

Titre: Analyse thermomécanique d'actuateurs en alliage à mémoire de
forme (AMF)
Title:

Auteur: Alain Leboeuf
Author:

Date: 1997

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Leboeuf, A. (1997). Analyse thermomécanique d'actuateurs en alliage à mémoire
de forme (AMF) [Mémoire de maîtrise, École Polytechnique de Montréal].
Citation: PolyPublie. <https://publications.polymtl.ca/6675/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/6675/>
PolyPublie URL:

**Directeurs de
recherche:** François Trochu
Advisors:

Programme: Non spécifié
Program:

UNIVERSITÉ DE MONTRÉAL

ANALYSE THERMOMÉCANIQUE D'ACTUATEURS
EN ALLIAGE À MÉMOIRE DE FORME (AMF)

ALAIN LEBOEUF
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE EN INGÉNIERIE
(GÉNIE MÉCANIQUE)
AVRIL 1997

© Alain Leboeuf, 1997.



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-33147-4

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

ANALYSE THERMOMÉCANIQUE D'ACTUATEURS
EN ALLIAGE À MÉMOIRE DE FORME (AMF)

présenté par : LEBOEUF Alain

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

M. TURENNE Sylvain, Ph.D., président

M. TROCHU François, Ph.D., membre et directeur de recherche

M. BRAILOVSKI Vladimir, Ph.D., membre

REMERCIEMENTS

J'aimerais remercier Dr François Trochu pour sa confiance, son soutien, ses conseils judicieux ainsi que son infatigable motivation à toujours donner le meilleur de soi.

J'aimerais également remercier :

- Le CRSNG pour son soutien financier;
- GEC Alstom T&D pour son soutien financier et son support matériel;
- IREQ pour l'accès à leur laboratoire et leur équipement de caractérisation des AMF;
- SMA Inc. pour les échantillons de TiNi;

- MM. Daniel Demissy et Hadi Alidou pour leur soutien et leur encadrement tout au long de mes stages chez GEC Alstom T&D;
- Dr Vladimir Brailovski pour son soutien, ses critiques constructives et son enthousiasme constant;
- Patrick Terriault et Marc-Antoine Meunier pour leur collaboration;

- Mes parents et amis qui m'ont encouragé pendant la durée de mes études, particulièrement mon épouse Martine sans qui je ne serais pas rendu où je suis maintenant.

RÉSUMÉ

Les propriétés de superélasticité et d'effet mémoire des AMF font de ces derniers d'excellents candidats pour la conception d'actuateurs électriques. Bien qu'il existe déjà plusieurs modèles de comportement thermomécanique, la modélisation des temps de réaction des AMF est un domaine fort peu développé. Ce mémoire présente un modèle bilinéaire du comportement thermomécanique des AMF jumelé à un modèle analytique des échanges thermiques pour un actuateur AMF à fil droit chauffé par effet Joule. Le modèle mécanique est une version linéarisée du modèle non linéaire KRIGAMF qui utilise l'interpolation par krigeage. Le modèle thermique calcule les temps d'activation d'un actuateur en utilisant les équations du transfert de chaleur et en supposant une température uniforme de l'échantillon ("lumped capacitance method"). Un coefficient de correction permet de réduire les erreurs imputables aux pertes par conduction sans avoir recours aux méthodes itératives de calcul. Une méthode d'interpolation de l'énergie latente de transformation est utilisée pour modéliser le temps requis pour compléter la transformation de phase. Cette approche consiste à exprimer la chaleur spécifique en fonction de la température par un polynôme du deuxième degré dans l'expression de l'énergie latente de transformation. Ce polynôme représente l'évolution de la chaleur spécifique en fonction de la température lors de la transformation de phase. On obtient donc une expression de la température en fonction du temps.

Le modèle peut calculer les temps de réaction dans le cas d'une contrainte constante, ainsi que dans le cas d'une contrainte variable (contre-ressort et/ou butée). Cette approche s'avère être simple et rapide. De plus, elle permet de prévoir les temps de réaction avec une bonne précision dans le cas où l'AMF est soumis à une contrainte constante. Il reste maintenant à valider le modèle pour les différents cas de contrainte variable. Cependant,

cette approche comporte certaines limites : la sous-estimation de la déformation de transformation, le calcul du temps des échanges thermiques lors de sous-cycles à contraintes variables n'est pas encore implanté et le temps des échanges thermiques du refroidissement est sous-estimé. Cette dernière limite est causée principalement par le fait que le coefficient de correction est considéré constant.

ABSTRACT

Because of their superelastic and shape memory properties, SMA are excellent materials for the design of electrical actuators. Although there already exist several models of their thermomechanical behavior, the prediction of reaction times for SMA actuation is a poorly developed area. This Masters thesis presents a bilinear model of the thermomechanical behavior of SMA, paired with an analytical model for thermal exchanges in the case of a straight wire actuator heated by the Joule effect. The mechanical model is a linearized version of the nonlinear model KRIGAMF which uses kriging. The thermal model computes the time of activation for an actuator by using heat transfer equations and supposing a uniform temperature in the sample (lumped capacitance method). A correction factor is used to take into account the energy dissipation at the extremities (conduction losses) without using iterative calculation methods. A method of interpolation of the latent heat of transformation is used for computing the time required to complete the phase transformation. This approach consists in expressing the specific heat with a polynomial of the second order in the expression of the latent heat of transformation. This polynomial represents the evolution of the specific heat in relation to temperature during the phase transformation. We therefore obtain an expression of the temperature in function of time.

The model can compute the actuation times in the case of a constant stress as well as in the case of a variable stress (bias spring and/or trust). This approach proves to be a simple and quick way to predict the reaction times of SMA with a good precision in the case of constant stress. The model must now be validated for the different cases of variable stress. This approach however presents some limits : underestimation of the transformation deformation, the computation of heat exchange times for subcycles in the

case of a variable stress is not yet implemented and the heat exchange time during cooling is underestimated. This last limit is mainly caused by the fact that the correction factor is considered constant.

TABLE DES MATIÈRES

REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	ix
LISTE DES ANNEXES	xii
LISTE DES TABLEAUX	xv
LISTE DES FIGURES	xvi
LISTE DES SIGLES ET ABRÉVIATIONS	xix
INTRODUCTION	1
1 - PRÉSENTATION DES AMF ET DES MODÈLES DE LEUR COMPORTEMENT THERMOMÉCANIQUE	3
1.1 Introduction aux AMF	3
1.1.1 Super-thermoélasticité	5
1.1.2 Comportement caoutchoutique et effet mémoire	6

1.1.3 Effet amortissant	8
1.2 Revue bibliographique sur la modélisation du comportement thermomécanique des AMF	8
1.2.1 Modèles micromécaniques	9
1.2.2 Modèles phénoménologiques	3
2 - PRÉSENTATION DU MODÈLE	19
2.1 Modèle d'analyse mécanique	20
2.1.1 Modèle à contrainte constante	21
2.1.2 Modèle à contrainte variable	24
2.1.3 Modèle avec une butée	25
2.2 Modèle d'analyse thermique	25
2.2.1 Température à atteindre	29
2.2.2 Spécification de la durée des échanges thermiques	30
2.3 Couplage des modèles mécanique et thermique	31
3 - ANALYSE QUALITATIVE DES RÉSULTATS	33
3.1 Analyse du calcul mécanique	34
3.2 Influence de l'incrément de température	38
4 - ANALYSE QUANTITATIVE DES RÉSULTATS	46
4.1 Comparaison des résultats calculés avec ceux de la littérature	46
4.1.1 Modèle mécanique	46
4.1.2 Modèle thermique	48
4.2 Comparaison des résultats du modèle couplé avec l'expérience	50

4.2.1 Caractérisation des échantillons	51
4.2.2 Essai de validation	56
4.2.3 Analyse des résultats	60
CONCLUSION	67
RÉFÉRENCES	68
ANNEXES	74

LISTE DES ANNEXES

ANNEXE I : Essai de validation pour l'échantillon cuivreux	74
ANNEXE II : Essai de validation pour l'échantillon de TiNi non traité	79
ANNEXE III : Courbe caractéristique de stabilisation de l'échantillon de TiNi	89
ANNEXE IV : Formats des fichiers	90
ANNEXE V : Documentation des bibliothèques de modélisation du comportement thermomécanique des AMF	95
1. PRÉSENTATION DES BIBLIOTHÈQUES	95
2. TYPAMF_L.LIB : BIBLIOTHÈQUE DE GESTION DES TYPES	97
2.1 Types pour le calcul de la loi de comportement	97
2.1.1 Type AmfData	97
2.1.2 Type Plan	98
2.1.3 Type TypeDePlan	98
2.1.4 Type Noeud	99
2.2 Types pour la gestion des points et des courbes	100
2.2.1 Type Point	100
2.2.2 Type Courbe	100
2.2.3 Type PointChn	101
2.2.4 Type CourbeChn	101
2.2.5 Le type SeqTP	101

3. MATAMF_L.LIB : LIBRAIRIE DE CALCUL DE LA LOI DE COMPORTEMENT	102
4. CALAMF_L.LIB : LIBRAIRIE DE CALCUL DE CHARGEMENT SIMPLE	103
4.1 Fonctions de calcul de la traction	104
4.2 Fonctions de calcul de la torsion	104
4.2.1 Calcul inverse	104
4.2.2 Calcul direct	105
4.3 Fonctions d'entrée/sortie des données	105
4.3.1 Gestion d'un contre-ressort	106
4.3.2 Affichage graphique	106
5. THRAMF_L.LIB : LIBRAIRIE DE CALCUL D'ÉCHANGE THERMIQUE	107
5.1 Types des variables	107
5.1.1 Type Coefficient	107
5.1.2 Type CtePhysique	107
5.1.3 Type GeomFil	108
5.1.4 Type Parametre	108
5.2 Fonctions	108
5.2.1 Entrée/sortie des données	108
5.2.2 Calcul des échanges thermiques	109
6. RESAMF_L.LIB : LIBRAIRIE DE CALCUL DE RESSORTS AMF HÉLICOÏDAUX	109
6.1 Type des variables	109

6.2 Fonctions	110
6.2.1 Entrée/sortie des données	110
6.2.2 Calcul de ressort	110
7. DOCUMENTATION DÉTAILLÉE DES TYPES ET FONCTIONS DES LIBRAIRIES	111
7.1 Gestion des types (typamf_1.lib)	111
7.1.1 Types	111
7.1.2 Fonctions	116
7.2 Loi de comportement (matamf_1.lib)	124
7.2.1 Fonctions	124
7.3 Chargement simple (calamf_1.lib)	127
7.3.1 Fonctions	127
7.4 Échanges thermiques (thramf_1.lib)	134
7.4.1 Types	134
7.4.2 Fonctions	136
7.5 Ressorts AMF hélicoïdaux	141
7.5.1 Types	141
7.5.2 Fonctions	142
RÉFÉRENCES	147

LISTE DES TABLEAUX

Tableau 2.1	Variables utilisées pour la modélisation d'un actuateur à fil droit.	32
Tableau 3.1	Données pour la validation qualitative.	33
Tableau 4.1	Données pour la validation partielle - modèle mécanique (contrainte = 174 MPa).	47
Tableau 4.2	Données pour la validation partielle - modèle mécanique (contrainte = 128 MPa).	48
Tableau 4.3	Données pour la validation partielle - modèle thermique (courant = 75 A).	49
Tableau 4.4	Données pour la validation partielle - modèle thermique (courant = 50 A).	50
Tableau 4.5	Données pour la validation du modèle.	55
Tableau 4.6	Paramètres de la chambre de fatigue de l'IREQ.	56

LISTE DES FIGURES

Figure 1.1	Évolution des températures de transformation en fonction de la contrainte.....	5
Figure 1.2	Comportement superélastique des AMF.....	6
Figure 1.3	Comportement caoutchoutique des AMF.....	7
Figure 1.4	Effet mémoire simple sens.....	7
Figure 1.5	Modèles bilinéaires.....	16
Figure 2.1	Cycle caractéristique d'utilisation d'un actuateur AMF.....	20
Figure 2.2	Espace d'interpolation du modèle KRIGAMF.....	21
Figure 2.3	Espace d'interpolation du modèle bilinéaire.....	22
Figure 2.4	Courbe de traction superélastique idéalisée.....	23
Figure 2.5	Diagramme d'état simplifié.....	23
Figure 2.6	Méthode de calcul du contre-ressort.....	25
Figure 2.7	Bilan thermique d'un fil échauffé par courant électrique.....	26
Figure 3.1	Déformation calculée en fonction de la température pour une contrainte imposée constante.....	34
Figure 3.2	Déformation calculée en fonction de la température pour une contrainte imposée variable (contre-ressort).....	35
Figure 3.3	Contrainte calculée en fonction de la température pour une contrainte imposée constante.....	36
Figure 3.4	Contrainte calculée en fonction de la température pour une contrainte imposée variable (contre-ressort).....	36
Figure 3.5	Contrainte calculée en fonction de la déformation.....	37
Figure 3.6	Courbes caractéristiques des contre-ressorts.....	38
Figure 3.7	Influence de l'incrément de température sur la déformation	

	(contrainte imposée constante).	39
Figure 3.8	Influence de l'incrément de température sur la déformation (contrainte imposée constante avec butée).	39
Figure 3.9	Influence de l'incrément de température sur la déformation (contrainte imposée variable, raideur du contre-ressort $k = 872\text{N/m}$).	41
Figure 3.10	Influence de l'incrément de température sur la déformation (contrainte imposée variable avec butée, raideur du contre-ressort $k = 872\text{N/m}$).	41
Figure 3.11	Influence de l'incrément de température sur la déformation (contrainte imposée variable, raideur du contre- ressort $k = 262\text{N/m}$).	42
Figure 3.12	Influence de l'incrément de température sur la déformation (contrainte imposée variable avec butée, raideur du contre-ressort $k = 262\text{N/m}$).	42
Figure 3.13	Influence de l'incrément de température sur le temps (contrainte imposée variable - cycle de chauffage).	43
Figure 3.14	Influence de l'incrément de température sur le temps (contrainte imposée variable - cycle de refroidissement).	44
Figure 4.1	Comparaison de courbes calculées de la déformation en fonction de la température avec des résultats tirés de la littérature	47
Figure 4.2	Comparaison de courbes calculées de la température en fonction du temps avec des résultats expérimentaux	49
Figure 4.3	Courbe superélastique de l'échantillon de TiNi traité.	53
Figure 4.4	Courbes de traction à différentes températures du TiNi traité.	54
Figure 4.5	Schéma de la chambre de fatigue de l'IREQ.	57
Figure 4.6	Photographie de la chambre de fatigue de l'IREQ.	58

Figure 4.7 Photographies de la fixation de l'échantillon et du support de la charge dans la chambre de fatigue de l'IREQ.	59
Figure 4.8 Interface de sortie du système d'acquisition de la chambre de fatigue de l'IREQ.	60
Figure 4.9 Déformation en fonction du temps (contrainte constante de 560,4 MPa).	61
Figure 4.10 Déformation expérimentale en fonction du temps pour différents niveaux de contrainte.	63
Figure 4.11 Déformation calculée en fonction du temps pour différents niveaux de contrainte.	63
Figure 4.12 Déformation en fonction du temps (contrainte constante de 620,7 MPa).	64
Figure 4.13 Température et déformation en fonction du temps.	65
Figure 4.14 Déformation en fonction de la température pour la validation du modèle.	66

