement at key times is shown in the following series of plots. Figure 32 shows the deformed shape after the right hand wire has been heated for four and a half seconds. As shown, it has partially transformed from martensite to austenite, but continues to contract with additional heating. Only the right half of the beam has been bent, since the left-hand wire has not yet been heated.

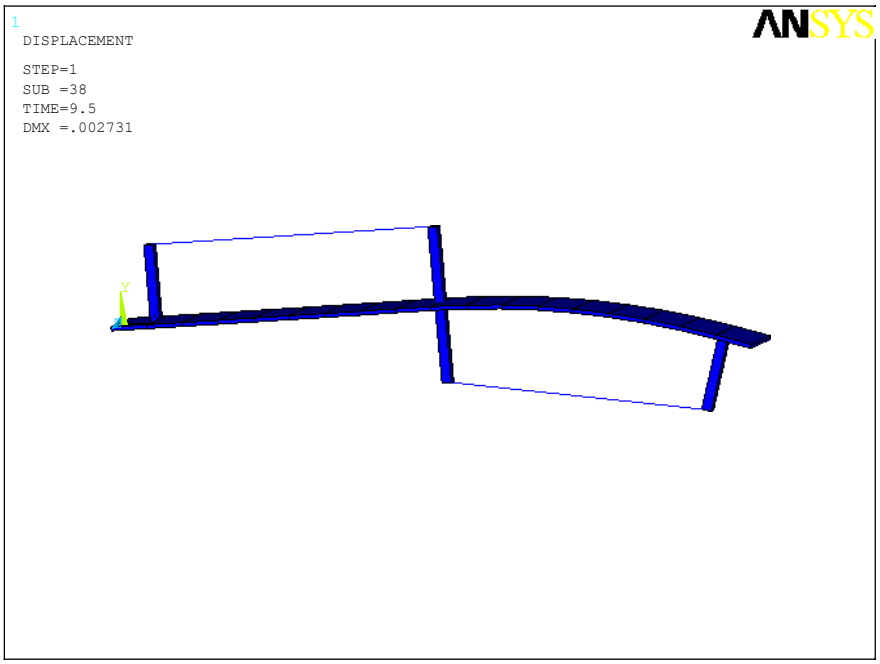

Figure 32: Beam deflection, right wire heated

Figure 33 shows the deformed beam while both actuators are being heated. At this point, the beam is bent into an ‘S’ shape approximating its second mode. As the left wire cools, the restoring force in the beam draws the actuator out to its original length, as shown in Figure 34 at 13.25 seconds into the simulation.

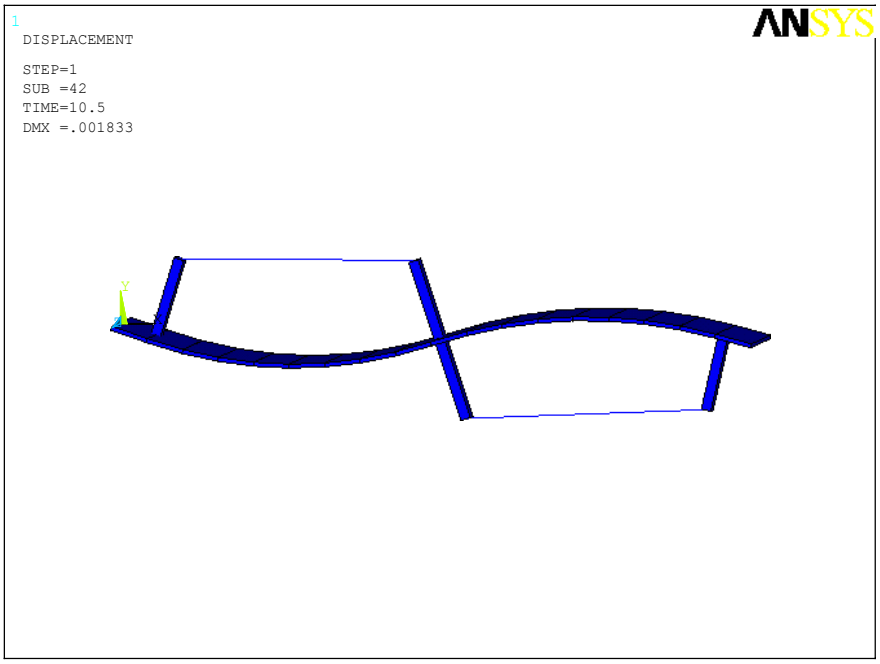

Figure 33: Beam deflection, both wires heated

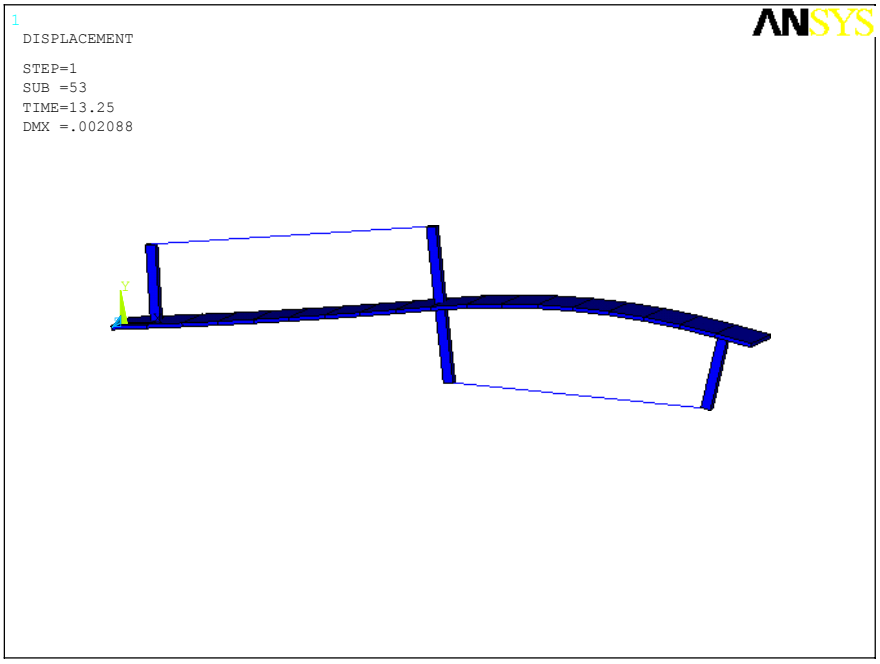

Figure 34: Beam displacement, left wire cooled

The right-hand wire is similarly draw out to its original length as it cools, so that, at the end of the simulation, the beam has returned to its original shape.

This relatively simple, two-actuator system confirms that the implementation is capable of analyzing several SMA actuators with differing heat applications, which is vital for many potential applications.

5.2. Aerospace application

From the validation of the SMA model implementation into the FE framework through a basic system, to a two-SMA actuator adaptive beam, consider now an illustration of the potential of the implementation in a more practical example of an adaptive structure. An aerospace application was chosen because of the preponderance of applications of adaptive structures in this field. Specifically, a flexible trailing edge of an airplane wing along the lines of a concept proposed by the Defense Advanced Research Projects Agency (DARPA), shown in the center of Figure 35, formed the basis of the simulation. Although the “Distributed Stack Concept” specifies piezoceramic actuation, for this study shape memory alloy actuators were used for actuation in this concept.

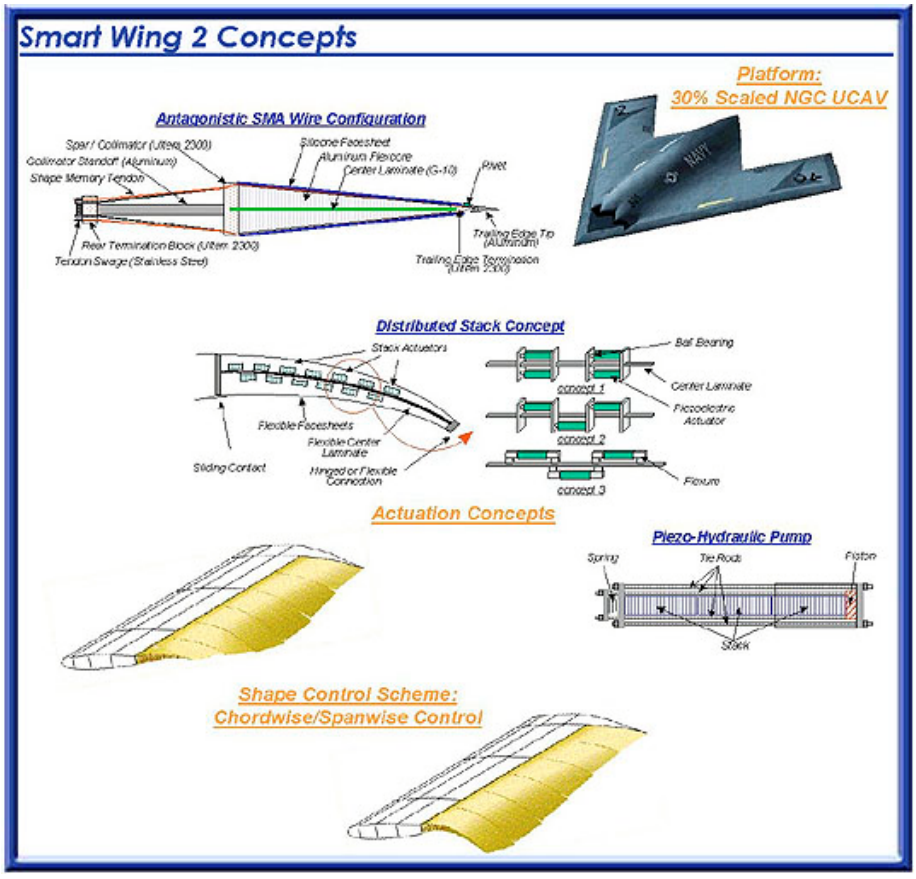

Figure 35: “Smart Wing” concepts [22]

The trailing edge section contains twelve SMA wires, six each above and below the center laminate, oriented in a manner similar to that shown in the “Distributed Stack Concept.” Material constants and dimensions used for the simulations appear in Table 2. The elastic modulus of the center beam and outer skins describes a typical polyimide thermoplastic. The total length of the trailing edge is prescribed at 420 mm, each of the SMA actuators having a length of 30 mm and diameter of half a millimeter.