LISTE DES SIGLES ET ABRÉVIATIONS

Symbole	Description	Unités
A	1 ^{er} coefficient de l'équation qui modélise la courbe $C_p(T)$	-
A_e	Aire d'échange thermique	m^2
A_f	Température finale de la transformation austénitique	$^{\circ}C$
AMF	Alliage à mémoire de forme	-
A_s	Température de début de la transformation austénitique	$^{\circ}C$
B	2 ^{ème} coefficient de l'équation qui modélise la courbe $C_p(T)$	-
C_p	Chaleur spécifique	$J / kg^{\circ}C$
d	Diamètre de l'échantillon	m
D	3 ^{ème} coefficient de l'équation qui modélise la courbe $C_p(T)$	-
$d\sigma/d\varepsilon$	Pente contrainte-déformation	MPa / m/m
$d\sigma/dT$	Pente contrainte-température	MPa / $^{\circ}C$
E_1	Module élastique d'austénite	MPa
E_2	Module élastique de martensite	MPa
h	Coefficient de convection	$W / m^2^{\circ}C$
i	Courant électrique	A
k	Constante de rigidité d'un ressort	N / m
l	Longueur de l'échantillon	m
LAUST	Plan d'austénite élastique du domaine superélastique	-
LCH	Plan de transformation du domaine superélastique	-
LMART	Plan de martensite élastique du domaine superélastique	-
M_f	Température finale de la transformation martensitique	$^{\circ}C$
M_s	Température de début de la transformation martensitique	$^{\circ}C$
P_{inst}	Pente d'instabilité	MPa / m/m
Q	Coefficient de correction	$W / ^{\circ}C$

R_e	Résistance électrique	Ω
S_y	Contrainte d'écoulement	MPa
T	Température	$^{\circ}\text{C}$
T_{AMB}	Température du milieu ambiant	$^{\circ}\text{C}$
t_i	Temps du $i^{\text{ème}}$ pas de calcul	sec
t_{i-1}	Temps du $(i-1)^{\text{ème}}$ pas de calcul	sec
T_i	Température du $i^{\text{ème}}$ pas de calcul	$^{\circ}\text{C}$
T_{i-1}	Température du $(i-1)^{\text{ème}}$ pas de calcul	$^{\circ}\text{C}$
T_{max}	Température maximum limitant le domaine superélastique	$^{\circ}\text{C}$
T_{min}	Température minimum limitant le domaine superélastique	$^{\circ}\text{C}$
T_{ref}	Température de référence du modèle	$^{\circ}\text{C}$
T_{∞}	Température du milieu ambiant lors du calcul des échanges thermiques	$^{\circ}\text{C}$
V	Volume de l'échantillon	m^3
Δt	Incrément de temps	sec
ε_f	Déformation à la fin de la transformation de phase	m/m
ε_s	Déformation au début de la transformation de phase	m/m
$\varepsilon_{\text{trans}}$	Déformation de transformation	m/m
ρ	Masse volumique	kg / m^3
$\rho_{\text{élec}}$	Résistivité électrique	Ωm
σ	Contrainte	MPa
σ_{Af}	Contrainte de fin d'induction de l'austénite	MPa
σ_{Mf}	Contrainte de fin d'induction de la martensite	MPa
σ_{Ms}	Contrainte d'induction de la martensite	MPa
σ_{plast}	Contrainte de plastification	MPa

INTRODUCTION

Les alliages à mémoire de forme constituent une famille de matériaux de plus en plus utilisés dans le domaine des actionneurs. En effet, leur propriété de superélasticité permet aux AMF de générer des efforts pour une déformation qui peut atteindre 5 à 6% et ainsi remplacer certains mécanismes complexes d'activation par une seule pièce en AMF. Cependant, lors de la conception d'actionneurs AMF, il est non seulement nécessaire d'estimer la réponse mécanique de l'actionneur, mais également d'en calculer les temps de réponse. Ce mémoire présente un modèle bilinéaire du comportement thermomécanique des AMF couplé à un modèle thermique limité au cas d'un actionneur linéaire à fil droit pour lequel la source d'activation est un courant électrique qui passe dans le corps de l'actionneur (chauffage par effet Joule). Le modèle mécanique est une version simplifiée d'un modèle non-linéaire qui utilise l'interpolation par krigeage [34, 35 et 36]. Il utilise une interpolation de la déformation sur les plans définissant l'espace thermomécanique qui caractérise l'actionneur. Le modèle thermique permet de calculer les temps d'activation de l'actionneur. Il utilise les équations du transfert de chaleur en supposant que la température est uniforme dans l'échantillon et donc que les gradients de température sont négligeables dans ce dernier ("lumped capacitance method"). De plus, un coefficient de correction est nécessaire pour tenir compte des pertes par conduction sans avoir recours aux méthodes itératives de calcul. Lors de la transformation de phase, une méthode d'interpolation de l'énergie latente de transformation est utilisée [38].

Ce rapport présente une explication détaillée de la méthode de calcul employée dans le modèle, ainsi qu'une validation de ce dernier. La validation se divise en trois étapes. La première consiste en une validation qualitative qui permet d'évaluer sommairement la validité des résultats obtenus avec le modèle. La seconde étape est une validation partielle qui consiste à comparer les calculs avec des résultats de la littérature.

Finalem^{ent}, une comparaison est effectuée entre les résultats calculés et ceux mesurés en laboratoire pour des fils de TiNi.

1 - PRÉSENTATION DES AMF ET DE LEURS MODÈLES DE COMPORTEMENT THERMOMÉCANIQUE

On ne peut parler de modélisation des AMF sans introduire ces derniers, sinon on ne peut être en mesure d'apprécier toute la complexité de la tâche à accomplir. La première section de ce chapitre se consacre donc à résumer brièvement les travaux de Patoor [1] et de Funakubo [2].

1.1 Introduction aux AMF

Les propriétés remarquables des alliages à mémoire de forme sont (1) la superélasticité, qui est la propriété de l'alliage à subir d'importantes déformations élastiques, (2) la mémoire de forme, qui est la propriété de l'alliage à récupérer sa forme initiale par chauffage après avoir subi une déformation plastique, et (3) un taux élevé d'amortissement. Ces propriétés sont imputables à un changement de phase réversible sans diffusion à l'état solide entre la phase haute température (austénite) et la phase basse température (martensite) appelée *transformation martensitique*. Un tel changement de phase se produit seulement à l'intérieur d'alliages qui possèdent plusieurs phases homogènes et métastables dans la plage des températures d'opération. Plus précisément, ces dernières conditions signifient que la structure cristalline du matériau doit être figée dans un état hors équilibre à l'aide d'une trempe, à partir de la phase désirée. De plus, il est généralement nécessaire d'effectuer un revenu afin d'éliminer la sursaturation de défauts créée lors de la trempe et pour remettre en ordre la structure cristalline de l'alliage. Ces traitements thermiques sont d'une très grande importance, car ils affectent significativement les plages de température où se manifestent les différentes propriétés des AMF. Il est donc important de bien contrôler la température et la qualité des milieux de mise en solution et de trempe ainsi que la température et le temps du revenu.

Contrairement à la transformation martensitique observée dans les aciers, le changement de phase des AMF est thermoélastique, c'est-à-dire que la croissance des aiguilles (variantes) de martensite progresse avec la température ainsi que sous l'effet d'une contrainte. De plus, la transformation inverse (martensite vers austénite) est obtenue par un mouvement de retour des interfaces austénite-martensite afin de restaurer complètement la phase mère, et non par une germination de l'austénite dans la martensite, comme c'est le cas dans les aciers. Comme ce fut déjà mentionné, la transformation martensitique s'effectue sans diffusion et elle est caractérisée par de petits déplacements corrélés et finis des atomes qui créent un cisaillement homogène dans la structure cristalline.

Le phénomène d'hystérésis observé lors de la transformation martensitique inverse est attribuable au fait qu'il faut non seulement considérer l'énergie libre de Gibbs qui gouverne le changement de phase, mais également l'énergie d'interface des variantes¹ de martensite, l'énergie élastique de déformation produite par le champ de contrainte appliqué, l'énergie élastique bloquée par le champ de contraintes internes ainsi que l'énergie dissipée par frottement lors des déplacements des interfaces. Par conséquent, les transformations de phase se produisent sur un intervalle de température qui varie selon le sens de la transformation (directe ou inverse) à cause de la dissipation d'énergie. On dénote donc quatre températures caractéristiques des AMF, soit les températures de début et de fin de transformation inverse, A_s (*austénite start*) et A_f (*austénite finish*), puis celles de la transformation directe, M_s (*martensite start*) et M_f (*martensite finish*). Ces températures de transformation ont la particularité de varier linéairement en fonction de la contrainte, tel qu'illustré à la figure 1.1. En effet, il peut être démontré que les droites d'évolution des températures de transformation en fonction de la contrainte correspondent à la relation de Clausius-Clapeyron :

¹ Pour des raisons de symétrie, en général la martensite peut se former seulement selon 24 variantes possibles à l'intérieur d'une maille du réseau cristallin.

$$\frac{d\sigma}{dT} = \frac{\Delta S}{V_0 \Delta \epsilon} \quad (1.1)$$

où V_0 est le volume, ΔS est la variation d'entropie, $\Delta \epsilon$ est la variation de la déformation et $d\sigma/dT$ est le taux de variation de la contrainte en fonction de la température. Si la variation de l'entropie ΔS ne dépend pas de la température, comme c'est généralement le cas pour la majorité des alliages, l'équation 1.1 présente la relation linéaire existante entre la contrainte et la température lors de la transformation de phase.

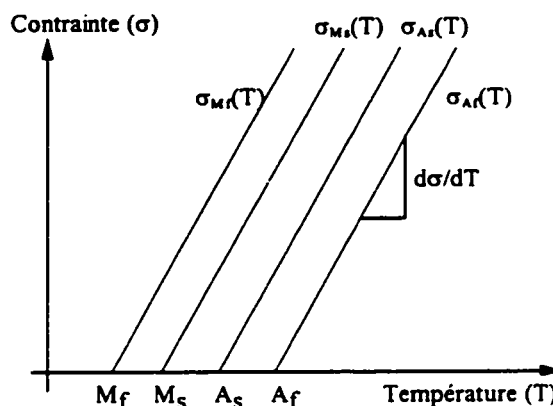


Figure 1.1 Évolution des températures de transformation en fonction de la contrainte.

1.1.1 Super-thermoélasticité

La super-thermoélasticité (figure 1.2) est la propriété des AMF à pouvoir se déformer élastiquement de quelques pourcentages (dans le cas de monocristaux, la déformation peut atteindre 10%) comparativement à quelques dixièmes de pourcentages pour les alliages classiques. Ce niveau de déformation est obtenu par l'application d'une contrainte supérieure à la contrainte de fin d'induction de martensite (σ_{Mf}) pour un échantillon austénitique. Dans ce cas, la déformation est principalement attribuable à l'orientation des variantes de martensite dans la direction de la contrainte appliquée. La déformation de transformation peut ensuite être récupérée soit par le relâchement de la contrainte, soit par l'augmentation de la température (cycle superthermique) jusqu'à une température pour laquelle la contrainte appliquée devient inférieure à la contrainte de fin

d'induction d'austénite (σ_{Af}). Notez que, lors d'un balayage en température, la contrainte doit être non-nulle sinon la martensite formée sera auto-accommodante et il n'y aura aucune déformation de transformation.

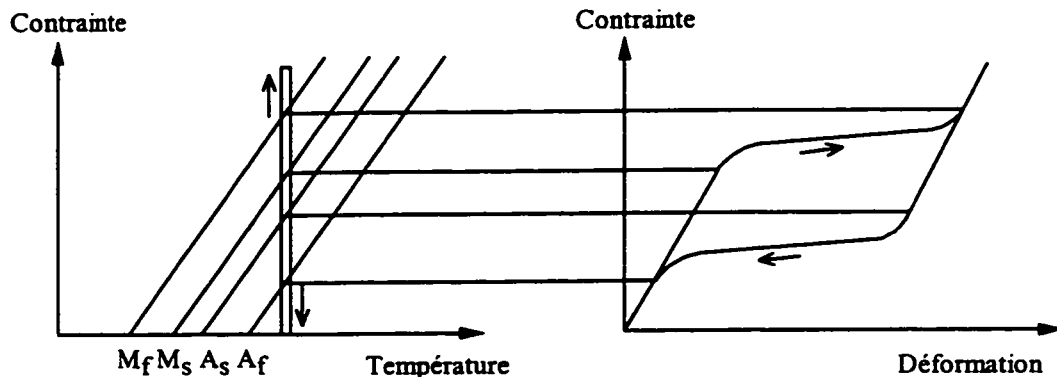


Figure 1.2 Comportement superélastique des AMF.

1.1.2 Comportement caoutchoutique et effet mémoire

Lorsqu'un AMF est sous la forme de martensite auto-accommodante à une température inférieure à M_f et que la contrainte est augmentée, une proportion croissante des variantes de martensite s'oriente de manière privilégiée dans la direction de la contrainte. Cependant, lors du relâchement de la contrainte, une déformation résiduelle persiste (figure 1.3). La martensite étant la phase de plus faible énergie à une température inférieure à M_p , la fraction de martensite orientée est à un niveau d'énergie plus stable et ne revient donc pas à sa configuration initiale, ce qui explique la déformation résiduelle. Ce phénomène est appelé *comportement caoutchoutique*. Par la suite, si on chauffe l'alliage à une température supérieure à A_f (figure 1.4), tous les cisaillements microscopiques du réseau cristallin générés par la martensite se résorbent et l'alliage reprend sa configuration initiale : c'est l'*effet mémoire simple sens*. Si un certain nombre de cycles thermiques sont effectués à contrainte non nulle, un réseau de dislocation se formera progressivement, favorisant l'apparition de certaines variantes de martensite au détriment des autres. Ainsi, une forme géométrique à basse température de l'alliage sera favorisée, c'est ce qu'on appelle l'éducation de l'alliage. Un alliage

éduqué qui subit un échauffement à contrainte nulle reprendra toujours sa configuration initiale, tandis qu'au refroidissement, toujours à contrainte nulle, il prendra la forme géométrique imposée par son éducation. Ce phénomène est appelé *effet mémoire double sens*.

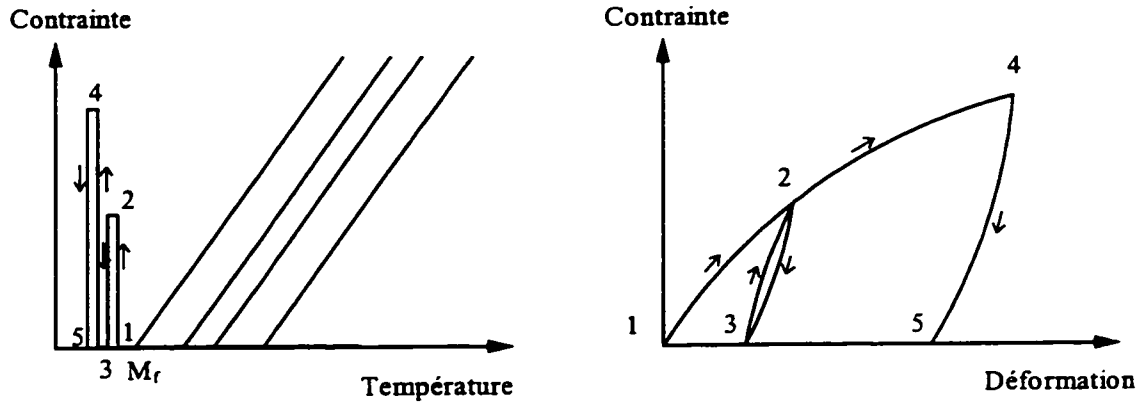


Figure 1.3 Comportement caoutchoutique des AMF.

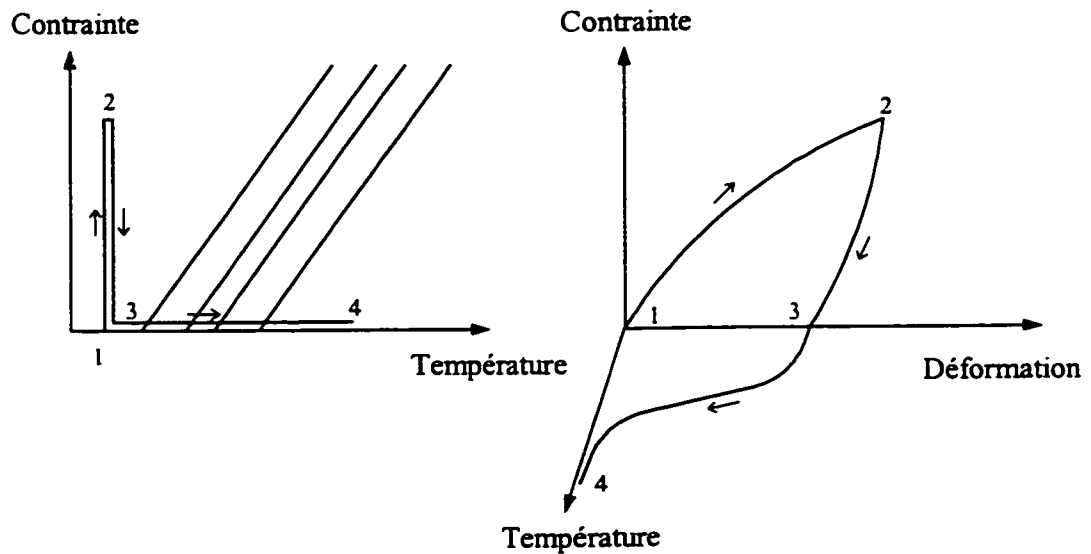


Figure 1.4 Effet mémoire simple sens.

Cependant, ce phénomène d'éducation, relié au phénomène de fatigue, qui permet d'obtenir l'effet mémoire double sens, produit également plusieurs effets indésirables dont le glissement des températures de transformation, la variation de l'étalement de la transformation et de la largeur de l'hystérésis et l'augmentation de la déformation permanente. L'évolution de ces caractéristiques est différente pour chaque AMF. Pour la majeure partie d'entre eux, les changements apparaissent lors des premiers cycles, puis se stabilisent. Cependant, trop de facteurs métallurgiques influencent le comportement en fatigue des AMF pour qu'il soit possible de le prédire sans avoir recours à des études expérimentales.

1.1.3 Effet amortissant

Lors de sollicitations cycliques, le frottement entre les aiguilles de martensite et l'austénite ainsi que la dissipation d'énergie imputable à la transformation de phase, confèrent aux AMF d'excellentes propriétés amortissantes. Notez qu'il est préférable d'utiliser un AMF en phase martensitique ou dans la zone de transformation à la température d'utilisation afin de maximiser le frottement entre les aiguilles de martensite. En ce qui concerne les sollicitations par impact, les propriétés amortissantes des AMF sont encore à l'étude.

1.2 Revue bibliographique sur la modélisation du comportement thermomécanique des AMF

Bien que les alliages à mémoire de forme constituent un domaine de recherche relativement jeune (1930), le nombre croissant d'applications industrielles a grandement contribué au démarrage des travaux sur la modélisation de leur comportement thermomécanique. Le but recherché par l'élaboration de modèles consiste à réduire les coûts et les temps de conception de pièces en AMF en plus d'aider les concepteurs à mieux cerner les différents mécanismes qui régissent le comportement particulier des

alliages à mémoire de forme. Un bon modèle thermomécanique devrait idéalement posséder les caractéristiques suivantes :

- simuler les états tridimensionnels de contrainte;
- simuler les comportements thermomécaniques remarquables des AMF pour tous les régimes de température;
- modéliser l'hystérésis et les sous-cycles;
- le modèle doit pouvoir être implanté dans un algorithme de calcul afin de simuler la réponse de pièces complexes;
- exiger un minimum de manipulations en laboratoire pour obtenir les propriétés intrinsèques de l'alliage;
- tenir compte de l'altération des propriétés du matériau causée par la fatigue.

Plusieurs approches ont été expérimentées par différents auteurs, dont certaines sont très générales tandis que d'autres consistent à modéliser des aspects précis du comportement des AMF. Certains auteurs, tels que Hawkins et al. [3], Liang et Rogers [4] ainsi que Baz et al. [5], utilisent déjà une analyse par éléments finis, mais avec tellement d'hypothèses qu'ils contournent les problèmes propres aux AMF. Bien que les simplifications apportées par chacun des auteurs soient justifiées dans la mesure où la proportion d'AMF dans le dispositif modélisé est faible, elles deviennent inadmissibles pour un modèle exclusivement consacré aux alliages à mémoire de forme. Mentionnons cependant, comme nous le verrons un peu plus loin, que certains auteurs ont réussi à développer des modèles d'éléments finis adéquats pour les AMF. Il existe deux classes principales de modèles qui permettent de simuler le comportement complexe des AMF : les modèles micromécaniques et les modèles phénoménologiques.

1.2.1 Modèles micromécaniques

Ce type d'approche, contrairement au modèle phénoménologique, présente une explication des phénomènes observés en plus de les décrire. Le comportement

macroscopique de l'alliage à mémoire de forme est obtenu en effectuant une moyenne des comportements de chaque petit volume de matière qui compose la pièce modélisée.

Travaux de Patoor

Le modèle de Patoor et al. repose sur la détermination de la fraction de martensite pour chaque variante de chaque grain par la minimisation d'un potentiel thermodynamique de Gibbs pour un volume polycristallin de matière. La déformation macroscopique de l'alliage est calculée à l'aide d'une opération de sommation pondérée par la proportion volumique de chaque grain². La résolution de ce potentiel, qui est fonction de la température, du tenseur des contraintes et des taux de martensite transformée de chaque variante dans chaque grain, peut être accomplie par un algorithme de minimisation linéaire [6] ou quadratique [7] ou bien par une méthode "auto-cohérente" [8, 9, 10, 11]. La méthode de minimisation linéaire présente l'avantage de faciliter le travail avec un état tridimensionnel de sollicitation, mais elle néglige l'énergie d'interaction entre les différentes variantes de martensite et ne permet pas de traiter le phénomène d'hystérésis. Le modèle avec minimisation quadratique (travaux actuellement en cours) devrait permettre de tenir compte de cette énergie d'interaction, ce qui conduira à une modélisation plus précise présentant les mêmes avantages et inconvénients que la minimisation linéaire. La méthode "auto-cohérente" utilise une forme mathématique différente du potentiel thermodynamique obtenu par minimisation quadratique qui permet de tenir compte des interactions entre les variantes de martensite ainsi que d'un potentiel dissipatif. Ce potentiel dissipatif (technique fortement inspiré de Sun et Hwang [12]), permet de reproduire, pour l'état superélastique seulement, le phénomène d'hystérésis qui était absent des modèles précédents. Malheureusement, aucune validation avec des résultats expérimentaux n'est disponible.

² Il est possible d'utiliser une approche statistique pour simuler le nombre et l'orientation des grains.

Travaux de Sun et Hwang

Sun et Hwang [12] comme Patoor ont développé un modèle fondé sur la minimisation d'un potentiel thermodynamique. Cependant, afin de déterminer la déformation macroscopique d'une pièce, ils utilisent un seul taux de martensite global ainsi qu'un taux de martensite orienté pour tenir compte de la réorientation des variantes. Bien que leur modèle couvre presque toutes les plages des températures caractéristiques d'un AMF (effet caoutchoutique, effet mémoire et superélasticité), leur méthode de résolution les oblige à développer des équations constitutives différentes pour chacune d'entre elle, ce qui complique l'implantation de leur modèle dans un module de calcul. Plusieurs validations expérimentales ont été effectuées, mais pour des alliages qui possèdent un comportement bilinéaire (le comportement thermomécanique peut être approximé par des droites) tel que le TiNi et le AuCd.

Travaux de L'excellent

L'étude des alliages à mémoire de forme de L'excellent est fondée sur une approche thermodynamique. L'excellent a développé différents modèles en collaboration avec plusieurs personnes, notamment Licht [13], Raniecki et al., Leclerc et al., ainsi que Bourdon. L'excellent et Raniecki [14] ont développé un modèle du comportement superélastique qui peut comporter des états tridimensionnels de contrainte, ainsi que calculer des sous-cycles lors de la transformation de phase. L'avantage de ce modèle est que toutes les constantes nécessaires aux calculs peuvent être obtenues à partir de courbes de traction uniaxiale. La validation expérimentale effectuée montre que le comportement superélastique du modèle s'approche assez bien de la réalité, mais il ne donne aucun résultat en ce qui a trait aux calculs des sous-cycles.

Leclerc et al. [15] ont par la suite proposé une version simplifiée du modèle dans le but de l'appliquer à la transition R (transformation de phase observée dans certains AMF en plus de la transformation martensitique) afin de calculer la force de rappel lors de ce changement de phase vers l'austénite. Ils ont cependant négligé le phénomène

d'hystérésis et les résultats de la validation expérimentale sont peu satisfaisants. Une seconde publication de Leclerc et al. [16] propose un modèle fondé sur une méthode incrémentale appliquée aux éléments finis. Ils travaillent avec des états tridimensionnels de contrainte, mais la température est prise en compte seulement en régime superélastique. Malheureusement, ce modèle n'est validé que pour des essais de traction simple.

Les travaux avec Bourdon [18] reprennent le potentiel d'énergie libre de Raniecki et al. [17] en ajoutant cependant une simulation du comportement en fatigue à l'aide d'un terme de contribution cyclique. Le modèle comprend également une variable supplémentaire qui tient compte de la fraction de martensite stabilisée qui serait responsable de la perte progressive des propriétés AMF lors de sollicitations cycliques. Cependant, ce modèle présente plusieurs limitations, dont la difficulté à modéliser la fatigue thermomécanique, et le fait qu'il faut connaître l'évolution de la fraction de martensite résiduelle en fonction du nombre de cycles. La validation présente d'excellents résultats pour les TiNi en traction uniaxiale (régime superélastique à température constante), mais aucun résultat n'est fourni pour les alliages cuivreux qui possèdent un comportement beaucoup plus non-linéaire.

Modélisation par éléments finis

Ganghoffer et al. [19] étudient l'évolution de la transformation martensitique à l'intérieur d'un grain de forme carrée par la méthode des éléments finis. Leurs résultats ont permis de faire ressortir l'importance des phénomènes d'autoaccommodation, ainsi que des contraintes internes dans la formation de plaquettes de martensite. Collins et Luskin [20] ont également étudié la formation de martensite dans un volume de matière à l'aide d'éléments finis. Malheureusement, leur approche ne s'applique qu'au type de transformation de phase que l'on retrouve dans les alliages très peu utilisés d'indium-thallium. À l'image des travaux de Ganghoffer et al., les résultats obtenus par Collins et

Luskin permettent une meilleure compréhension du phénomène de formation des plaquettes de martensites, mais ils sont peu utiles pour la modélisation de pièces réelles.

1.2.2 Modèles phénoménologiques

Ce type de modèle est purement descriptif, c'est-à-dire qu'il propose une description du phénomène étudié sans toutefois fournir d'explication sur le phénomène physique proprement dit. Il devient donc important de connaître l'allure du phénomène à l'aide de données expérimentales.

Modèles fondés sur l'approche de Tanaka

Tanaka [21] a développé des équations constitutives thermomécaniques ainsi que des équations de cinétique de transformation obtenues à partir de considérations thermodynamiques. Il a ensuite appliqué [22] ces équations à un modèle capable de modéliser la superélasticité, l'effet mémoire simple sens, l'effet caoutchoutique ainsi que de calculer la force de rappel lors d'un cycle superthermique à déplacement nul (transformation de phase induite par un cyclage en température). Il exprime la variation de la déformation comme étant le résultat de trois contributions : (1) variations de contraintes, (2) variations de température et (3) proportion de martensite transformée. Bien que le modèle soit développé pour un état tridimensionnel de contrainte, il n'est applicable que pour des cas simples de chargement car, comme nous l'avons vu précédemment, il est très difficile d'obtenir le taux de martensite transformée en fonction de la contrainte et de la température dans un cas de chargement complexe. De plus, le modèle est incapable de traiter les sous-cycles d'hystérésis. Cependant, les validations expérimentales pour des alliages de TiNi (isothermes) ont présenté d'excellents résultats pour l'effet caoutchoutique et la superélasticité.

Tobushi et al. [23] ont appliqué les concepts développés par Tanaka pour modéliser la transition R dans les alliages TiNi en traction uniaxiale. Une fois de plus, le phénomène d'hystérésis n'a pas été étudié. Par contre, la validation expérimentale montre que le

modèle semble bien respecter toute la région de la transition R, mais seulement dans le cas du chargement. Tobushi et al. [24] ont repris cette approche pour modéliser les propriétés cycliques d'un alliage de TiNi. Ils abordent les propriétés (1) d'augmentation des contraintes et températures de transformation lorsque l'échantillon est soumis à des cycles de chargements thermiques à contrainte constante, (2) d'augmentation de la contrainte résiduelle avec le nombre de cycle et (3) de diminution de l'allongement en fonction du nombre de cycles. Le modèle a été validé pour la fatigue superélastique (cycles à contrainte variable pour une température constante) ainsi que pour la fatigue thermomécanique (cycles à température variable pour une contrainte constante). Par contre, ils ont rencontré les mêmes difficultés que Tanaka avec son modèle sans fatigue [22].

Brinson et Lammering [25] ont davantage développé les concepts de Tanaka dans le but de les appliquer aux éléments finis. Ils ont développé les équations d'un élément tige pouvant supporter la traction uniaxiale³, mais les auteurs font remarquer que les transferts de chaleur sont négligés et qu'il serait important que cette lacune soit corrigée lors de développements ultérieurs. Une validation expérimentale de l'élément tige pour son comportement superélastique ainsi que pour le calcul de la force de rappel lors d'un chauffage à contrainte constante, offrent des résultats respectables. De plus, une modélisation d'une bague de serrage à l'aide de plusieurs éléments tige joints bout à bout est comparée aux résultats de calculs analytiques. Les résultats obtenus sont également respectables.

Bien que Boyd et Lagoudas [26] aient travaillé sur la modélisation de matériaux composites à base d'AMF, il est important de mentionner ici leur importante contribution aux modèles fondés sur les travaux de Tanaka, soit l'expression de la proportion de martensite formée en fonction de la température et de la contrainte pour un

³ À l'instar des modèles présentés précédemment, la nécessité d'exprimer les taux de martensite en fonction de la température et de l'état de contrainte limite le modèle.

état tridimensionnel de contrainte. Cependant, leur modèle traite seulement le cas des composites à base d'AMF et non celui des AMF "purs".

Modèles bilinéaires

Les modèles appelés bilinéaires sont des modèles qui approximent les domaines de l'austénite, de la transformation de phase et de la martensite par trois droites distinctes. Bien entendu, seul le domaine superélastique peut ainsi être représenté. Müller et Xu [27] proposent un modèle fondé sur une approche thermodynamique. En traçant les courbes contrainte-déformation isothermes correspondants au potentiel d'énergie libre, une zone d'instabilité correspondant à la transformation de phase devient alors apparente (figure 1.5a). À partir de constructions géométriques suivies d'un calcul de surface, il est possible d'obtenir une courbe contrainte-déformation caractérisée par une hystérésis traversée par la droite théorique d'équilibre thermodynamique (figure 1.5b). Dans ce cas, l'enveloppe de l'hystérésis est délimitée par deux droites parallèles à l'axe de la déformation, ce qui sous-entend que la transformation de phase se fait à contrainte constante, ce qui est généralement faux, particulièrement dans le cas d'alliages polycristallins. Les sous-cycles sont considérés comme étant un retour élastique jusqu'à la droite théorique d'équilibre thermodynamique suivi d'une transformation de phase à contrainte constante jusqu'à la droite élastique de l'austénite (figure 1.5b). Ce modèle semble simple à utiliser, mais les coefficients nécessaires pour décrire l'énergie libre du système sont difficiles à déterminer. La validation expérimentale pour des monocristaux offre de bons résultats, mais il est difficile d'imaginer que le modèle soit aussi précis dans le cas d'alliages polycristallins.

Brandon et Rogers [28] proposent un modèle qui ressemble beaucoup à celui de Müller et Xu à l'exception de la pente de transformation qui peut être non-nulle et donc mieux adapté aux polycristaux. Cependant, ils ne font que proposer une façon de traiter l'hystérésis et ses sous-cycles et ce, sans vraiment développer les équations qui justifient

cette approche. Il reste donc énormément de travail à faire avant que le modèle puisse correspondre aux mesures expérimentales.