Table 2: Trailing edge section material constants

	Elastic Modulus [GPa]	Thickness [mm]	Width [mm]
Center Beam	2.756	2.5	10.0
Skins	2.756	0.75	10.0
SMA Mounts	3.00E+04	25	10.0

The section is modeled with shell elements (SHELL63) and represents one of possibly many such sections continuously spanning the trailing edge of a wing. The model mesh and element types appear in Figure 36.

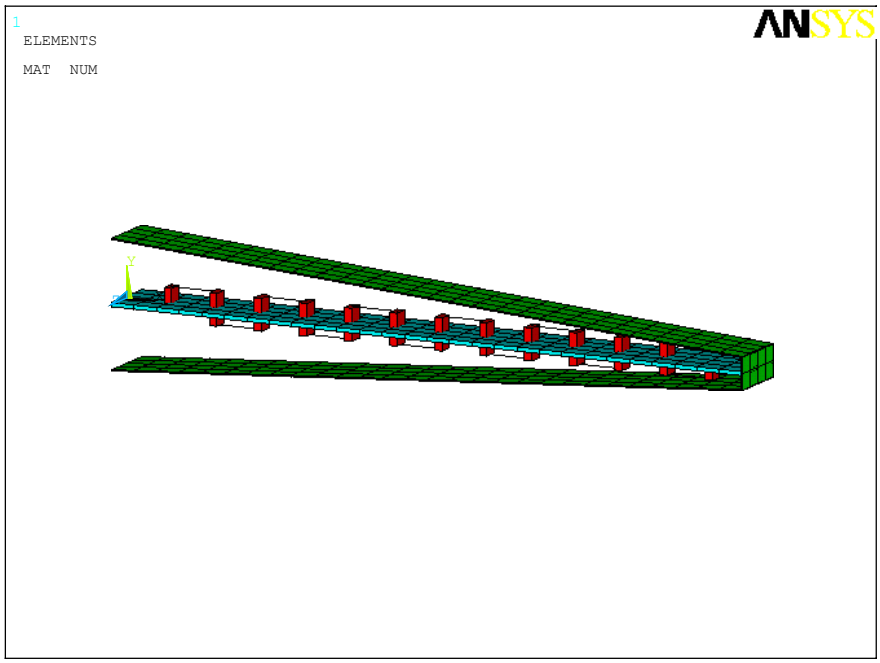

Figure 36: Trailing edge discretization

The black elements attached to the red beam elements represent the SMA actuators. Upper wires and lower wires are assigned unique element type numbers to allow different heating inputs to be applied to each set of wires. For the simulations, the left-hand node of the centerline beam is constrained in all directions, and the left-hand node of each of the wing skins is constrained to move only left or right.

The simulations use a time increment of 0.25 seconds and a displacement convergence tolerance of 10^{-4}. Heating is applied in an asymmetric manner as shown in Figure 37. All upper wires are heated at a relatively low power over a relatively long duration. As the upper wires are cooling, the lower wires are heated with a much higher power but for a shorter duration. Wire temperature in both cases never exceeds 380 K, preserving the performance characteristics of the wires.

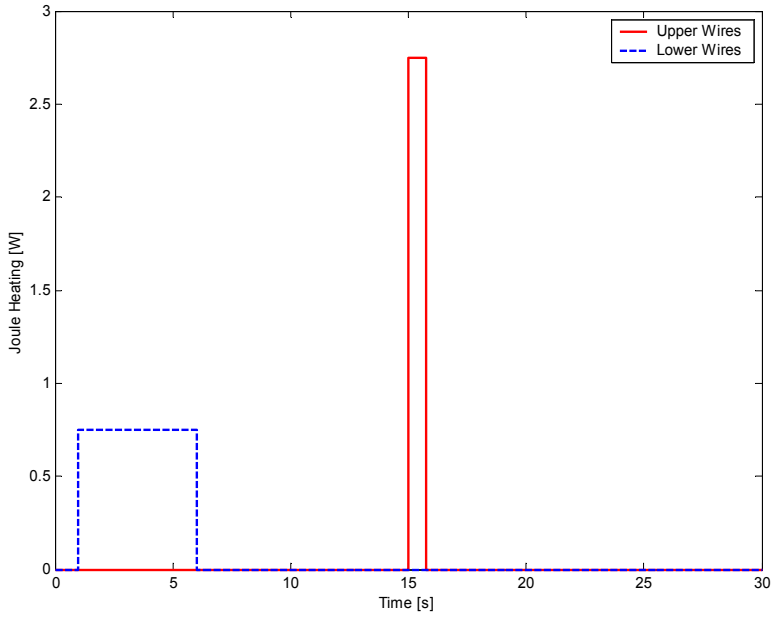

Figure 37: Trailing edge joule heating input

A convenient reference point to track the motion of the trailing edge is the centerline end node. The tip deflection history of this node appears in Figure 38. The maximum deflection is

nearly 13 cm, which, at over a quarter the overall length of the section, represents a sizeable amount.

Note that the heat pulse magnitude and duration influence the rate at which the section deflects. The first pulse, with a low magnitude and long duration, causes a deflection that increases over several seconds to its maximum. In contrast, the second, powerful but brief heat pulse results in a faster response, maximum displacement being attained within a second. These details indicate that the implementation permits calculation of the time-dependent reaction of an adaptive structure in response to a prescribed electric power input.

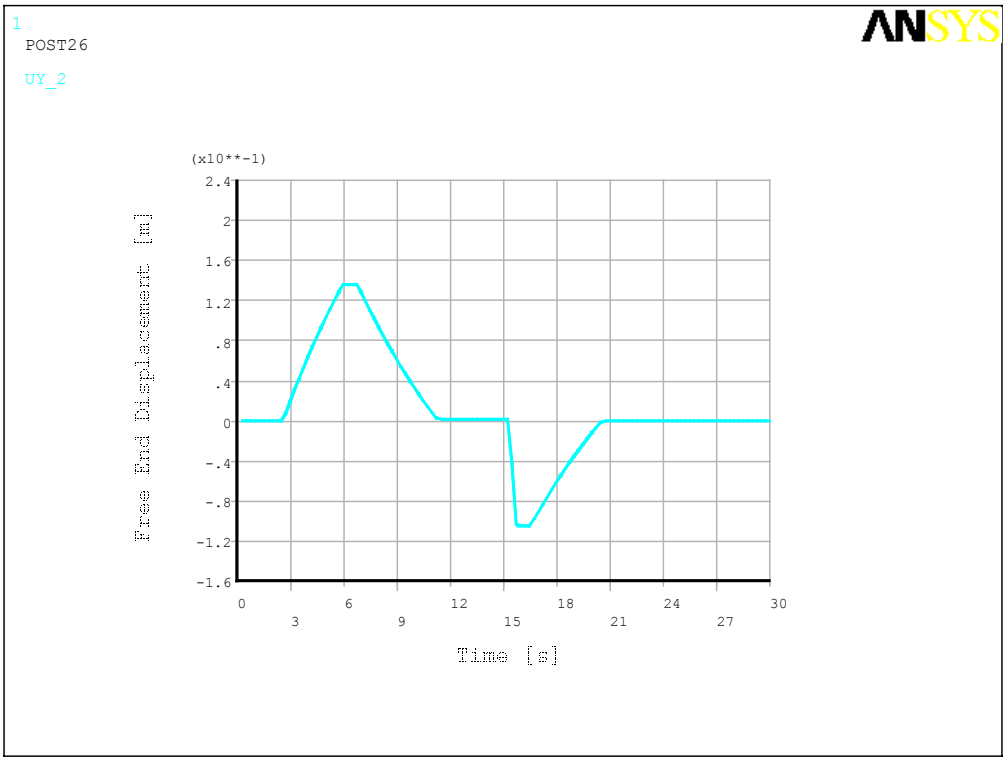

Figure 38: Wing tip deflection

Another interesting feature of the displacement plot is that the amplitude of the displacement on the downward side is less than the high side. This can be explained by the asymmetric fixation of the wires within the structure, which produces asymmetric displacement

behavior. Consider only the two SMA elements closest to the origin; the left-hand lower wire alone cannot produce the same end displacement that the left-hand upper wire alone can produce simply because the arm upon which it works is shorter. The same holds for each upper wire and its lower complement down the length of the section.

The deformed shapes corresponding to maximum deflections are shown below.