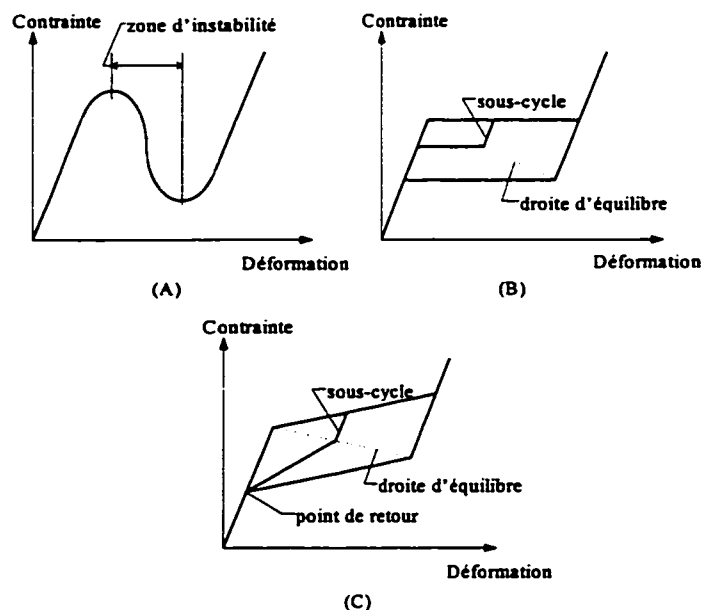


Figure 1.5 Modèles bilinéaires.

Le modèle de Gillet et al. [30,31,32] utilise le calcul du seuil de transformation d'un monocristal pour une température et un état tridimensionnel de contrainte donné de Patoor et al. [29]. Ils élaborent ensuite une loi d'écoulement qui tient compte de la variation du tenseur des contraintes et de la température. L'aspect phénoménologique du modèle s'explique par l'extension des calculs au polycristal par la supposition de l'allure de certaines fonctions. Cependant, le modèle n'est pas en mesure de prévoir la fin de la transformation. Notez que les hypothèses proposées ressemblent à celles élaborées par Müller et Xu ainsi que par Brandon et Rogers, sauf qu'après le retour élastique jusqu'à la droite théorique d'équilibre thermodynamique, le retour ne s'effectue pas parallèlement à la droite de transformation de phase, mais vers une seule destination appelée "point de retour" (figure 1.5c). Des calculs effectués pour un ressort en tension présentent une impressionnante concordance aux les résultats expérimentaux, autant au niveau de

l'enveloppe de l'hystérésis que des sous-cycles. L'aspect tridimensionnel des lois d'écoulement et des seuils de transformation a également été validé à l'aide d'essais de flexion 3-points et 4-points pour différentes températures. Les résultats confirment qu'il y a une étroite concordance entre la théorie et la pratique.

Travaux d'Ortin

Ortin [33] a développé un modèle pour calculer les sous-cycles d'hystérésis. Le modèle est fondé sur une fonction (la même au chargement et au déchargement) utilisée pour connaître la variation de la déformation associée à une contrainte initiale de relâchement sur l'enveloppe d'hystérésis ainsi qu'à une contrainte comprise à l'intérieur de cette enveloppe. L'aspect intéressant de ce modèle est qu'il serait éventuellement possible d'utiliser une seule courbe expérimentale pour déterminer les données initiales en prévoyant cependant un facteur d'échelle pour s'ajuster à tous les cas de chargements possibles. La validation expérimentale, bien qu'elle offre d'excellents résultats, n'est réalisée que pour des monocristaux soumis à une traction uniaxiale. Il s'agit maintenant de savoir si les résultats seraient aussi bons pour des polycristaux soumis à des états tridimensionnels de contraintes. De plus, connaître la fonction d'Ortin pour un modèle plus général (différents types de sollicitation à différentes températures) s'avère être une tâche ardue.

Modèles surfaciques par krigeage

Terriault et Trochu [34] utilisent la méthode du krigeage pour modéliser le comportement superélastique isotherme d'un AMF. Le krigeage est une méthode numérique d'interpolation, fondée sur une méthode d'origine statistique, d'une quantité de données connues. Chaque donnée est décrite par ses coordonnées et par des observations (ex. : temps, vitesse, viscosité, etc.). Le krigeage permet de retrouver par interpolation n'importe quelle observation associée à des coordonnées contenues dans le domaine défini par les données initiales.

Ce modèle ne nécessite qu'une courbe superélastique, comprenant le cycle complet d'hystérésis et quelques sous-cycles, discrétisés en une série de points. À partir de cet ensemble de points et du taux de martensite transformée, le modèle est par la suite en mesure de calculer autant de sous-cycles que nécessaire et ce, de façon récursive, c'est-à-dire qu'on peut aussi calculer des sous-cycles à l'intérieur d'autres sous-cycles. Cette approche a l'avantage de pouvoir représenter un grand nombre de sollicitations différentes. En effet, si les courbes utilisées initialement sont, par exemple, la déflexion d'un ressort AMF en fonction de la charge, alors la réponse du modèle représentera le comportement de ce ressort en tension. Cependant, lorsque la contrainte n'est pas constante dans toute la pièce, il devient plus difficile d'appliquer ce modèle.

Terriault et al. [35] ont étendu ce modèle à tout le domaine des AMF (superélasticité, effet caoutchoutique et effet mémoire). Le nombre de courbes initiales à discrétiser est inévitablement plus important (quelques unes pour chaque domaine), mais ces dernières peuvent être obtenues à température constante. À partir de ces courbes, le modèle krige deux surfaces distinctes : la première représente le chargement, tandis que la seconde représente le déchargement. De la même façon qu'avec le modèle Terriault et Trochu, il est possible de représenter n'importe quel type de chargement (incluant des chargements de fatigue) sur une pièce simple et ce, pour n'importe quelle plage de température. À titre d'exemple, on peut citer les travaux de Meunier et al. [36] qui ont modélisé la fatigue thermomécanique d'un ressort en compression. Cependant, dans le cas de la fatigue, le modèle ne permet de prévoir que les cycles mesurés.

2 - PRÉSENTATION DU MODÈLE

Le but recherché dans le développement du présent logiciel est de pouvoir modéliser des actuators AMF à fils droits activés par courant électrique. Le type d'essai à modéliser consiste donc à cycler thermiquement une ou plusieurs fois un échantillon à l'aide d'un courant électrique. La figure 2.1 schématise le type de cycle que le modèle de calcul de l'actuateur à fil droit doit être en mesure de reproduire. À partir d'un fil austénitique au repos, on fixe un poids à l'échantillon afin d'induire la contrainte nécessaire pour le rendre martensitique (figure 2.1, étape 1). Ceci génère une déformation de transformation dans l'échantillon. La même charge est conservée tout au long de l'essai. Un courant électrique est ensuite passé dans l'échantillon afin que celui-ci s'échauffe par effet Joule (figure 2.1, étape 2). L'échantillon redevient alors austénitique et récupère sa déformation de transformation. Lorsque le courant est interrompu, l'échantillon se refroidit et la transformation inverse a lieu, il redevient donc martensitique et récupère sa déformation de transformation (figure 2.1, étape 3). Ce type d'analyse où l'échantillon est cyclé en température est appelé *cycle superthermique*. Le modèle doit calculer le temps de déformation de l'actuateur à fil droit en fonction de la charge appliquée, du courant induit et des conditions d'opérations.

Le logiciel, présentement appelé WIREAMF, se divise en deux parties principales couplées ensemble, soit un module mécanique et un module thermique. Le format des fichiers utilisés lors de l'exécution du logiciel est présenté à l'annexe IV. L'annexe V contient la documentation de toutes les fonctions utilisées dans les différents logiciels de modélisation bilinéaire des AMF dont WIREAMF. À cette étape-ci, il peut imposer à l'actuateur trois types de conditions limites :

- (1) contrainte constante;
- (2) contrainte variable (appui sur un contre-ressort¹);

¹ Le comportement du contre-ressort peut être linéaire ou non-linéaire.

(3) dans les deux cas précédents, une butée rigide peut limiter la course de l'actuateur.

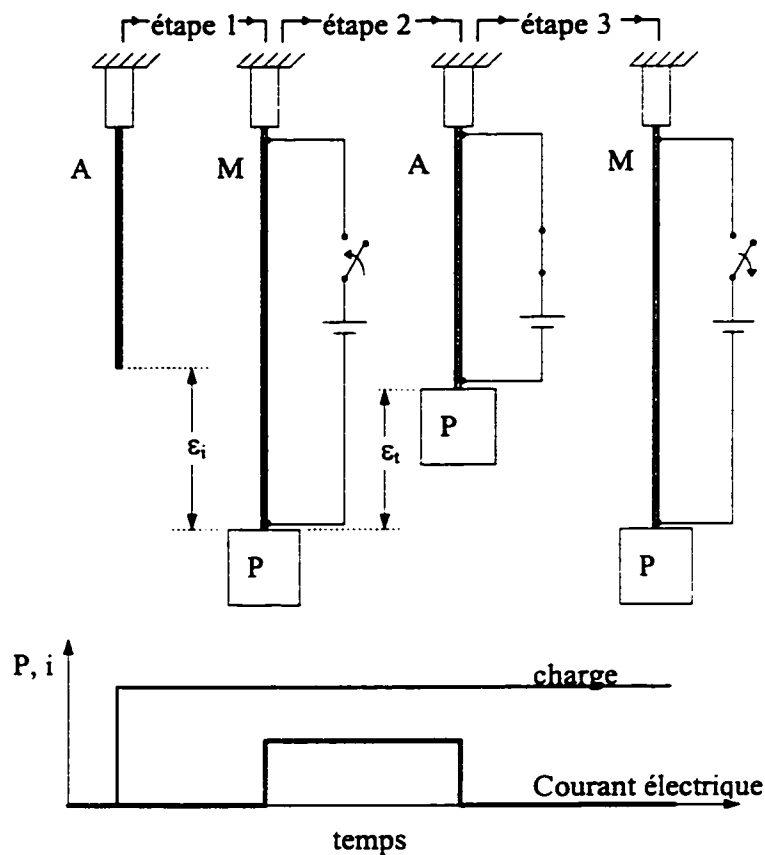


Figure 2.1 Cycle caractéristique d'utilisation d'un actuateur AMF.

2.1 Modèle d'analyse mécanique

Le modèle d'analyse mécanique est séparé en deux parties qui permettent de traiter les cas de sollicitation mentionnés plus haut, c'est-à-dire que l'actuateur est soit soumis à une contrainte constante, soit appuyé sur un contre-ressort (contrainte variable). De plus, chaque sous-module peut traiter le cas d'une butée rigide qui limite la course de l'actuateur.

2.1.1 Modèle à contrainte constante

Le module mécanique est fondé sur la version bilinéaire simplifiée du modèle KRIGAMF [37]. KRIGAMF est un modèle non-linéaire qui utilise l'interpolation par krigeage de données expérimentales et de données calculées afin de prédire le comportement thermomécanique d'un AMF. La figure 2.2 présente l'espace d'interpolation créé par le modèle KRIGAMF. Cette version bilinéaire du modèle se limite à représenter seulement l'état superélastique de l'AMF (délimité par des traits plus foncés sur la figure 2.2), c'est-à-dire pour des températures supérieures à A_f . Cette limitation permet néanmoins d'étudier un grand nombre d'applications qui utilisent les propriétés superélastiques des AMF.

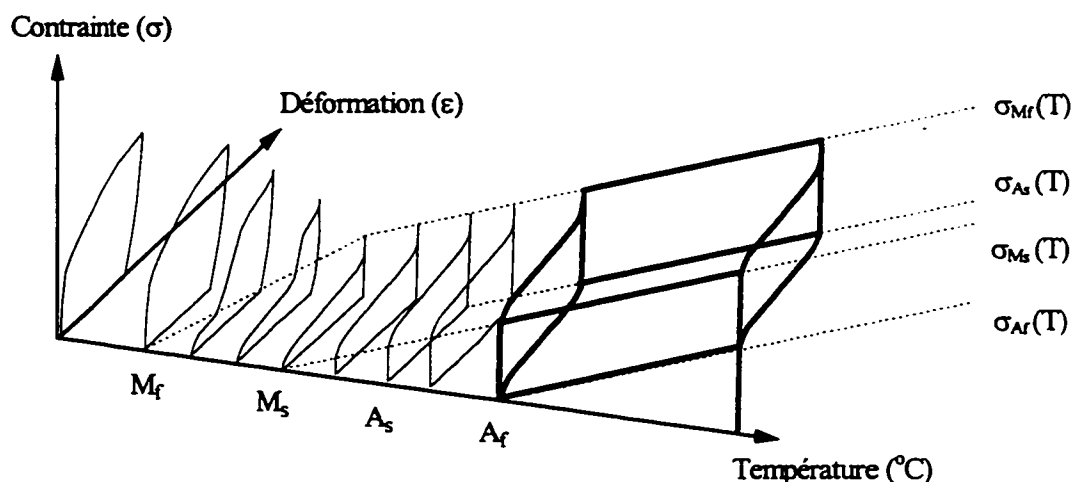


Figure 2.2 Espace d'interpolation du modèle KRIGAMF.

Au départ, les trois plans de base du domaine, soit LAUST, LCH et LMART (austénite élastique, transformation-charge et martensite élastique) tel qu'illustré à la figure 2.3, sont calculés à partir des données suivantes (figures 2.4 et 2.5) :

- (1) les modules de Young des trois plans : E_1 , $d\sigma/d\varepsilon$ et E_2 ;
- (2) la déformation de transformation $\varepsilon_{\text{trans}}$;

- (3) la limite d'écoulement S_y , la contrainte d'induction de transformation directe σ_{Ms} et celle de fin d'induction de transformation inverse σ_{Af} ;
- (4) la pente $d\sigma/dT$ et la pente d'instabilité $P_{inst} = (\sigma_{Ms} - \sigma_{As}) / (\epsilon_{trans} - E_2(\sigma_{Mf} - \sigma_{As}))$.

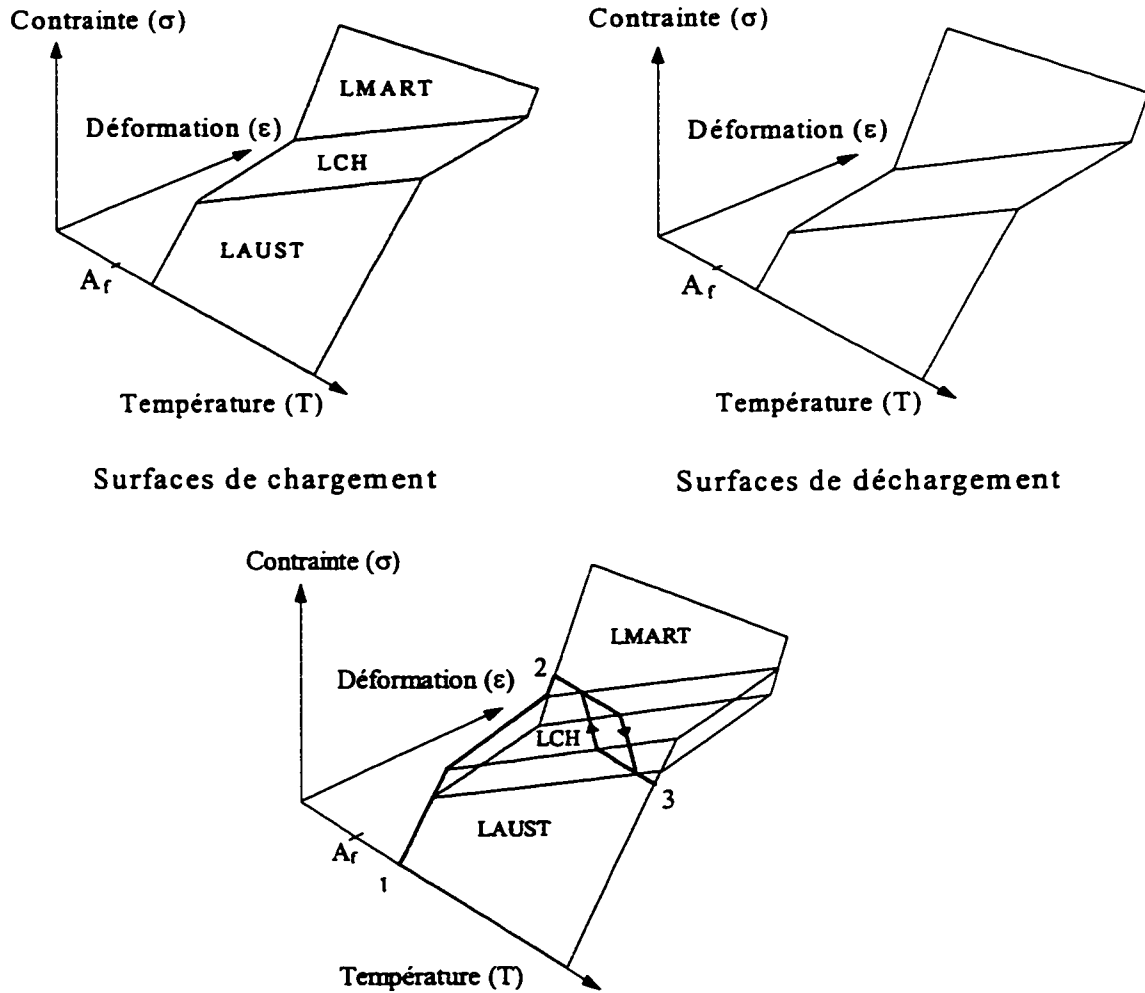


Figure 2.3 Espace d'interpolation du modèle bilinéaire.