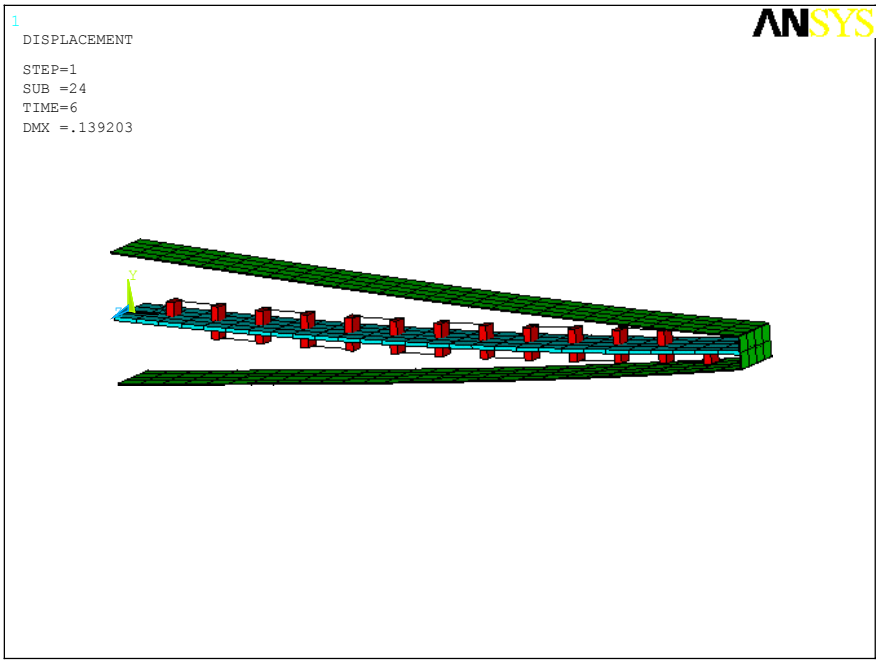

Figure 39: Trailing edge deformed shape, upper wires heated

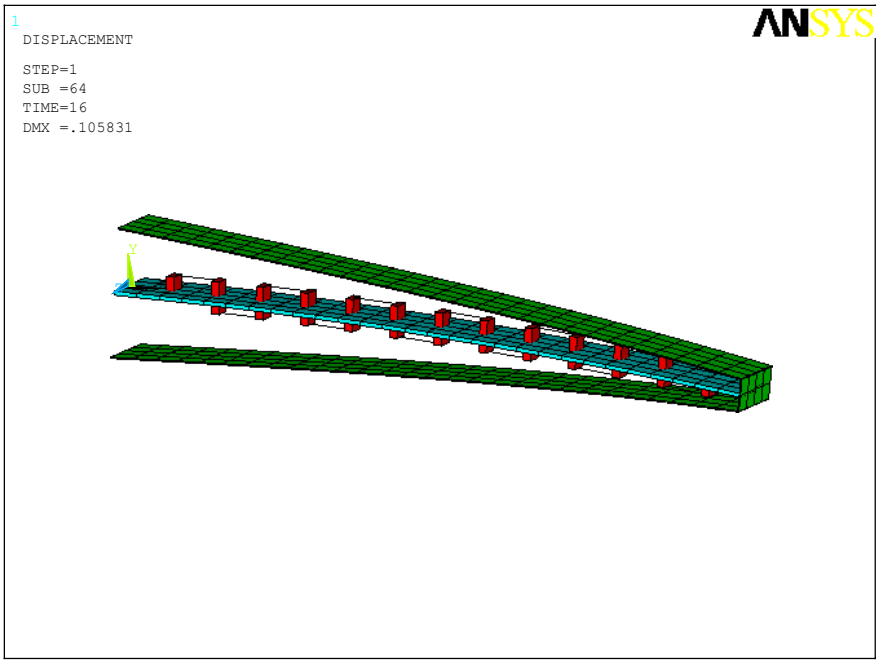

Figure 40: Trailing edge deformed shape, lower wires heated

This simulation demonstrates the ability of the implementation to simulate a relatively complex structure having multiple SMA actuator elements.

5.3. Micro-actuator

Shape memory alloys are not only attractive in aerospace applications, where their light weight and potential for structural integration are well-suited, but also potentially in micro-actuation devices. In these applications, their infinite resolution and clean-room compatibility present notable attributes. An added benefit in SMA micro-actuators is that, as the ratio of surface area to volume becomes smaller, the actuation driving frequencies can be much higher than in macro-scale applications. This is due to increase in the rate at which heat can be transferred to the environment; with a small surface area to volume ratio, the material cools with almost no gradient through the thickness. In contrast, macro-scale applications require longer cooling times since heat must not only be transferred from the surface, but also conducted from the interior volume.

The task for this simulation was to design a device to be fabricated of silicon with a thin SMA film deposited for actuation. The required total stroke of the interrogation region was set at 100-150 μm. The recommended configuration appears below, in Figure 41.

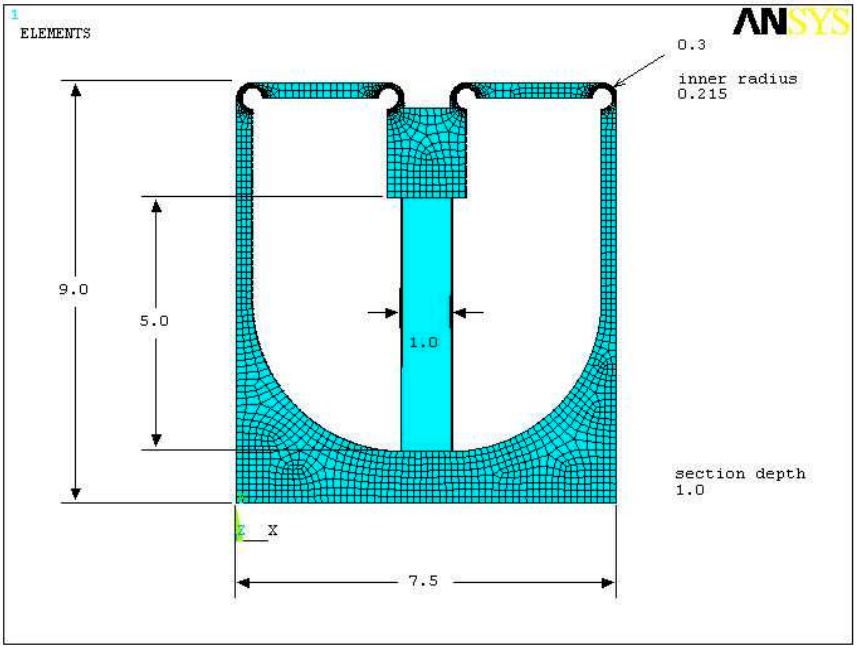

Figure 41: Micro-actuator dimensions (in mm)

The dimensions shown are in millimeters. The proposed thickness of the SMA film is about 2 μm.

To obtain this configuration, the device was simulated over a series of design iterations, where dimensions were minimized while maintaining the operational displacement requirement and not exceeding a critical stress level in the brittle silicon.

Although silicon is anisotropic in nature, for simplicity the material was approximated by an isotropic one with averaged material properties representing a crystal in the <100> plane, as given in [23]. The Youngs modulus was set at 150 GPa, and Poisson's ratio at 0.17. The silicon is modeled with solid elements, ANSYS type SOLID45.

Figure 42 shows the maximum displacement. At 125 μm, this falls within the specified requirement, though the figure could be increased with either a larger actuator, increasing actuation force, or larger device dimensions, increasing the leverage of the actuator.

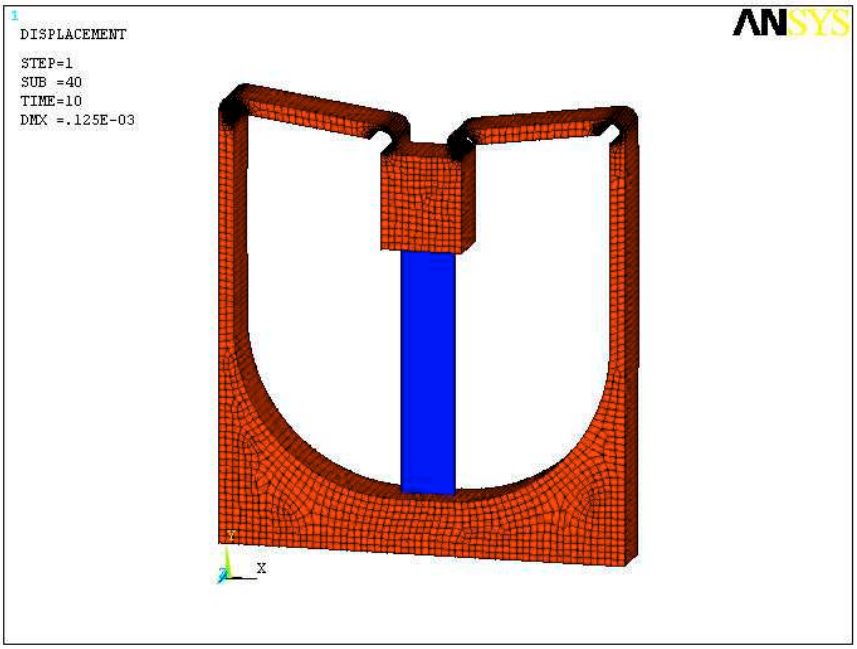

Figure 42: Micro-actuator, maximum displacement

However, since silicon is a brittle material, the stress level in the silicon plays an important role in this device. A second goal in the design of the device was to maintain the maximum stress level in the same order of magnitude as the failure stress nominated in [23]. The

von Mises stress is shown in Figure 43. The maximum stress, occurring at the flexing radius indicated by ‘mx’ in the figure, is about 530 MPa, which is less than twice the conservative 300 MPa failure stress promoted in [23]. A less conservative failure stress figure is about 7000 MPa [24]. With this in mind, for a first approach, the calculated maximum stress seems acceptable.