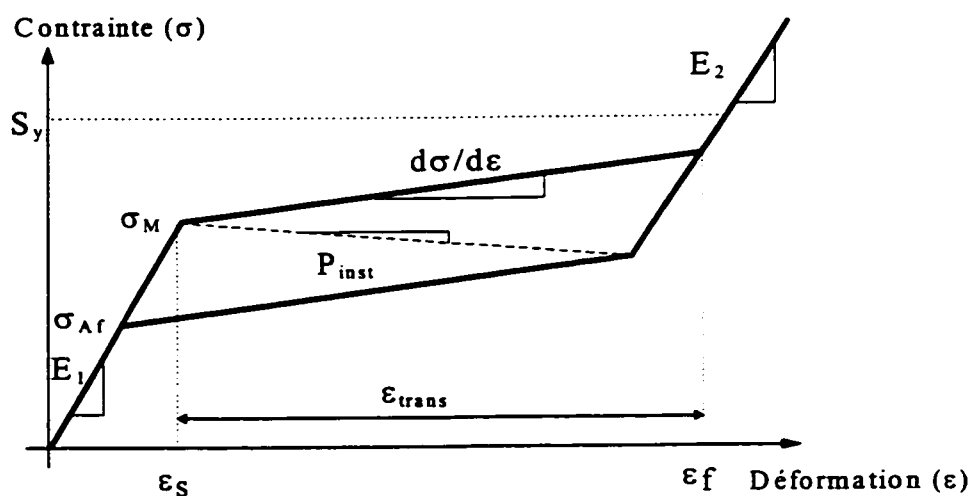


Figure 2.4 Courbe de traction superélastique idéalisée.

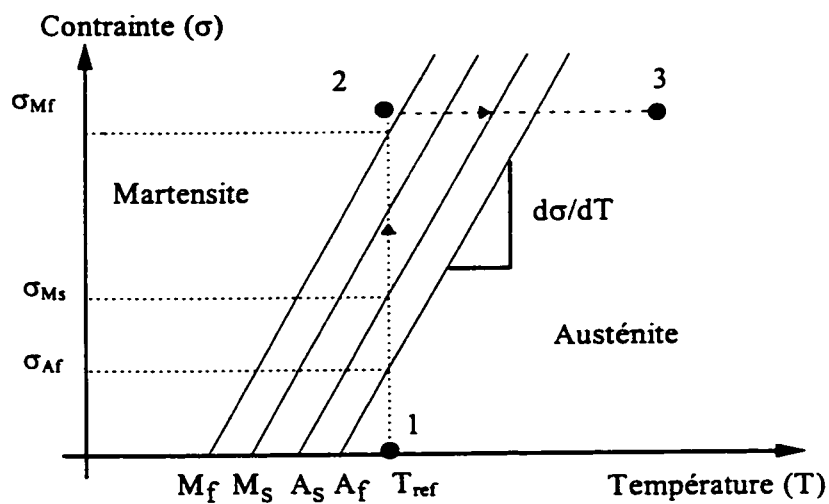


Figure 2.5 Diagramme d'état simplifié.

La séquence de calcul suit pas à pas la procédure de l'essai telle qu'expliqué précédemment (figure 2.1). La première étape est l'initialisation du modèle à une contrainte et une déformation nulle à la température de référence (température de l'échantillon au début des calculs d'échanges thermiques). Le premier point de la séquence de chargement (point 1 sur la figure 2.5) doit donc être le point de contrainte

nulle à la température de référence. La seconde étape est l'application d'une contrainte, assez grande pour dépasser, à la température de référence, la contrainte de fin d'induction de martensite σ_{Mf} (point 2 sur la figure 2.5). Cette condition est nécessaire pour obtenir à l'échauffement une transformation de phase complète dans l'échantillon (ligne 2-3 sur la figure 2.4) et obligatoire pour débiter correctement le calcul du temps lors du cyclage en température. Le cas des cycles partiels en cours de calcul sera discuté plus loin. La séquence de chargement en température peut alors être appliquée.

La séquence de chargement en température est calculée à partir des températures à atteindre, ou encore à partir des temps d'échauffement et de refroidissement. Le parcours thermomécanique de l'actuateur est trouvé en interpolant la déformation correspondant au point de la séquence de chargement ou de déchargement sur le domaine superélastique. Un exemple d'un tel parcours dans le cas où la contrainte appliquée est constante est présenté à la figure 2.3.

2.1.2 Modèle à contrainte variable

La figure 2.6 illustre les étapes suivies lors du calcul de la réponse du modèle avec contrainte variable : la première étape consiste à étirer le contre-ressort de façon à fixer son extrémité libre à l'AMF ; l'étape 2 consiste à trouver le point d'équilibre du système AMF/contre-ressort à partir des données 1, 2 et 3 ; l'étape 3, l'insertion de la butée, est optionnelle et l'étape 5, omise dans le dessin, représente la sollicitation de l'actuateur AMF par le courant électrique. Le comportement du contre-ressort peut être linéaire ou non-linéaire, la seule restriction imposée est que le dernier point contenu dans le fichier du contre-ressort soit celui qui correspond à la déflexion de ce dernier lorsque l'AMF n'est pas déformé (dimension 3 sur la figure 2.6).

La première opération consiste donc à trouver le point d'équilibre entre le contre-ressort et l'AMF à l'aide de la méthode présentée ci-dessus. Ensuite, pour chaque point, la

force générée par le contre-ressort est calculée à partir de la déformation du pas de calcul précédent. Cette force est ensuite transformée en contrainte dans l'AMF et la déformation peut être interpolée sur les plans de chargement. La contrainte est donc en retard sur la déformation et la température.

2.1.3 Modèle avec une butée

La méthode est identique, sauf lorsque la butée est atteinte. La déformation est alors maintenue constante et la contrainte est interpolée sur les plans. Un calcul parallèle sans butée permet de déterminer quand l'actuateur s'éloigne de la butée au refroidissement. À ce moment, la contrainte est de nouveau maintenue fixe et la déformation est interpolée sur les plans du domaine superélastique.

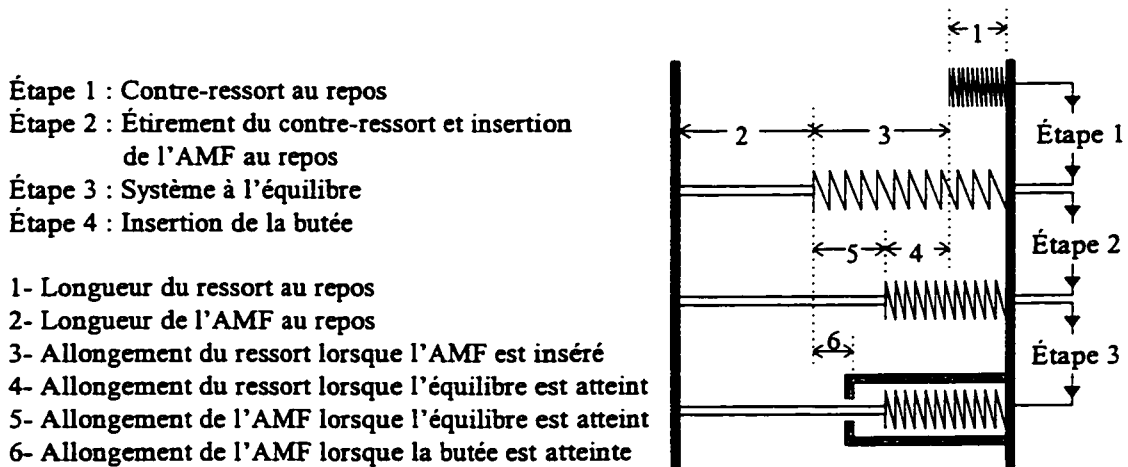


Figure 2.6 Méthode de calcul du contre-ressort.

2.2 Modèle d'analyse thermique

Le modèle d'analyse thermique permet de calculer la durée des échanges thermiques à l'aide des équations de transfert de chaleur en présence d'une source interne avec perte par convection. On suppose que la température de l'échantillon est uniforme, et donc que les gradients de température dans ce dernier sont négligeables ("lumped capacitance method"). De plus, les pertes par conduction et radiation sont négligées. Un coefficient

de correction permet de diminuer les erreurs causées par les pertes par conduction aux mors. Celles-ci en effet ne sont pas prises en compte par le modèle car on suppose que la température des mors est constante et égale à la température ambiante. Dans le cas de l'actuateur à fil droit, on considère un volume de contrôle sur une tige cylindrique. L'expression du bilan thermique schématisé à la figure 2.7 est donnée par l'équation (2.1) ci-dessous :

puissance emmagasinée = puissance produite - puissance dissipée

$$\frac{d}{dt} [\rho V C_p T] = \dot{E}_p - h A_e (T - T_\infty) \quad (2.1)$$

où $\dot{E}_p = i^2 R$ est la puissance produite par la source électrique, h est le coefficient de convection, A_e est l'aire d'échange thermique, V est le volume, ρ est la masse volumique et C_p est la chaleur spécifique.

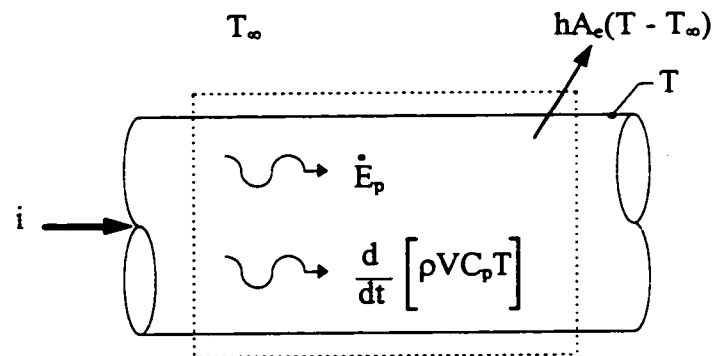


Figure 2.7 Bilan thermique d'un fil échauffé par courant électrique.

En ajoutant un coefficient de correction dans le dernier terme de l'équation (2.1), il vient

$$\frac{d}{dt} [\rho V C_p T] = E_p - (h A_e + Q) * (T - T_\infty) \quad (2.2)$$

Après séparation des variables et intégration entre T_0 et T , on obtient l'équation qui donne le temps nécessaire pour atteindre la température T à partir d'une température initiale T_0 lorsque l'actuateur AMF est chauffé par un courant électrique :

$$t = \frac{\rho C_p V}{-(h^* A_e) + Q} \ln \left[\frac{(T - T_\infty) - \frac{i^2 R}{(h^* A_e) + Q}}{(T_0 - T_\infty) - \frac{i^2 R}{(h^* A_e) + Q}} \right] \quad (2.3)$$

Le coefficient de correction est déterminé en imposant à l'argument du logarithme de ne pas être négatif. Sachant que $(T_0 - T_\infty) - \frac{i^2 R}{(h^* A_e) + Q}$ est toujours négatif, il s'ensuit que

$$T - T_\infty < \frac{i^2 R}{(h^* A_e) + Q} \quad (2.4)$$

Le coefficient de correction est donc donné par

$$Q = \left[\frac{i^2 R}{(T_e - T_\infty)} \right] - (h^* A_e) \quad (2.5)$$

où T_e est la température d'équilibre observée dans l'expérience lorsque le courant est appliqué suffisamment longtemps et T_∞ est la température du milieu ambiant.

Lors de la transformation de phase, on utilise une approximation parabolique de la chaleur spécifique C_p en fonction de la température [38]. Ceci permet de tenir compte de l'énergie latente de transformation. Pour prendre en considération l'étalement de la transformation en respectant le bilan thermique, l'algorithme suivant est utilisé :

- 1- La courbe expérimentale d'évolution de la chaleur spécifique du matériau en fonction de la température est approchée par le polynôme du deuxième ordre :

$$C_p = aT^2 + bT + d \quad (2.6)$$

où T est la température normalisée $T = (T_i - T_{min}) / (T_{max} - T_{min})$, T_i est la température courante, T_{min} est la température minimale soit A_s au chauffage et M_r au refroidissement, T_{max} est la température maximale soit A_r au chauffage et M_s au refroidissement. Ici les paramètres a , b et d sont les coefficients de la parabole d'approximation.

- 2- La valeur de l'enthalpie de transformation est obtenue après intégration de cette fonction polynomiale (2.6) par rapport à la température :

$$H = \int_{T_1}^{T_2} \rho C_p dT \quad (\text{J/m}^3) \quad (2.7)$$

- 3- La chaleur latente de transformation est $q = HV$.

$$\frac{q}{\rho V} = \frac{A(T_2^3 - T_1^3)}{3} + \frac{B(T_2^2 - T_1^2)}{2} + D(T_2 - T_1) \quad (2.8)$$

où

$$A = a / (T_{max} - T_{min})^2$$

$$B = (T_{max}b - T_{min}(b + 2a)) / (T_{max} - T_{min})^2$$

$$D = (d(T_{max} - T_{min})^2 + (a + b)T_{min}^2 - bT_{max}T_{min}) / (T_{max} - T_{min})^2$$

- 4- La température dans la zone de transformation peut ainsi être calculée en résolvant l'équation cubique par rapport à T_i :

$$\frac{A(T_2^3 - T_1^3)}{3} + \frac{B(T_2^2 - T_1^2)}{2} + D(T_2 - T_1) - \frac{\Delta q}{\rho V} = 0 \quad (2.9)$$

où $\Delta q = i^2 R \Delta t - (hA_e + Q)(T_{i-1} - T_\infty) \Delta t$ dans le cas du chauffage

et $\Delta q = (hA_e + Q)(T_{i-1} - T_\infty) \Delta t$ dans le cas du refroidissement.

Selon les conditions réelles d'échauffement, on peut fixer les températures limites de l'essai et donc calculer le temps nécessaire pour les atteindre, ou alors imposer les temps de chauffage et de refroidissement et calculer la température atteinte.

2.2.1 Température à atteindre

Dans ce cas-ci, la réponse de l'actuateur est calculée par le module du calcul mécanique. On obtient ainsi les points contrainte-déformation-température qui caractérisent la réponse de l'actuateur soumis à une sollicitation en température. Par la suite, la série des points de température est envoyée au module de calcul du temps qui, selon le plan associé à chaque point, calculera le temps avec l'équation appropriée. En effet, on détermine s'il y a transformation à l'aide du plan associé à chaque point de chargement. Par exemple, si le point se situe sur le plan d'austénite élastique (LAUST), on sait qu'il n'y a pas de transformation. Par contre, si le point se situe sur le plan LCH, il y a alors transformation de phase et on utilise la méthode d'interpolation sur la fonction de l'énergie latente de transformation trouvée à partir de l'approximation de la fonction $C_p(T)$ (équation 2.9). On obtient donc les points température-temps qui caractérisent le cycle de sollicitation de l'actuateur. En combinant les résultats des deux modules de calcul, on obtient les points contrainte-déformation-température-temps qui constituent la réponse de l'actuateur à la sollicitation en température qui a été imposée.

Cette approche permet de calculer les cas où la contrainte n'est pas constante dans l'échantillon, soit lorsqu'il y a un contre-ressort et/ou une butée. Par contre, le modèle n'est pas encore capable de calculer le temps pour des sous-cycles en température. Les cycles doivent être complets, c'est-à-dire que la transformation de phase doit être complétée avant d'entamer le refroidissement, sinon le temps calculé par le modèle sera erroné. Cette limitation est due au fait que les températures de transformation d'un AMF changent en fonction des sous-cycles en température (M , augmente). En effet, le modèle ne peut modifier les températures de transformation qu'en fonction d'une seule condition. Étant donné qu'il doit déjà être en mesure de modifier les températures de transformation en fonction de la contrainte qui peut varier, il ne peut pas calculer les sous-cycles. Cependant, le modèle peut calculer plusieurs cycles successifs si la transformation est complète.

2.2.2 Spécification de la durée des échanges thermiques

Cette approche utilise le calcul inverse de la méthode précédente, c'est-à-dire que la séquence de température associée aux temps de chauffage et de refroidissement est calculée, puis le parcours thermomécanique de l'échantillon est trouvé à partir de cette séquence. Ici, on détermine que la transformation a lieu quand les températures de début de transformation sont atteintes, et on considère que la transformation se poursuit tant que l'énergie latente de transformation n'est pas épuisée.

Cette façon de procéder limite le modèle à un cycle de chauffage-refroidissement à contrainte constante. En effet, la température étant calculée avant la déformation et la contrainte, le modèle n'est pas en mesure de prévoir la modification des températures de transformation causée par un changement de contraintes dans l'échantillon. Cependant, contrairement au cas où le temps est imposé, le cycle n'est pas obligé d'être un cycle complet, c'est-à-dire que la transformation de phase au chauffage ne doit pas nécessairement être complétée avant de débiter la séquence de refroidissement. Notez

que dans les deux cas précédents, le temps est calculé seulement lorsque la température varie, car les équations ne tiennent compte que de la durée des échanges thermiques.

2.3 Couplage des modèles mécanique et thermique

Le couplage des modèles mécanique et thermique peut être effectué de deux façons différentes, soit à partir des échanges thermiques qui sont dictés par une température à atteindre ou en fonction du temps d'échauffement prédéterminé. Dans le premier cas, le module mécanique calcule le parcours thermomécanique de l'actuateur indépendamment du temps (figure 2.3). Il transfère ensuite la séquence de température au module de calcul thermique qui, en fonction de l'intensité du courant électrique et des conditions d'échange thermique avec l'environnement, trouve le temps associé à la température pour chaque point de la séquence de chargement. Dans le second cas, le module thermique est utilisé pour déterminer la température associée à la durée des échanges thermiques. Cette séquence de température est ensuite transférée au module mécanique qui calcule le parcours thermomécanique de l'actuateur associé à la séquence de température pour une contrainte constante. La combinaison des deux modules de calcul permet donc d'obtenir les points contrainte-déformation-température-temps qui constituent le parcours thermomécanique de l'actuateur en fonction des conditions d'utilisation imposées. Le tableau 2.1 résume tous les paramètres nécessaires pour l'utilisation du modèle couplé. On y retrouve également le type d'essai expérimental requis pour les obtenir.

Afin de valider ce modèle, une démarche en trois étapes est adoptée : (1) une évaluation qualitative permet de vérifier la cohérence des résultats du modèle; (2) une comparaison des résultats des modèles mécanique et thermique est effectuée avec ceux de la littérature; (3) enfin, les résultats du modèle couplé sont comparés avec des expériences réalisées au laboratoire de l'IREQ.

Tableau 2.1 Variables utilisées pour la modélisation d'un actuateur à fil droit.

Paramètres	Type d'essai	Unité
Géométrie de l'éprouvette (longueur et diamètre)	Mesure	m
Contrainte appliquée (s)	Imposée	MPa
Température ambiante (T_{AMB})	Imposée	$^{\circ}C$
Coefficient de convection (h)	Estimé	$W/m^2 \text{ } ^{\circ}C$
Intensité du courant (i)	Imposée	A
Facteur de correction	Échauffement jusqu'à équilibre	$W/^{\circ}C$
Températures de transformations (M_s , M_f , A_s , A_f)	Calorimétrie ou traction superélastique à différentes températures (fig. 2.4)	$^{\circ}C$
Coefficients d'approximation de la courbe de calorimétrie au chauffage et au refroidissement	Calorimétrie	
Contrainte d'induction de mart. (σ_{M_s})	Traction superélastique (fig. 2.4)	MPa
Contrainte fin d'induction d'aust. (σ_{A_f})	Traction superélastique (fig. 2.4)	MPa
Pente $d\sigma/dT$	Traction superélastique à différentes températures (fig. 2.4)	$MPa/^{\circ}C$
Déformation de transformation (ϵ)	Traction superélastique (fig. 2.4)	%
Module d'élasticité en austénite (E_1)	Traction superélastique (fig. 2.4)	$MPa / m/m$
Module de transformation ($d\sigma/d\epsilon$)	Traction superélastique (fig. 2.4)	$MPa / m/m$
Module d'élasticité en martensite (E_2)	Traction superélastique (fig. 2.4)	$MPa / m/m$
Pente d'instabilité (P_{inst})	Traction superélastique (fig. 2.4)	$MPa / m/m$
Masse volumique (ρ)	Littérature ⁴	kg/m^3
Chaleur spécifique (C_p)	Littérature ⁴	$J/kg \text{ } ^{\circ}C$
Résistivité électrique (ρ_{elec})	Littérature ⁴	$\Omega \cdot m$

⁴ Un certain nombre de données étant difficiles à obtenir du point de vue expérimental, celles-ci seront estimées à partir des informations tirées de la littérature.

3 - ANALYSE QUALITATIVE DES RÉSULTATS

La première étape avant la validation du modèle repose sur l'évaluation qualitative de ce dernier. Autrement dit, il faut s'assurer que le modèle donne des résultats cohérents, sinon, il serait inutile de le valider. Trois aspects doivent être vérifiés : le calcul mécanique, le calcul du temps et l'influence de l'incrément de température imposé à l'AMF. Le tableau 3.1 présente les données utilisées lors de cette analyse.

Tableau 3.1 Données pour la validation qualitative. Matériaux : Cu-16 at.% Zn-15 at. % Al

Catégories	Données	Valeur
Géométrie	Longueur de l'éprouvette (l)	0.3 m
	Diamètre de l'éprouvette (d)	0.008 m
Chargement thermomécanique	Contrainte appliquée (σ)	245 MPa
	Température ambiante (T_{AMB})	10.0 °C
	Coefficient de convection (h)	30 W/m ² °C
	Intensité du courant (i)	75 A
	Facteur de correction	0.0 W/°C
	Butée	5.0 %
	Matériau	Température M_s
Température M_f		-74.0 °C
Température A_s		-68.5 °C
Température A_f		-61.0 °C
Coefficients d'approximation de la courbe de calorimétrie au chauffage (A, B, D)		-9200.0, 9200.0, 0.0
Coefficients d'approximation de la courbe de calorimétrie au refroidissement (A, B, D)		-9200.0, 9200.0, 0.0
Contrainte d'induction de mart. (σ_{Ms})		188.25 MPa
Contrainte fin d'induction d'aust. (σ_{Af})		178.21 MPa
Pente $d\sigma/dT$		2.51 MPa/°C
Déformation de transformation (ϵ_{trans})		8.8 %
Module d'élasticité en austénite (E_1)		24600.0 MPa / m/m
Module de transformation ($d\sigma/d\epsilon$)		213.92 MPa / m/m
Module d'élasticité en martensite (E_2)		24600.0 MPa / m/m
Pente d'instabilité (P_{inst})		100.294 MPa / m/m
Masse volumique (ρ)		6500 kg/m ³
Chaleur spécifique (Cp)		700 J/kg°C
Résistivité électrique (ρ_{elec})		0.000000725 Ω m

3.1 Analyse du calcul mécanique

Le calcul mécanique permet de déterminer trois relations : contrainte-déformation, contrainte-température et déformation-température. Les figures 3.1 et 3.2 présentent respectivement la relation contrainte-déformation lors d'un cyclage en température pour une contrainte imposée constante et pour une contrainte imposée variable. Les résultats présentés à ces deux figures montrent que le modèle respecte le comportement mécanique d'un AMF en déformation-température :

- (1) présence de l'hystérésis;
- (2) pente de transformation identique en chargement et déchargement sans butée;
- (3) les courbes avec butée respectent la pente d'instabilité;
- (4) la courbe avec un contre-ressort respecte la linéarité de ce dernier (la figure 3.6 présente la courbe caractéristique du contre-ressort utilisé).

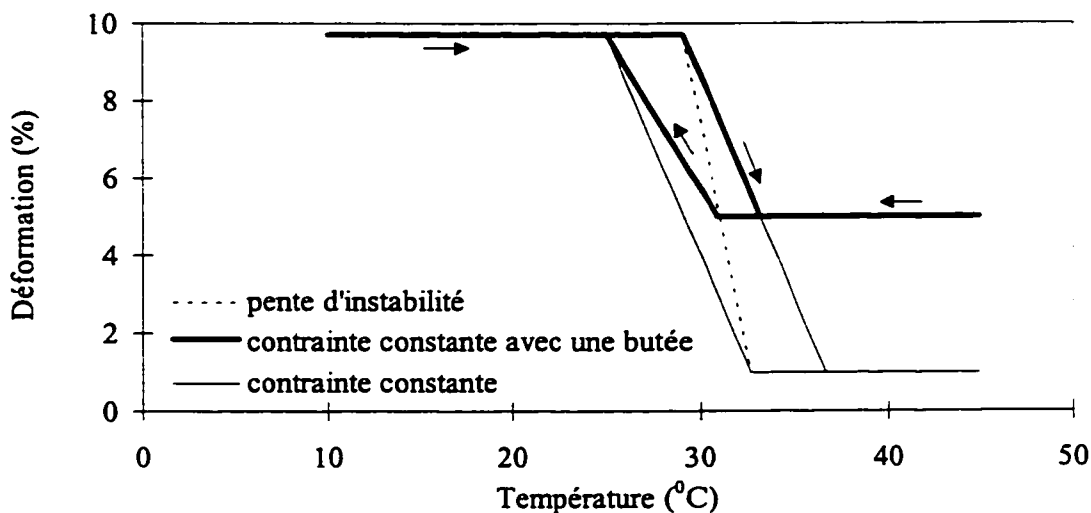


Figure 3.1 Déformation calculée en fonction de la température pour une contrainte imposée constante.

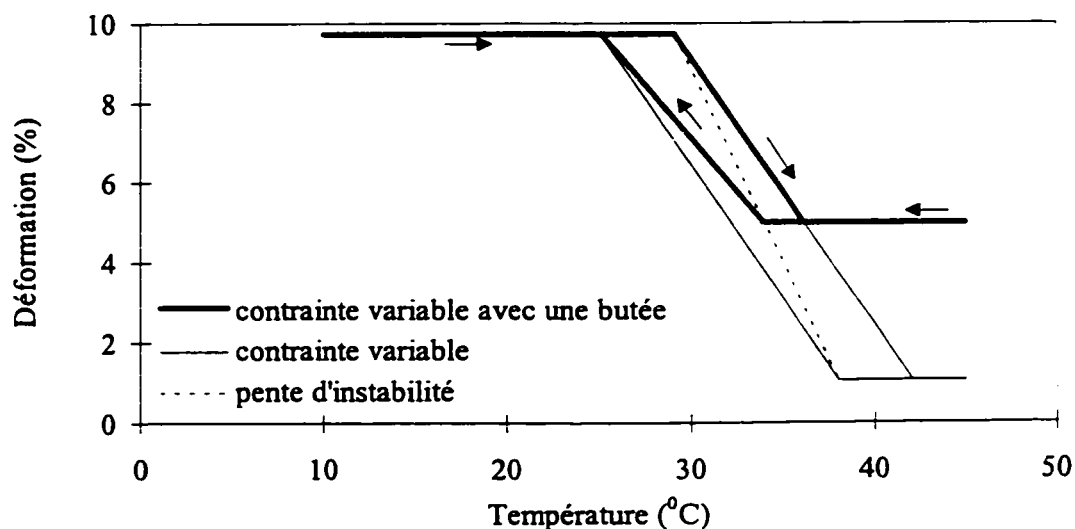


Figure 3.2 Déformation calculée en fonction de la température pour une contrainte imposée variable (contre-ressort).

Les figures 3.3 et 3.4 présentent respectivement la relation contrainte-température lors d'un cyclage en température pour une contrainte imposée constante et pour une contrainte imposée variable. Les résultats présentés à ces deux figures montrent que le modèle respecte le comportement mécanique d'un AMF en contrainte-température :

- (1) les courbes avec butée ne présentent pas de palier supérieur;
- (2) présence de l'hystérésis pour les courbes avec contre-ressort et butée;
- (3) la courbe à contrainte constante sans butée est constante;
- (4) la courbe avec un contre-ressort respecte la linéarité de ce dernier (figure 3.6);
- (5) pente de transformation identique en chargement et déchargement ;

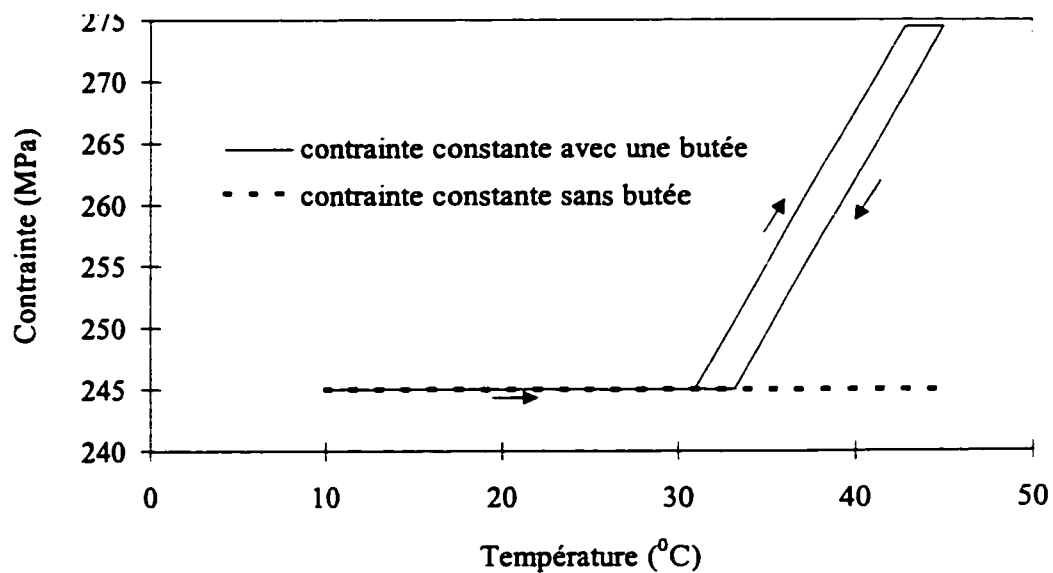


Figure 3.3 Contrainte calculée en fonction de la température pour une contrainte imposée constante.