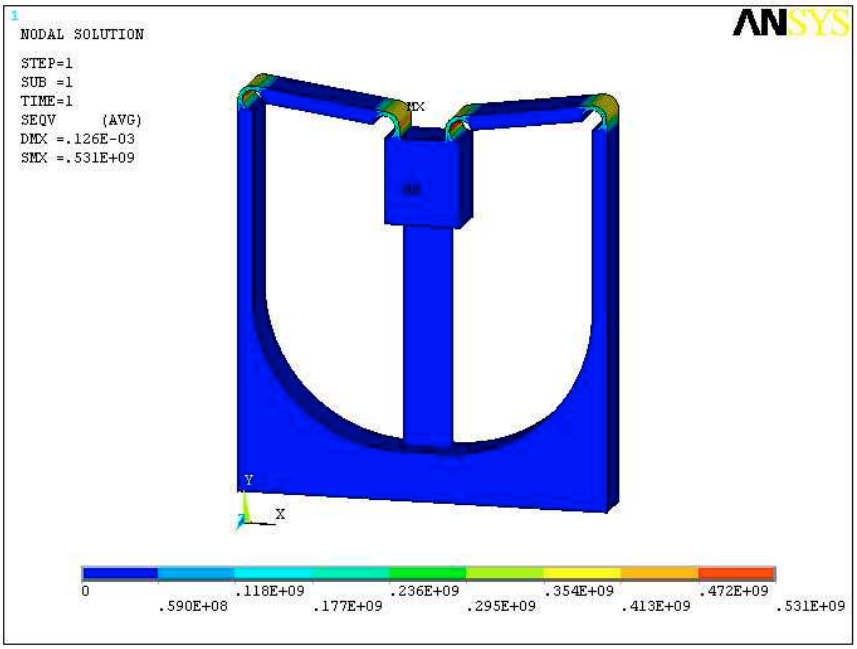

Figure 43: Micro-actuator, von Mises stress under maximum deformation

6. CONCLUSION

This thesis has presented the implementation of the Müller-Achenbach-Seelecke shape memory alloy actuator model into the commercial finite element code ANSYS. Finite element simulations of a SMA-spring system show that the implementation functions as intended. The parametric study of solution parameters suggest that accuracy and convergence depend not only on the chosen tolerance values both at the FE level and at the element level, but also on the time increment. Along with accuracy comes increased computational cost, which, in the manner presented in this study, may be minimized for a desired accuracy. The example simulations demonstrate the potential of performing feasibility studies of potential applications on multiple scales. The implementation represents a versatile and novel tool for the simulation of adaptive structural components using SMA actuators.

For future work, a more thorough study of convergence behavior for the more complex cases may provide additional insight into the relationship between the solution parameters. As mentioned previously, any improvements in the SMA model may be easily incorporated into the implementation. Use of the improved, multi-crystal SMA model will increase the ability of the implementation to simulate the experimentally observed behavior of SMA actuators. Finally, widening the scope of the implementation to a multi-physics environment, where both the structural response and heat transfer are analyzed at the finite element level may further increase the capabilities of the simulations.

REFERENCES:

1. http://smartmaterials.info/materials/comparison/stress_vs_strain.html
2. http://oea.larc.nasa.gov/PAIS/smart_plane.jpg
3. http://www.darpa.mil/dso/thrust/md/chap/briefings/timchap2000day1/Kudva.PDF
4. Kudva, J. N., Overview of the DARPA/AFRL/NASA smart materials and structures and Smart Wing program, *Proceedings of SPIE*, Vol. 4698, San Diego, 18-21 March 2002
5. Martin, C. A., Design, fabrication, and testing of scaled wind tunnel model for the Smart Wing Phase 2 program, *Proceedings of SPIE*, Vol. 4698, San Diego, 18-21 March 2002
6. Scherer, L. B., DARPA/AFRL Smart Wing Phase 2 wind tunnel test results, *Proceedings of SPIE*, Vol. 4698, San Diego, 18-21 March 2002.
7. Brinson, L.C., Lammering, R., *Finite Element Analysis of the Behavior of Shape Memory Alloys and their Applications*, Int. J. Solids Structures, 30, 3261-3280, 1993
8. Auricchio F and Sacco E, A temperature-dependent beam for shape-memory alloys: Constitutive modeling, finite-element implementation and numerical simulations, *Comp Meth Appl Mech Eng*, 174, 171–190, 1999
9. Seelecke, S, *Adaptive Structures with SMA Actuators – Modeling and Simulation* (in German), Habilitation Thesis, TU Berlin, 1999
10. S. Seelecke, N. Papenfuß, A Finite Element Formulation for SMA Actuators, *Journal of Applied Mechanics and Engineering*, Vol. 5, No.1, 2000
11. ANSYS User's Manual for Revision 6.1, Swanson Analysis Systems, Inc
12. http://www.mae.ncsu.edu/homepages/seelecke/JavaSimWWW/Sma_Sim_Background.html
13. Ortín, J., Delaey, L.: Hysteresis in shape-memory alloys, *International Journal of Non-Linear Mechanics*, 37, 1275-1281, 2002.
14. Funakubo, H. (ed.), *Shape Memory Alloys*, Gordon and Breach Science Publishers, 1984
15. Otsuka, K., Wayman, CM (eds): *Shape Memory Materials*, Cambridge University Press, UK, 1998

16. Achenbach, M., Müller I: Simulation of Material Behavior of Alloys with Shape Memory, *Arch. Mech*., 37(6), 573-585, 1985
17. Seelecke, S., Müller, I., Shape Memory Alloy Actuators in Smart Structures – Modeling and Simulation, *Applied Mechanics Review*, 2003, to appear
18. Zienkiewicz, O.C., Taylor, R.L.: *The Finite Element Method, Volume 1, Fourth Edition*, McGraw Hill, Berkshire, UK, 1989
19. Zienkiewicz, O.C., Taylor, R.L.: *The Finite Element Method, Volume 2, Fourth Edition*, McGraw Hill, Berkshire, UK, 1991
20. Bathe, K. J.: *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Englewood Cliffs, 1982
21. S. Seelecke, Control of Beam Structures by Shape Memory Wires, Proceedings of the 2nd Sci. Conf. Smart Mechanical Systems – Adaptronics, Otto-von-Guericke Univ. Magdeburg. 1997
22. http://www.darpa.mil/dso/thrust/md/smartmat/Programs/smartwing2.html
23. http://bell.mma.edu/~jmcent/siliconprop.htm
24. Petersen, K.E., *Silicon as a mechanical material*, Proc. IEEE. Vol. 70, No. 5, 1982
25. www.ansys.com/industry/nonlinear/ published_papers/usermat.pdf

APPENDIX A: ANSYS / element data exchange

The data passed between ANSYS and the modified element type link180 are listed below, along with a brief explanation of each item [25]:

Inputs to usermat.F:

1. matId: material number (integer variable)
2. elemId: element number (integer variable)
3. kDomIntPt: material integration point number (integer variable)
4. kLayer *: layer number (integer variable)
5. kSectPt *: section point number (integer variable)
6. ldstep: load step number (integer variable)
7. isubst: sub-step number (integer variable)
8. nDirect: number of direct stress components (integer)
9. nShear: number of shear components (integer)
10. ncomp: sum of nDirect and nShear (integer)
11. nstatev: number of state variables (integer) (defined by macro command TB,STATE)
12. nProp: number of properties defined in the macro (integer) (defined by command TB,USER)
13. Temp: temperature at the incremental start time (double precision variable)
14. dTemp: temperature increment (double precision variable)
15. Time: time at the start of the increment (double precision variable)
16. dTime: time increment length (double precision variable)
17. Strain: initial strain of time increment (double precision array)
18. dStrain: strain increment (double precision array)
19. prop: property values defined in the macro (double precision array)

20. coords: coordinates of integration points (double precision 3x3 matrix)
21. rotateM *: incremental rotation matrix (double precision 3x3 matrix)
22. defGrad_t *: initial deformation gradient (double precision 3x3 matrix)
23. defGrad *: final deformation gradient (double precision 3x3 matrix)
24. stress: stress state at Time (double precision array)
25. statev: state variables at Time (double precision array)
26. sedEl *: elastic work
27. sedPl *: plastic work
28. epseq *: = 0
29. tsstif *: transverse shear stiffness (double precision array)
30. epsPl *: plastic strains (double precision array)
31. variables *: other variables (up to 10) passed to the element from the macro (feature does not work)

Outputs from usermat.F:

1. stress: updated stress state at Time + dTime (double precision array)
2. statev: updated state variables at Time + dTime (double precision array)
3. sedEl *: see above
4. sedPl *: see above
5. epsPl *: see above
6. tsstif *: see above
7. keycut *: key for loading bisection control
8. dsdePl: Jacobian, tangent stiffness (double precision array)
9. epsZZ *: out of plane strain in plane stress problems only

Items marked with an asterisk play no role in the link180 element. The flow of data is shown in Figure 44, below.