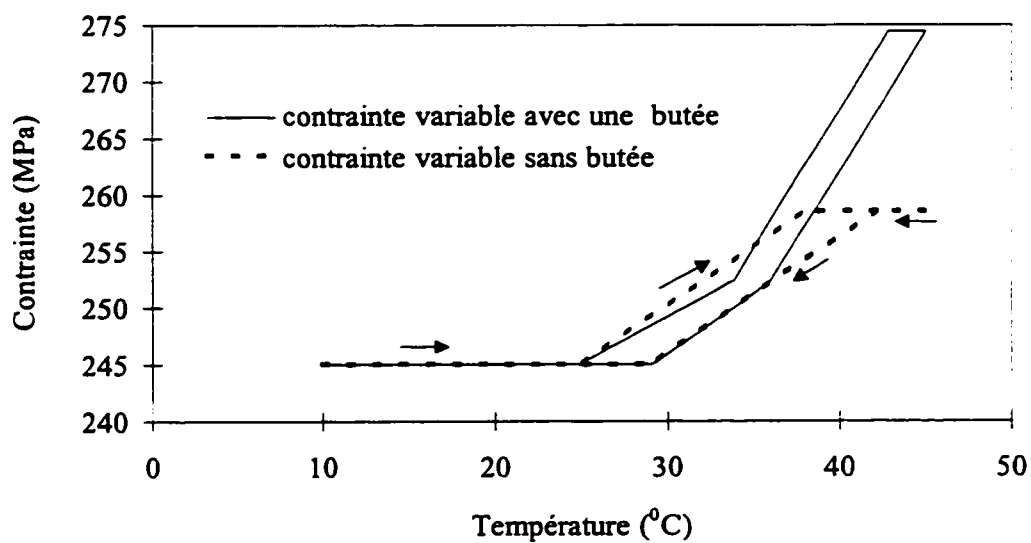


Figure 3.4 Contrainte calculée en fonction de la température pour une contrainte imposée variable (contre-ressort).

Finalement, la figure 3.5 présente la relation contrainte-déformation pour un cyclage en température. Les résultats présentés à cette figure montrent que le modèle respecte le comportement mécanique d'un AMF en contrainte-déformation :

- (1) il ne présente aucune hystérésis;
- (2) la pente de la courbe à contrainte variable est égale à la pente du contre-ressort;
- (3) la contrainte dans l'AMF augmente lorsque sa déformation est bloquée par une butée;
- (4) le niveau de déformation des courbes avec une butée est limité à la valeur imposée par la butée (tableau 3.1)

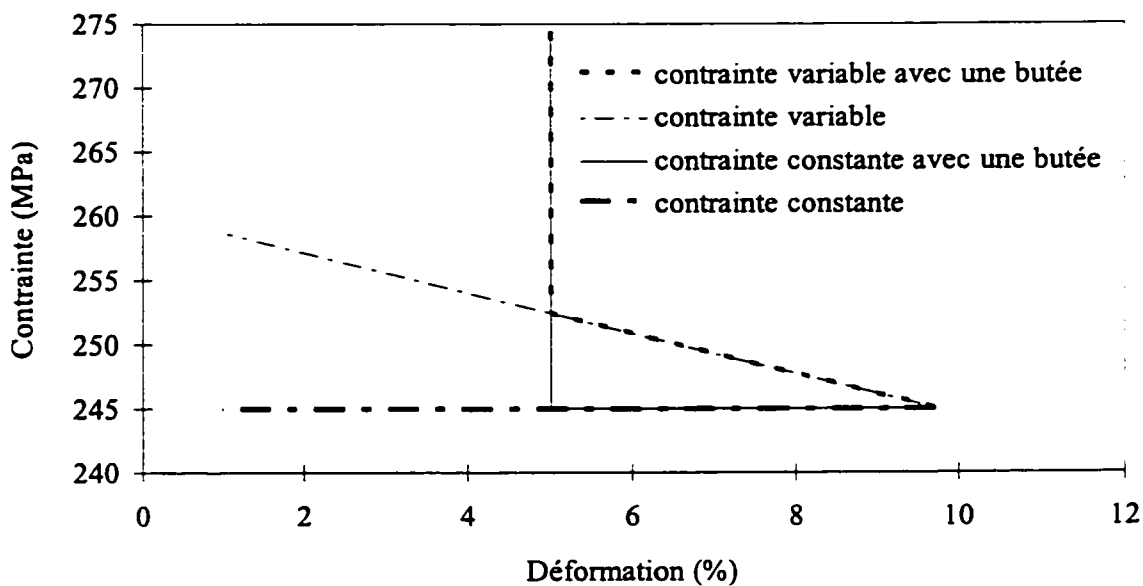


Figure 3.5 Contrainte calculée en fonction de la déformation.

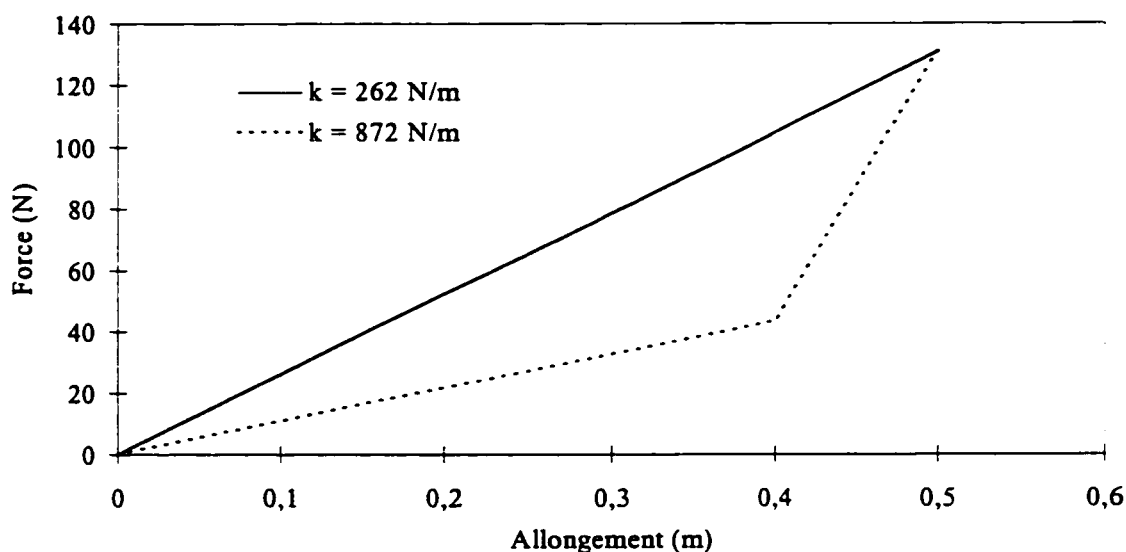


Figure 3.6 Courbes caractéristiques des contre-ressorts¹.

3.2 Influence de l'incrément de température

Les figures 3.7 et 3.8 présentent respectivement l'influence de l'incrément de température sur la précision de la courbe déformation-température calculée par le modèle pour une contrainte imposée constante avec et sans la présence d'une butée. L'étude des figures 3.7 et 3.8 permet de conclure que l'incrément de température n'a qu'une très faible influence sur le calcul de la déformation du modèle pour une contrainte constante. En effet, la seule erreur introduite par un incrément grossier de la température est que le début et la fin de la transformation de phase ne sont pas respectés, car la précision de l'incrément ne permet pas de passer exactement par les températures de transformation. De plus, la présence d'une butée n'influence pas le niveau de précision inhérent à la valeur de l'incrément de température.

¹ Pour les calculs présentés dans ce chapitre, les contre-ressorts ne sont sollicités que pour des déflexions de 0.4 m à 0.5 m.

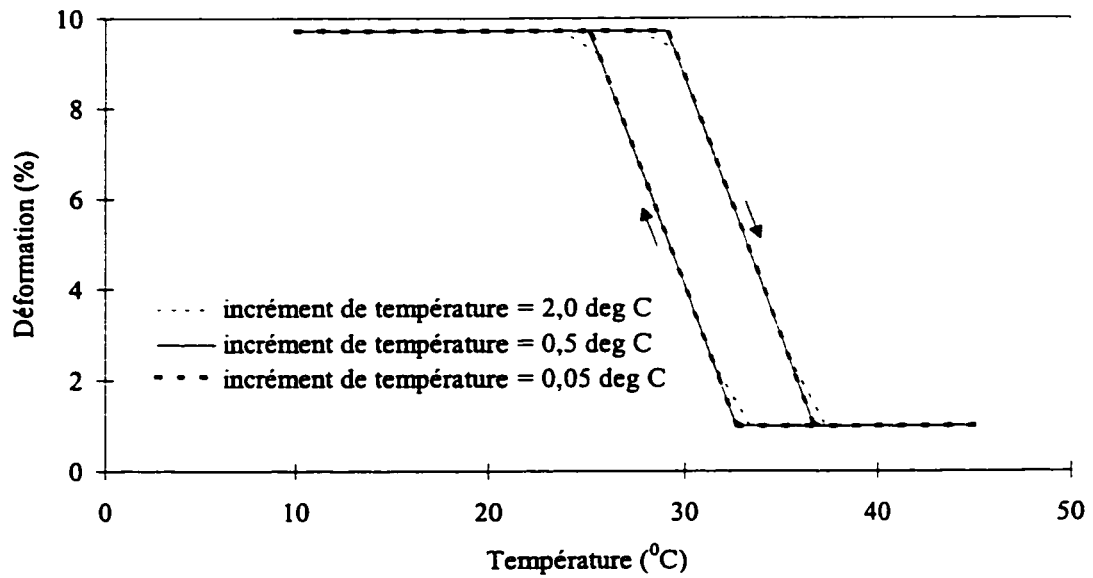


Figure 3.7 Influence de l'incrément de température sur la déformation (contrainte imposée constante).

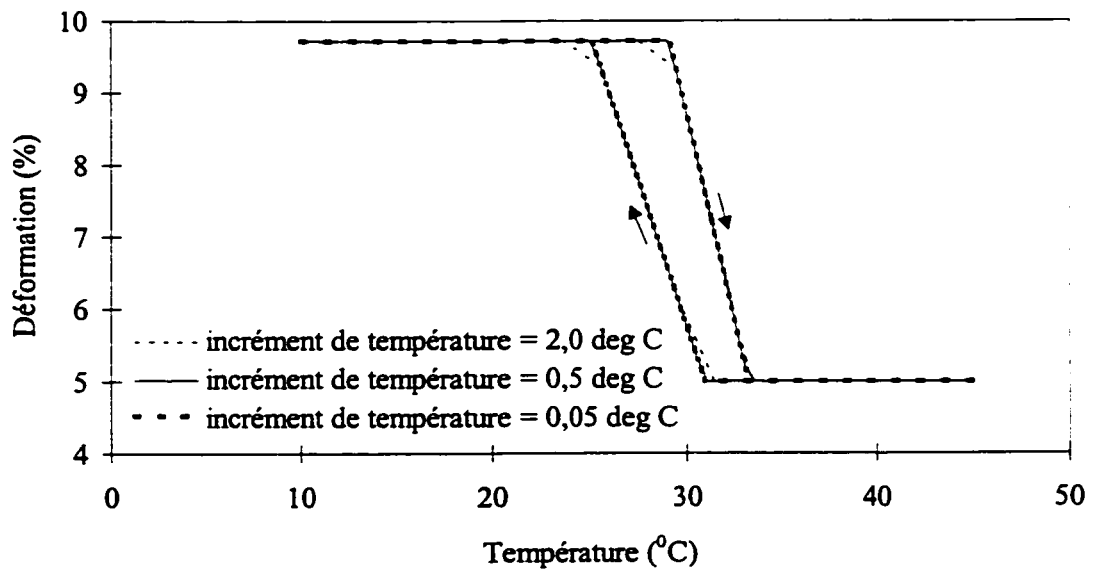


Figure 3.8 Influence de l'incrément de température sur la déformation (contrainte imposée constante avec une butée).

Les figures 3.9 et 3.10 présentent respectivement l'influence de l'incrément de température sur la précision de la courbe déformation-température calculée par le modèle pour une contrainte imposée variable (constante du contre-ressort de 872 N/m) avec et sans la présence d'une butée. Les figures 3.11 et 3.12 présentent ces mêmes courbes pour une constante du contre-ressort de 262 N/m. On remarque que, lorsque la contrainte imposée est variable, l'incrément de température devient critique. En effet, plus le saut de température est grand, moins la méthode de calcul est précise. Cette méthode consiste à utiliser la déformation du pas de calcul précédent pour calculer la force générée par le contre-ressort. Cette force est ensuite transformée en contrainte dans l'AMF et la déformation peut être interpolée sur les plans de chargement. La contrainte a donc toujours un "demi-pas" de retard sur la déformation. Si cet écart devient trop important, l'état thermomécanique de l'AMF calculé pour ce pas de chargement n'est plus cohérent avec celui calculé par le reste de la séquence de chargement. En conséquence, un nouveau plan de chargement est créé pour ce pas de calcul, mais qui ne correspond pas avec l'état thermomécanique réel de l'AMF pour les conditions de chargement imposées. On peut en effet remarquer sur les figures 3.9 à 3.12 que plus le pas est grand, plus la courbe est en "escalier". Ceci est particulièrement vrai lorsque le contre-ressort est plus rigide (figures 3.9 et 3.10), car la contrainte augmente rapidement pour de faibles déformations. Il devient alors nécessaire de diminuer l'incrément de température, bien que l'espace mémoire devienne vite insuffisant pour la visualisation des résultats. En effet, pour chaque diminution de moitié de l'incrément de la température, l'espace mémoire nécessaire est doublé et le système d'exploitation DOS ne met que 64K à notre disposition. Néanmoins, le logiciel retourne une réponse et on peut utiliser un autre logiciel, par exemple *EXCEL*, pour visualiser les résultats.

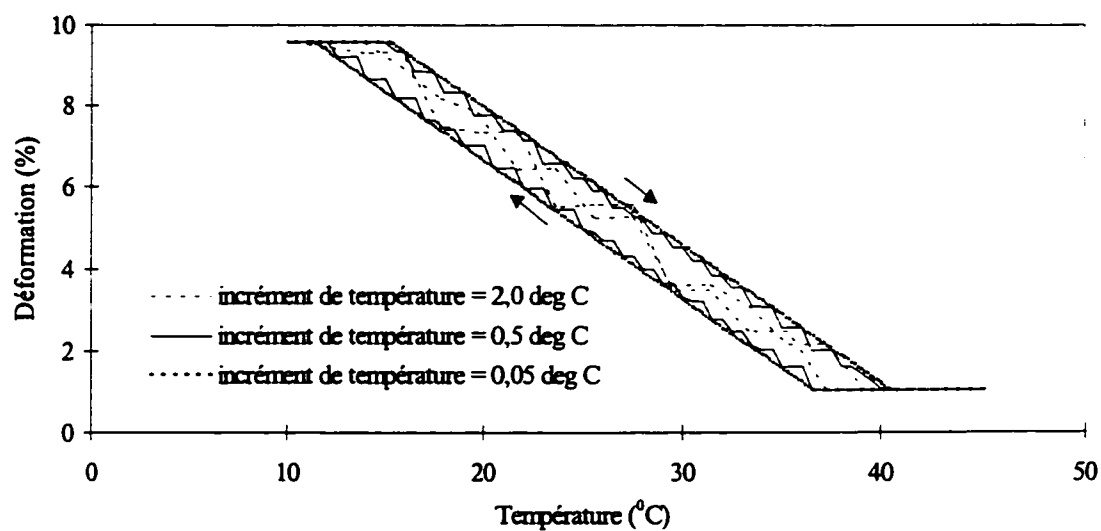


Figure 3.9 Influence de l'incrément de température sur la déformation (contrainte imposée variable, raideur du contre-ressort $k = 872\text{N/m}$).

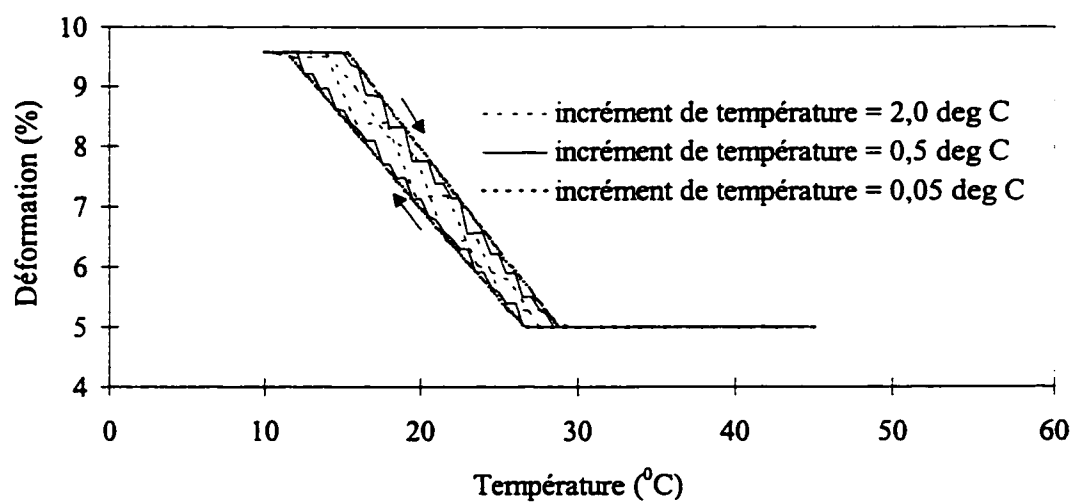


Figure 3.10 Influence de l'incrément de température sur la déformation (contrainte imposée variable avec butée, raideur du contre-ressort $k = 872\text{N/m}$).