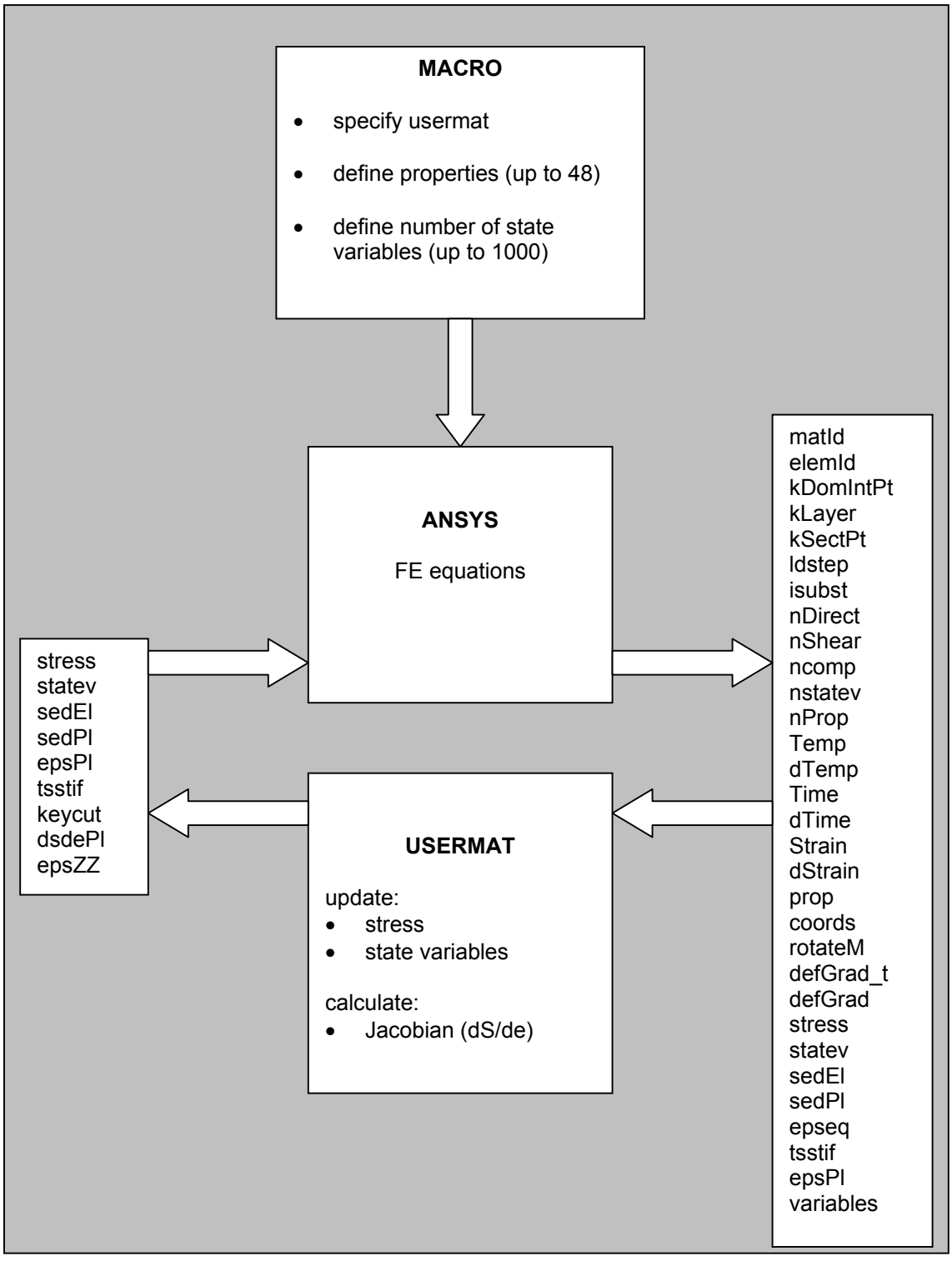

Figure 44: ANSYS data flow chart

APPENDIX B: usermat.F subroutine

The *usermat* subroutine in this implementation serves as a medium through which data is passed from the global FE level to the SMA material model, which remains essentially unchanged from its local form. As such, its primary function is to call the SMA model subroutine, passing all relevant data, and to translate the model's output into a form useful at the global FE level. This includes calculating the stress at the end of the time increment and determining the tangent stiffness. Initial conditions are specified in the macro through the 'TB,STATE' command.

A pseudo-code for the *usermat* subroutine appears below:

Usermat pseudocode:

```
subroutine usermat(
    &           matId, elemId,kDomIntPt, kLayer, kSectPt,
    &           ldstep,isubst,keycut,
    &           nDirect,nShear,ncomp,nStatev,nProp,
    &           Time,dTime,Temp,dTemp,
    &           stress,statev,dsdePl,sedEl,sedPl,epseq,
    &           Strain,dStrain, epsPl, prop, coords,
    &           rotateM, defGrad_t, defGrad,
    &           var1, var2, var3, var4, var5,
    &           var6, var7, var8, var9, var10)
    implicit none
    INTEGER
    &           matId, elemId,
    &           kDomIntPt, kLayer, kSectPt,
    &           ldstep,isubst,keycut,
    &           nDirect,nShear,ncomp,nStatev,nProp
```

```
      DOUBLE PRECISION
     &              Time,    dTime,   Temp,    dTemp,
     &              sedEl,   sedPl,   epseq
      DOUBLE PRECISION
     &              stress  (ncomp  ), statev  (nStatev),
     &              dsdePl  (ncomp,ncomp),
     &              Strain  (ncomp  ), dStrain (ncomp  ),
     &              epsPl   (ncomp  ), prop    (nProp  ),
     &              coords  (3),       rotateM (3,3),
     &              defGrad_t(3,3),    defGrad(3,3),
     &              var1, var2, var3, var4, var5,
     &              var6, var7, var8, var9, var10
c***************** User defined part ***********************************
      real*8 ea,ux,dux
      real*8 em,gamt,gam, y(3)
      real*8 pap,pam,pma,ppa
      real*8 nume,deno,dsde,dsdx1,dsdx2
      real*8 x1dot,x2dot,edot,xaus
      real*8 emea,test1,test2,test3
      real*8 yold(3),dx1de,dx2de
      real*8 part(3),dy(3),tst(2)
      real*8 endstrain,siga,sigm
      real*8 disp,ddisp
      integer numprop,numstatev,first,ii
      data first /0/
      data (yold(ii),ii=1,3) /3*0.0d0/
      if (first .eq. 0) then
      call opfile
```

```
      endif


      if (elemid .eq. 1) then
      first = first + 1
      endif
            ea = prop(1)
            em = prop(2)
            gamt = prop(3)
      do ii = 1,nStatev
         yold(ii) = statev(ii)
      end
      numprop   = nProp
      numstatev = nStatev
      keycut    = 0
      disp = Strain(1)*prop(19)
      ddisp = dStrain(1)*prop(19)
c*******************************************************
            call sma(statev,Strain,dStrain,ncomp,
     %                     Time,dTime,prop,numprop)
c*******************************************************
c CALCULATE Stress:
      y(1) = statev(1)
      y(2) = statev(2)
      y(3) = statev(3)
      endstrain = Strain(1) + dStrain(1)
      call stressstrain(stress,endstrain,y,numstatev)
      temp = y(3)
      call transstress(siga,sigm,temp)
```

```
      call pplsa (ppa,stress,sigm,temp)
      call papls (pap,stress,siga,temp)
      call pmnsa (pma,stress,sigm,temp)
      call pamns (pam,stress,siga,temp)
c CALCULATE Jacobi-Matrix:
      xaus  = 1.0d0-y(1)-y(2)
      emea  = em/ea
      nume  = Strain(1) + dStrain(1) - (y(1) - y(2))*gamt
      deno  = y(1) + y(2) + em/ea*(xaus)
      dsde  =  em/deno
      dsdx1 = -em*((1.0d0-em/ea)*nume/deno/deno + gamt/deno)
      dsdx2 = -em*((1.0d0-em/ea)*nume/deno/deno - gamt/deno)
      dx1de = (y(1)-yold(1)) / dStrain(1)
      dx2de = (y(2)-yold(2)) / dStrain(1)
      x1dot = -y(1)*ppa + (xaus)*pap
      x2dot = -y(2)*pma + (xaus)*pam
      edot  = dStrain(1)/dTime
c 'exact' Jacobi matrix
      if (dTime .eq. 0.0d0 .or. dStrain(1) .le. 1.0d-20) then
        dsdePl(1,1) = dsde
      else
      dsdePl(1,1)= max(dsde+dsdx1*x1dot/edot+dsdx2*x2dot/edot,5.0d8)      c (1)
      endif
      return
      end
```

(1) This minimum bound is introduced for numerical stability.

APPENDIX C: macro pseudo-code

The macro serves several purposes. First, it indicates that a user defined element will be used. Second, it defines at table listing all the necessary material properties the SMA model requires. Third, it defines the solution parameters, i.e. analysis type, convergence criteria, convergence tolerances, iteration scheme, time step, and load steps. Also, the system under inquiry is modeled at the macro level, and all necessary boundary conditions are defined.

A pseudo-code for a typical macro appears below.

Macro pseudo-code:

```
/PREP7
        ET,1,link180
        R,1,1e-6
!
! define SMA material parameters, put them into a table, and define initial state variables:
!
        ea      = 71.1e9
        em      = 30.9e9
        gamt    = 0.044
        vs      = 5e-23
        taux    = 1e-2
        tl      = 323
        tu      = 353
        tauatl  = 431.58e6
        tauatu  = 663.16e6
        delta   = 295e6
                htcoeff  = 2.3e1
```

```
cv      = 0.45e3
        extemp = 273
xp0     = 0
xm0     = 0
t0      = 273
rho     = 6.4e3
radius  = 9e-7
length0  = 6.35e-2
atol    = 1e-6
rtol    = 1e-6
!
mat,1
tb,user, 1 , 1, 21
        tbdata , 1  , ea    , em    , gamt   , vs      , taux    , tl
        tbdata , 7  , tu    , tauatl  , tauatu  , delta, htcoeff  , cv
        tbdata , 13, extemp, xp0, xm0    , t0      , rho     , radius
        tbdata , 19, length0, atol, rtol
tb,state,1,,3
        tbdata , 1  , xp0     , xm0   , t0

tblist,user
!
! create the model and mesh it
!
        K,1,0,0,0,
        K,2,1,0,0,
        LSTR,1,2
        LESIZE,ALL, , ,1, ,1, , ,0,
```

```
        LMESH,1
!
! define solution parameters, boundary conditions, and applied forces / displacements for
! load steps
!
        ANTYPE,0
        AUTOTS,Off
        NROPT,AUTO, ,OFF
        cnvtol,f,,1e-3
        !
        dk,1,all,0.0d0,,1
        dk,2, uy,0.0d0,,1
        dk,2, uz,0.0d0,,1
        !
        time,20.0
        dk,2,ux,0.06
        KBC,0
        NSUBST,50,0,0,0
        OUTRES,ALL,ALL,
        lswrite,1
        !
        time,60.0
        dk,2,ux,-0.06
        KBC,0
        NSUBST,100,0,0,0
        OUTRES,ALL,ALL,
        lswrite,2
        !
```

```
        time,80.0
        dk,2,ux,0.00
        KBC,0
        NSUBST,50,0,0,0
        OUTRES,ALL,ALL,
        lswrite,3
        !
        FINISH
!
! solve
!
        /SOLU
        outeq,on
        /STATUS,SOLU
        LSSOLVE,1,3,1
        FINISH
```

APPENDIX D: ANSYS solution process, force control example

Shape memory alloys exhibit a strongly nonlinear behavior that, in the perfect single crystal, quasi-static (isothermal) case, can be approximated by a multi-linear stress-strain curve. In order to elucidate the solution process used by ANSYS with such materials, the simplest related example is chosen, a bi-linear material with a three node, two element system.

Consider the model system, shown below:

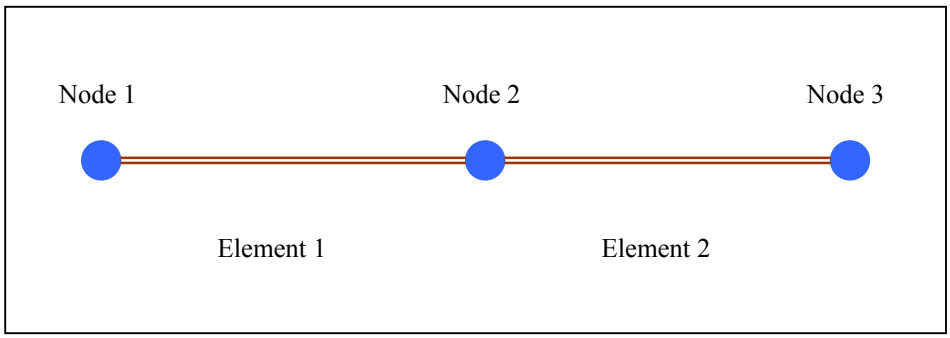

Figure 45: Model system, 2 links, 3 nodes, for example solution process

Let the material behavior be defined as in Figure 46:

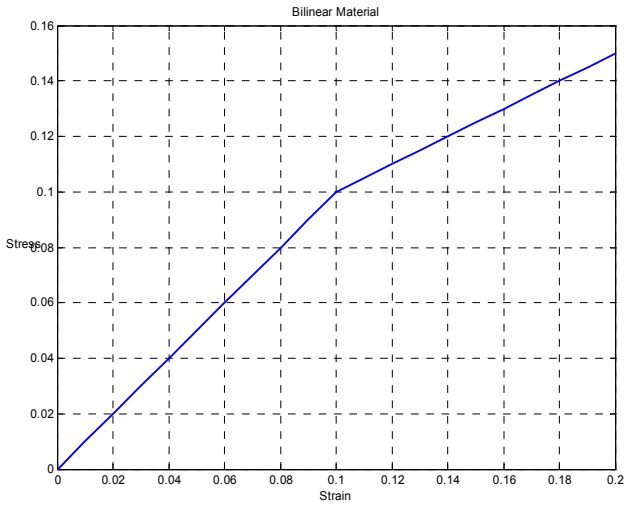

Figure 46: Bi-linear material behavior used in example solution process

Let a tensile force be prescribed at node 3. From

$$\mathbf{K}_{n+1}^{(i)}\left(\delta\mathbf{a}_{n+1}^{(i+1)}\right)=\mathbf{P}_{n+1}^{ext}-\mathbf{I}_{n+1}^{(i)} \tag{A4.1}$$

the expanded matrix equation thus becomes:

$$\begin{aligned}
k_{1,n+1}^{(i)}\left(\delta a_{1,n+1}^{(i+1)}\right) - k_{1,n+1}^{(i)}\left(\delta a_{2,n+1}^{(i+1)}\right) &= P_{1,n+1}^{ext}-I_{1,n+1}^{(i)}\\
k_{1,n+1}^{(i)}\left(\delta a_{1,n+1}^{(i+1)}\right)+\left(k_{1,n+1}^{(i)}+k_{2,n+1}^{(i)}\right)\left(\delta a_{2,n+1}^{(i+1)}\right)-k_{2,n+1}^{(i)}\left(\delta a_{3,n+1}^{(i+1)}\right) &= P_{2,n+1}^{ext}-I_{2,n+1}^{(i)}\\
-k_{2,n+1}^{(i)}\left(\delta a_{2,n+1}^{(i+1)}\right)+k_{2,n+1}^{(i)}\left(\delta a_{3,n+1}^{(i+1)}\right) &= P_{3,n+1}^{ext}-I_{3,n+1}^{(i)}
\end{aligned} \tag{A4.2}$$

For the model problem, with $L_0 = A = 5$, and an initial load step of $P^{ext}_{3,1} = 0.375$ (3/8), the iteration scheme works as follows:

Force convergence criterion:

The tangent stiffness and element stresses are calculated from an assumed initial displacement increment of zero; hence element stress and restoring force are zero, as are nodal forces $\mathbf{I}^{(0)}_{all,1}$. Then, for the first iteration: (note: $k_{all,1}^{(0)} = 1$ and $\delta a^{(all)}_{1,1}=0$)

$$\begin{aligned}
(2)\delta a_{2,1}^{(1)}-\delta a_{3,1}^{(1)} &= 0\\
-\delta a_{2,1}^{(1)}+\delta a_{3,1}^{(1)} &= P_{3,1}^{ext}=3/8
\end{aligned} \tag{A4.3}$$

From the above,

$$\delta a_{2,1}^{(1)}=\frac{\delta a_{3,1}^{(1)}}{(2)}$$

and, consequently,

$$-\frac{\delta a_{3,1}^{(1)}}{2}+\delta a_{3,1}^{(1)}=\frac{\delta a_{3,1}^{(1)}}{2}=P_{3,n+1}^{ext}$$

$$\begin{aligned}
\delta a_{3,1}^{(1)} &= 2P_{3,1}^{ext}=3/4=0.75\\
\delta a_{2,1}^{(1)} &= 3/8=0.375
\end{aligned} \tag{A4.4}$$

Checking for convergence:

$$\begin{aligned}
\varepsilon_{all,1}^{(1)} &= 3/40 = 0.075 \\
\sigma_{all,1}^{(1)} &= 3/40 = 0.075 \\
k_{all,1}^{(1)} &= 1 \\
F_1^{e1,e2(1)} &= 3/8 = 0.375 \\
I_{1,1}^{(1)} &= -3/8 = -0.375 \\
I_{2,1}^{(1)} &= 0 \\
I_{3,1}^{(1)} &= 3/8 = 0.375
\end{aligned} \tag{A4.5}$$

$$\mathbf{P}_1^{ext} = \begin{Bmatrix} 0 \\ 0 \\ 0.375 \end{Bmatrix}$$

where F^{en} is force corresponding to element stresses.

Then,

$$\mathbf{P}_1^{ext} - \mathbf{I}_1^{(1)} = \begin{Bmatrix} 0.375 \\ 0 \\ 0 \end{Bmatrix}$$

The reference force used in setting the convergence criterion is found from the magnitude of the applied force vector and the components of the residual force vector corresponding to constrained nodes. In this case, the reference force is:

$$R_{ref}^{(1)} = \sqrt{0.375^2 + 0.375^2} = 0.5303$$

With a convergence tolerance of 10^{-6}, the convergence criterion is 0.5303×10^{-6}.

The force convergence value is calculated from the magnitude of the components of the residual force vector corresponding to degrees of freedom (nodes 2 and 3), in this case zero. Since the force convergence value is less than the criterion, solution is converged after only one iteration, which is to be expected in a linear regime.

Consider a second load step, with a force application of 0.625 (5/8) on node 3. Note that $\mathbf{a}_0$ comes from the converged solution of the previous load step.

$$\mathbf{I}_2^{(0)} = \begin{pmatrix} -0.375 \\ 0 \\ 0.375 \end{pmatrix}$$

$$\mathbf{P}_2^{ext} = \begin{pmatrix} 0 \\ 0 \\ 0.625 \end{pmatrix}$$

The first iteration is:

$$\begin{aligned} \left(k_{1,2}^{(0)} + k_{2,2}^{(0)}\right)\left(\delta a_{2,2}^{(1)}\right) - k_{2,2}^{(0)}\left(\delta a_{3,2}^{(1)}\right) &= P_{2,2}^{ext} - I_{2,2}^{(0)} \\ -k_{2,2}^{(0)}\left(\delta a_{2,2}^{(1)}\right) + k_{2,2}^{(0)}\left(\delta a_{3,2}^{(1)}\right) &= P_{3,2}^{ext} - I_{3,2}^{(0)} \end{aligned} \tag{A4.6}$$

So, then

$$(2)\delta a_{2,2}^{(1)} - \delta a_{3,2}^{(1)} = 0 - 0$$
$$-\delta a_{2,2}^{(1)} + \delta a_{3,2}^{(1)} = 1/4$$

and

$$\delta a_{2,2}^{(1)} = \frac{\delta a_3^{(1)}}{2}$$

$$\begin{aligned} \delta a_{3,2}^{(1)} &= 1/2 \\ \delta a_{2,2}^{(1)} &= 1/4 \end{aligned} \tag{A4.7}$$

Corresponding nodal displacements are (initial values correspond to the previous load step converged solution):

$$a_{2,2}^{(1)} = a_{2,2}^{(0)} + \delta a_{2,2}^{(1)} = 3/8 + 1/4 = 5/8$$
$$a_{3,2}^{(1)} = a_{3,2}^{(0)} + \delta a_{3,2}^{(1)} = 3/4 + 1/2 = 10/8$$

Checking for convergence:

$$\varepsilon_{all,2}^{(1)} = 0.125 = 5/40$$
$$\sigma_{all,2}^{(1)} = 0.1125 = 9/80$$
$$k_{all,2}^{(1)} = 1/2$$
$$F_2^{e(1)} = 0.5625 = 9/16$$
$$\mathbf{I}_2^{(1)} = \begin{pmatrix} -0.5625 \\ 0 \\ 0.5625 \end{pmatrix}$$

Then,

$$\mathbf{P}_2^{ext} - \mathbf{I}_2^{(1)} = \begin{Bmatrix} 0.5625 \\ 0 \\ 0.0625 \end{Bmatrix}$$

The reference force is:

$$R_{ref}^{(1)} = \sqrt{0.625^2 + 0.5625^2} = 0.8409$$

The convergence criterion is 0.8409 x 10^{-6}.

The force convergence value here is 0.0625, which is greater than the criterion. Hence, the solution is not converged and another iteration is needed.

For the next iteration:

$$\begin{aligned} \delta a_{2,2}^{(2)} - \delta a_{3,2}^{(2)} / 2 &= 0 \\ -\delta a_{2,2}^{(2)} / 2 + \delta a_{3,2}^{(2)} / 2 &= 5/8 - 9/16 \end{aligned} \tag{A4.8}$$

$$\delta a_{2,2}^{(2)} = \delta a_{3,2}^{(2)} / 2$$

$$\begin{aligned} \delta a_{3,2}^{(2)} &= 1/4 \\ a_{2,2}^{(2)} &= 1/8 \end{aligned} \tag{A4.9}$$

Corresponding nodal displacements are:

$$a_{2,2}^{(2)} = a_{2,2}^{(1)} + \delta a_{2,2}^{(2)} = 5/8 + 1/8 = 6/8$$
$$a_{3,2}^{(2)} = a_{3,2}^{(1)} + \delta a_{3,2}^{(2)} = 10/8 + 1/4 = 12/8$$

Now check again for convergence:

$$\varepsilon_{all,2}^{(2)} = 0.15 = 3/20$$
$$\sigma_{all,2}^{(2)} = 0.125 = 5/40$$
$$F_2^{e(2)} = 0.625 = 5/8$$
$$\mathbf{I}_2^{(2)} = \begin{pmatrix} -0.625 \\ 0 \\ 0.625 \end{pmatrix}$$

Then,

$$\mathbf{P}_2^{ext} - \mathbf{I}_2^{(2)} = \begin{Bmatrix} 0.625 \\ 0 \\ 0 \end{Bmatrix}$$

The reference force is:

$$R_{ref}^{(1)} = \sqrt{0.625^2 + 0.625^2} = 0.8839$$

The convergence criterion is 0.8839 x 10^{-6}.

The force convergence value here is 0.0, which is less than the criterion. Hence, the solution is converged after two iterations. Because of the change in the tangent stiffness, some iterations should be expected for solution convergence.

To sum up, in the case of a bi-linear material under the force convergence criterion, the solution in a force-driven simulation converges in one iteration in a linear regime and in two iterations over a slope discontinuity in the stress-strain curve.

Displacement convergence criterion:

Under displacement convergence criterion, the process is identical up to equation A4.4. From this displacement, a convergence criterion is determined based on the tightest of the infinitive, L1, and L2 norms of the displacement vector. In this case,

$$\|\mathbf{a}\|_{\infty} = \max(|a_i|) = 3/4$$
$$\|\mathbf{a}\|_{L1} = 3/4 + 3/8 = 9/8$$
$$\|\mathbf{a}\|_{L2} = \sqrt{(3/4)^2 + (3/8)^2} = 0.8385$$

If the convergence tolerance is 10^{-6}, then the chosen convergence criterion is 0.75 x 10^{-6}. The displacement convergence value is determined from the L2 norm of the displacement increment, which for the first iteration of the first sub-step is the same as the L2 norm of the displacement, here 0.8385. Since this value is greater than the convergence criterion, a second iteration is required.

However, the second iteration, using the updated values shown in equation A4.5, produces no incremental change in the displacements, so that the L2 norm of the second iteration increment is zero:

$$\begin{aligned}(2)\delta a_{2,1}^{(2)} - \delta a_{3,1}^{(2)} &= 0 \\ -\delta a_{2,1}^{(2)} + \delta a_{3,1}^{(2)} &= P_{3,1}^{ext} - I_{3,1}^{(1)} = 0\end{aligned} \tag{A5.8}$$

This leads to a converged solution after the second iteration.

This process repeats in the second load step, so that convergence is achieved after three iterations. According to this case, then, displacement convergence requires one extra iteration per load step over the force convergence criterion.

APPENDIX E: ANSYS solution process, displacement control example

Shape memory alloys exhibit a strongly nonlinear behavior that, in the perfect single crystal, quasi-static (isothermal) case, can be approximated by a multi-linear stress-strain curve. In order to elucidate the solution process used by ANSYS with such materials, the simplest related example is chosen, a bi-linear material with a three node, two element system.

Consider the model problem, shown below:

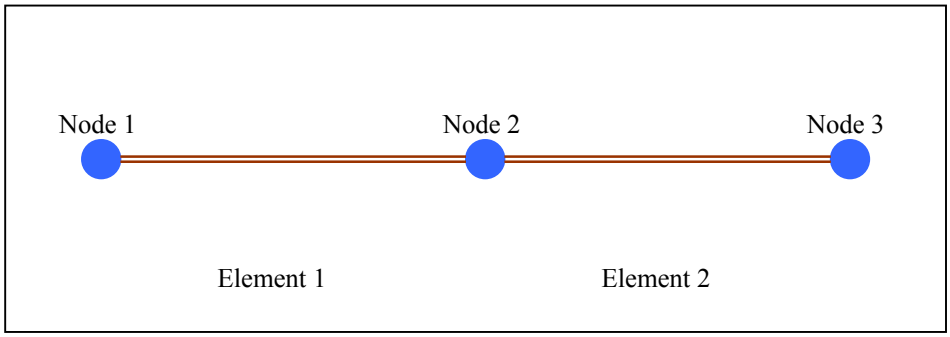

Figure 47: Model system, 2 links, 3 nodes, for example solution process

Let the material behavior be defined as in Figure 48:

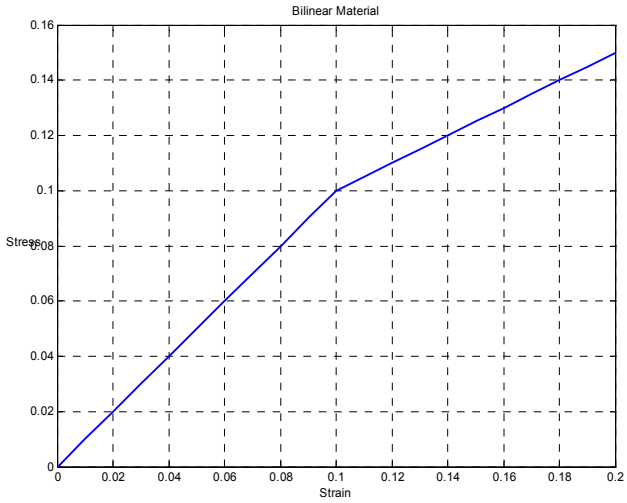

Figure 48: Bi-linear material behavior used in example solution process

Let a positive *displacement* be prescribed at node 3. Then the conditions

$$a_{3,1}^{(0)} = 0$$
$$a_{3,n+1}^{(0)} = a_{3,n}^{(converged)}$$
$$a_{3,n+1}^{(i\neq 0)} = a_{3,n+1}^{*}$$

hold, so that

$\delta \mathbf{a}^{(1)}_{3,n+1} \neq 0$, but

$\delta \mathbf{a}^{(i\neq 0)}_{3,n+1} = 0$

From

$$\mathbf{K}_{n+1}^{(i)}\left(\delta \mathbf{a}_{n+1}^{(i+1)}\right) = \mathbf{P}_{n+1}^{ext} - \mathbf{I}_{n+1}^{(i)} \tag{A5.1}$$

The expanded matrix equation is:

$$\begin{aligned} k_{1,n+1}^{(i)}\left(\delta a_{1,n+1}^{(i+1)}\right) - k_{1,n+1}^{(i)}\left(\delta a_{2,n+1}^{(i+1)}\right) &= P_{1,n+1}^{ext} - I_{1,n+1}^{(i)} \\ k_{1,n+1}^{(i)}\left(\delta a_{1,n+1}^{(i+1)}\right) + \left(k_{1,n+1}^{(i)} + k_{2,n+1}^{(i)}\right)\left(\delta a_{2,n+1}^{(i+1)}\right) - k_{2,n+1}^{(i)}\left(\delta a_{3,n+1}^{(i+1)}\right) &= P_{2,n+1}^{ext} - I_{2,n+1}^{(i)} \\ - k_{2,n+1}^{(i)}\left(\delta a_{2,n+1}^{(i+1)}\right) + k_{2,n+1}^{(i)}\left(\delta a_{3,n+1}^{(i+1)}\right) &= P_{3,n+1}^{ext} - I_{3,n+1}^{(i)} \end{aligned} \tag{A5.2}$$

To reiterate, for iterations other than the first, both the first and third equations, where displacement is prescribed, should not be used for solution. In the first iteration,

$\delta a^{(1)}_{3,1}$ = the prescribed displacement increment.

For the model problem, with $L_0 = A = 5$, and an initial load step of $a_{3,1} = 0.75$ (3/4), the iteration scheme works as follows:

Force convergence criterion:

The tangent stiffness and element stresses are calculated from an assumed initial displacement increment of $\{\delta \mathbf{a}^{(0)}_{all,1}\} = 0$ (hence element stress and restoring force are zero, as are nodal forces $\mathbf{I}^{(0)}_{all,1}$). Then, for the first iteration (note: $k_{all,1}^{(0)} = 1$ and $\delta a^{(all)}_{1,1}=0$), all known

quantities in the system of equations are moved to the right-hand-side of the equation for solving for the incremental displacement of node 2:

$$\left(k_{1,1}^{(0)} + k_{2,1}^{(0)}\right)\left(\delta a_{2,1}^{(1)}\right) = k_{2,1}^{(i)}\left(\delta a_{3,1}^{(1)}\right) + P_{2,1}^{ext} - I_{2,1}^{(0)} \tag{A5.3}$$

Then, for the first iteration:

$$(2)\delta a_{2,1}^{(1)} = 0.75$$

or,

$$\delta a_{2,1}^{(1)} = 0.375 = 3/8 \tag{A5.4}$$

Checking for convergence:

$$\begin{aligned}
&\varepsilon_{all,1}^{(1)} = 3/40 = 0.075 \\
&\sigma_{all,1}^{(1)} = 3/40 = 0.075 \\
&k_{all,1}^{(1)} = 1 \\
&F_1^{e1,e2(1)} = 3/8 = 0.375 \\
&I_{1,1}^{(1)} = -3/8 = -0.375 \\
&I_{2,1}^{(1)} = 0 \\
&I_{3,1}^{(1)} = 3/8 = 0.375
\end{aligned} \tag{A5.5}$$

The residual force vector is:

$$\mathbf{R} = \begin{Bmatrix} P_{1,n+1}^{ext} - I_{1,1}^{(1)} \\ P_{2,1}^{ext} - I_{2,1}^{(1)} \\ P_{3,1}^{ext} - I_{3,1}^{(1)} \end{Bmatrix} = \begin{Bmatrix} 0.375 \\ 0 \\ -0.375 \end{Bmatrix}$$

The reference force used in setting the convergence criterion is found from the magnitude of the applied force vector, which is zero, and the components of the residual force vector corresponding to constrained nodes. In this case, the reference force is:

$$R_{ref}^{(1)} = \sqrt{0.375^2 + 0.375^2} = 0.5303$$

With a convergence tolerance of 10^{-6}, the convergence criterion is 0.5303 x 10^{-6}.

The force convergence value, calculated from the magnitude of the components of the residual force vector corresponding to degrees of freedom (node 2 only, in this case), is zero. Since the convergence value is less than the criterion, the solution is converged after only one iteration, which is to be expected in a linear regime.

Consider a second load step, with a displacement prescription of 1.5 (3/2) on node 3.

$$a_{2,2}^{(0)} = 3/8 = 0.375$$
$$a_{3,2}^{(0)} = 3/4 = 0.75$$
$$\varepsilon_{all,2}^{(0)} = 3/40 = 0.075$$
$$k_{all,2}^{(0)} = 1$$
$$\sigma_{all,2}^{0} = 3/40 = 0.075$$
$$F_2^{e1,e2(0)} = 3/20 = 0.375$$
$$\mathbf{I}_2^{(0)} = \begin{pmatrix} -0.375 \\ 0 \\ 0.375 \end{pmatrix}$$

And

$$\left(k_{1,2}^{(0)} + k_{2,2}^{(0)}\right)\left(\delta a_{2,2}^{(1)}\right) = k_{2,n+1}^{(i)}\left(\delta a_{3,2}^{(1)}\right) + P_{2,2}^{ext} - I_{2,2}^{(0)} \qquad \text{(A5.6)}$$

So, the first iteration is:

$$2\delta a_{2,2}^{(1)} = 0.75$$

or,

$$\delta a_{2,2}^{(1)} = 0.375 \qquad \text{(A5.7)}$$

The corresponding nodal displacement is:

$$a_{2,2}^{(1)} = a_{2,2}^{(0)} + \delta a_{2,2}^{(1)} = 0.375 + 0.375 = 0.75$$

Checking for convergence:

$$\varepsilon_{all,2}^{(1)} = 0.15 = 3/20$$
$$\sigma_{all,2}^{(1)} = 0.125 = 5/40$$
$$F_2^{e1,e2(1)} = 0.625 = 5/8$$
$$I_{1,2}^{(1)} = -0.625$$
$$I_{2,2}^{(1)} = 0$$
$$I_{3,2}^{(1)} = 0.625$$

The residual force vector is

$$R = \begin{Bmatrix} P_{1,n+1}^{ext} - I_{1,1}^{(1)} \\ P_{2,1}^{ext} - I_{2,1}^{(1)} \\ P_{3,1}^{ext} - I_{3,1}^{(1)} \end{Bmatrix} = \begin{Bmatrix} 0.625 \\ 0 \\ -0.625 \end{Bmatrix}$$

In this case, the reference force is:

$$R_{ref}^{(1)} = \sqrt{0.625^2 + 0.625^2} = 0.8839$$

With a convergence tolerance of 10^{-6}, the convergence criterion is 0.8839 x 10^{-6}.

The force convergence value is again zero. Since the convergence value is less than the criterion, the solution is converged after only one iteration. This unexpected outcome results from the fact that the ratio of stiffnesses between the elements remains constant, even though the actual stiffness value changes.

Displacement convergence criterion:

Under displacement convergence criterion, the process is identical up to equation (A5.5). From this displacement, a convergence criterion is determined based on the tightest of the infinitive, L1, and L2 norms of the displacement vector. In this case,

$$\|\mathbf{a}\|_{\infty} = \max(|a_i|) = 3/4$$
$$\|\mathbf{a}\|_{L1} = 3/4 + 3/8 = 9/8$$
$$\|\mathbf{a}\|_{L2} = \sqrt{(3/4)^2 + (3/8)^2} = 0.8385$$

If the convergence tolerance is 10^{-6}, then the chosen convergence criterion is 0.75×10^{-6}. The displacement convergence value is determined from the L2 norm of the displacement increment, which for the first iteration of the first sub-step is the same as the L2 norm of the displacement, here 0.8385. Since this value is greater than the convergence criterion, a second iteration is required.

However, the second iteration, using the updated values shown in equation A5.5, produces no incremental change in the displacements, so that the L2 norm of the second iteration increment is zero:

$$\left(k_{1,1}^{(1)} + k_{2,1}^{(1)}\right)\left(\delta a_{2,1}^{(2)}\right) = P_{2,1}^{ext} - I_{2,1}^{(1)} = 0 \qquad \text{(A5.8)}$$

This leads to a converged solution after the second iteration.

This process repeats in the second load step, so that convergence is achieved after two iterations. According to this case, then, displacement convergence requires one extra iteration per load step over the force convergence criterion.