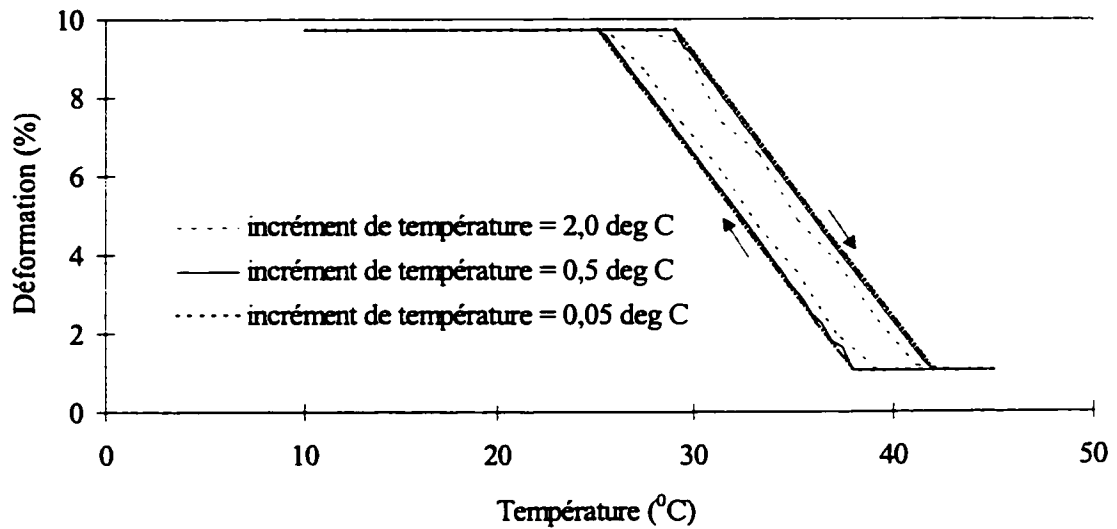


Figure 3.11 Influence de l'incrément de température sur la déformation (contrainte imposée variable, raideur du contre-ressort $k = 262\text{N/m}$).

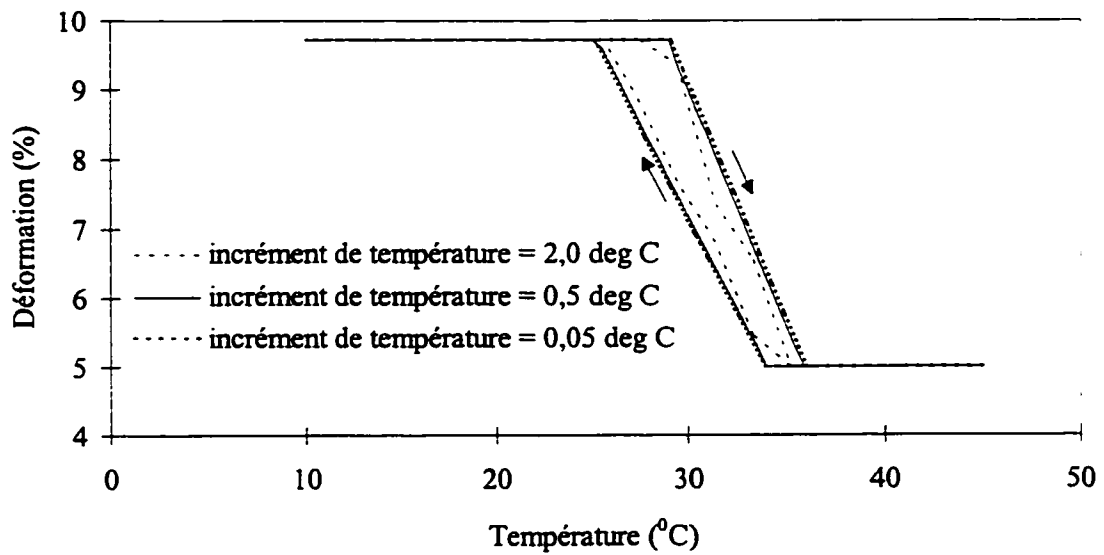


Figure 3.12 Influence de l'incrément de température sur la déformation (contrainte imposée variable avec butée, raideur du contre-ressort $k = 262\text{N/m}$).

Les figures 3.13 et 3.14 illustrent l'influence de l'incrément de température sur le calcul du temps pour le cas le plus critique, soit un échantillon couplé à un contre-ressort. On remarque que plus l'incrément est grossier, plus la progression du temps est discontinue. En effet, de fortes variations de température impliquent une alternance dans les plans de chargement. On ne se maintient pas sur un seul plan lors de la transformation, mais on progresse plutôt en escalier sur des plans avec et sans transformation, ce qui amplifie les erreurs de calcul imputables au modèle, car celui-ci calcule une transformation lorsqu'il n'y en a pas et vice versa. Cependant, on remarque que l'erreur sur le temps total de transformation est peu importante, ce qui amène à conclure que la précision sur l'incrément de température n'est pas aussi importante pour le calcul du temps que pour le calcul mécanique, car les erreurs s'auto-compensent.

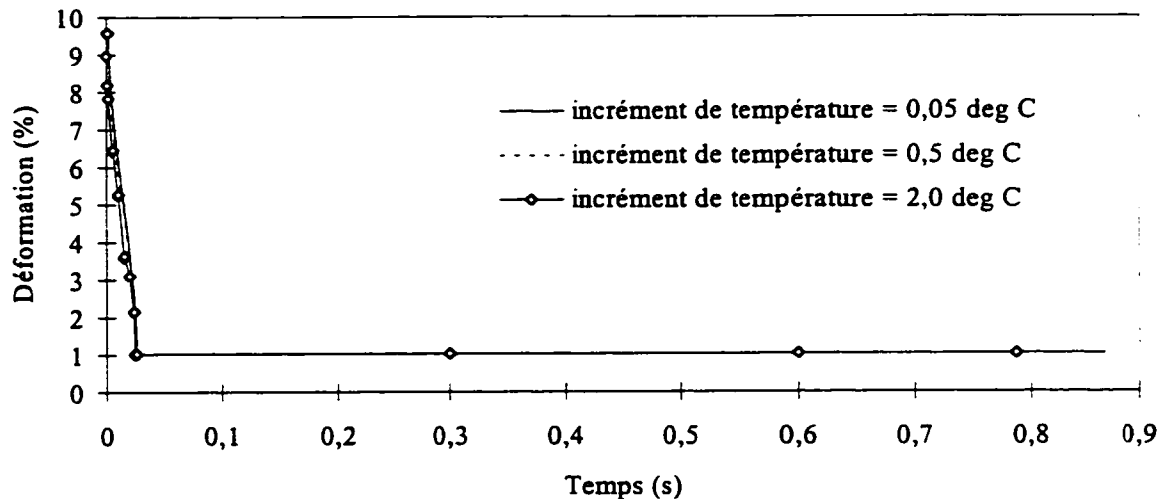


Figure 3.13 Influence de l'incrément de température sur le temps (contrainte imposée variable - cycle de chauffage).

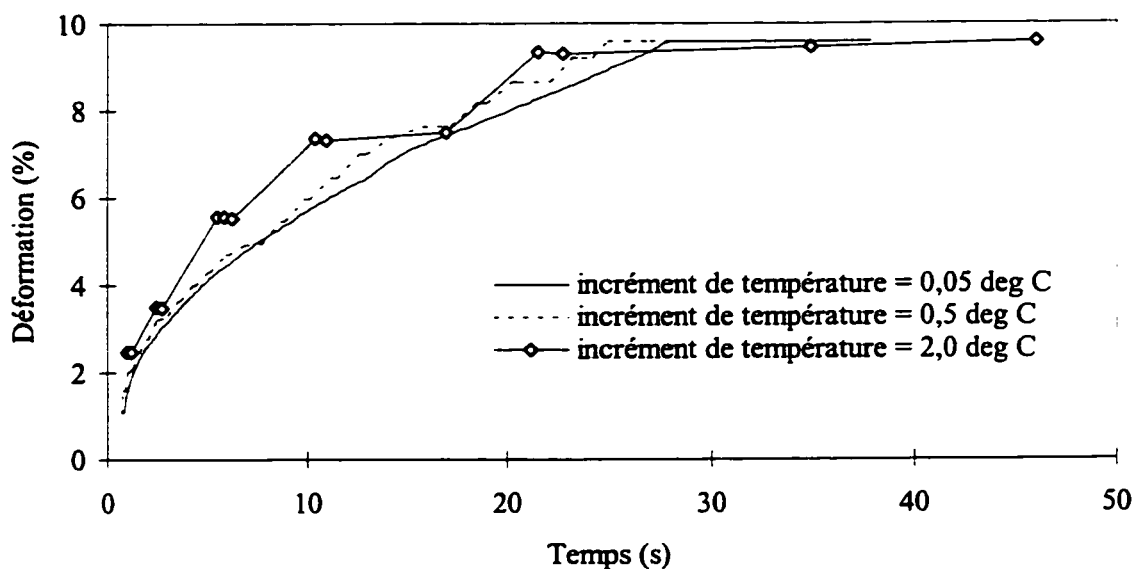


Figure 3.14 Influence de l'incrément de température sur le temps (contrainte imposée variable - cycle de refroidissement).

Pour pallier au problème du pas d'incrément de la température, il serait probablement préférable d'utiliser un pas ajustable, c'est-à-dire que le pas pourrait s'adapter à la rigidité du contre-ressort. Cette idée est cependant rejetée, car le temps de transformation est peu influencé par la précision sur l'incrément de température. En effet, dans le cas où l'on souhaite obtenir rapidement une approximation des temps de transformation de l'AMF, un pas d'incrément grossier en température est satisfaisant, tandis qu'un pas d'incrément plus précis obligerait l'utilisateur à traiter les données obtenues à l'aide d'un autre logiciel, ce qui peut devenir fastidieux. On peut donc conclure que lorsqu'on veut calculer les temps de transformation de l'AMF, un pas d'incrément grossier en température est adéquat. Si l'aspect mécanique du calcul nous intéresse plus particulièrement, il est préférable de raffiner le pas d'incrément, surtout s'il s'agit d'un cas où la contrainte a tendance à varier brusquement.

En revanche, quand on calcul la température à partir d'un temps d'échauffement fixé, les remarques précédentes ne s'appliquent pas. En effet, la température étant calculée à partir du temps, le pas d'incrément de celle-ci n'est pas constant. De plus, pour simplifier le code de calcul, l'incrément de temps est fixé à une seconde, sans possibilité pour l'utilisateur de modifier cet incrément. Cependant, différentes corrections sur le temps des échanges thermiques sont effectuées au cours du calcul afin de respecter précisément les températures de début et de fin de transformation. De cette façon on élimine l'erreur causée par un pas d'incrément trop grand de la température lorsque la contrainte est maintenue constante.

4 - ANALYSE QUANTITATIVE DES RÉSULTATS

En premier lieu, une validation partielle des modèles mécanique et thermique est présentée à partir d'une comparaison avec des données de la littérature, puis la démarche suivie pour effectuer une validation complète, c'est-à-dire la comparaison des calculs réalisés par le logiciel avec des résultats expérimentaux, est décrite. Finalement, les résultats de validation du modèle couplé sont exposés.

4.1 Comparaison des résultats calculés avec ceux de la littérature

À cette étape-ci, les modèles mécanique et thermique sont étudiés séparément afin de s'assurer du bon fonctionnement de chacun. Les résultats calculés par le modèle mécanique seront comparés aux courbes de déformation en fonction de la température obtenues par Patoor et al. [39], tandis que les courbes d'échauffement d'un fil cuivreux obtenues à l'IREQ [40] seront utilisées pour la comparaison avec les calculs effectués par le modèle thermique.

4.1.1 Modèle mécanique

Avant d'effectuer des essais en laboratoire, il est important d'évaluer sommairement le volet "calcul mécanique" du logiciel. En effet, il s'agit d'être en mesure de reproduire des courbes déformation-température tirées de la littérature. Les courbes de déformation en fonction de la température de Patoor, Eberhardt et Berveiller [39] ont été utilisées ici (échantillon monocristaux Cu -16 at.% Zn - 15 at.% Al). La figure 4.1 présente le résultat obtenu avec un incrément de température de 0.05°C. Les données concernant chacune de ces courbes [39] sont fournies aux tableaux 4.1 et 4.2.

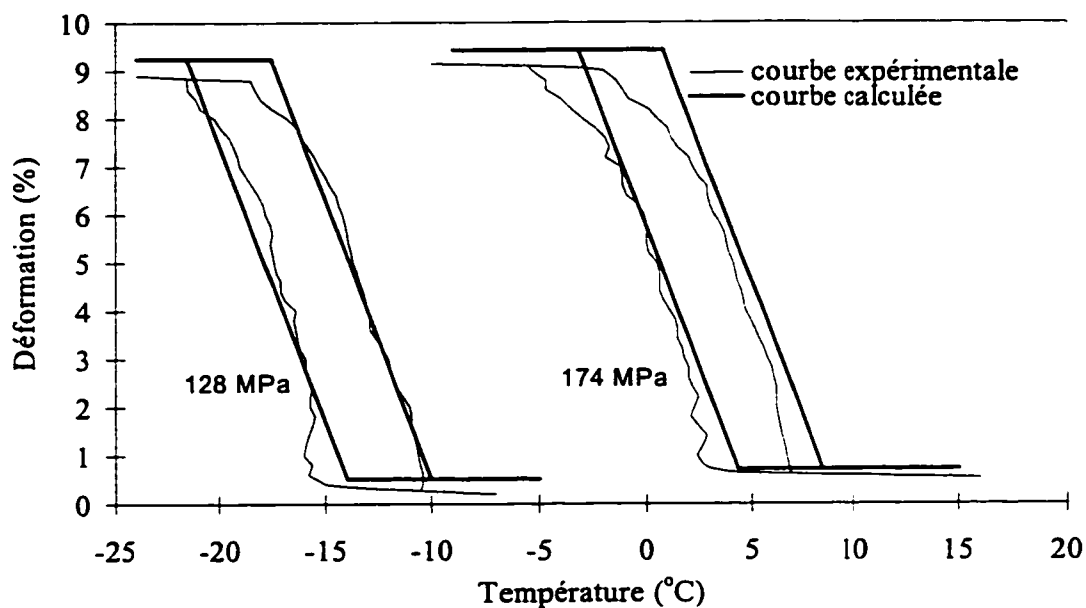


Figure 4.1 Comparaison de courbes calculées de la déformation en fonction de la température avec des résultats tirés de la littérature.

Tableau 4.1 Données pour la validation partielle - modèle mécanique (contrainte = 174 MPa). Matériaux : monocristaux Cu -16 at.% Zn - 15 at.% Al

Catégories	Données	Valeur
Géométrie	Longueur de l'éprouvette (l)	-
	Diamètre de l'éprouvette (d)	0.00234 m
Chargement thermomécanique	Contrainte appliquée (σ)	174 MPa
	Température ambiante (T_{AMB})	-5.0 °C
Matériau	Température M_s	-65.0 °C
	Température M_f	-74.0 °C
	Température A_s	-68.5 °C
	Température A_f	-61.0 °C
	Contrainte d'induction de mart. (σ_{M_s})	150.6 MPa
	Contrainte fin d'induction d'aust. (σ_{A_f})	140.56 MPa
	Pente $d\sigma/dT$	2.51 MPa/°C
	Déformation de transformation (ϵ_{trans})	8.8 %
	Module d'élasticité en austénite (E_1)	24600.0 MPa / m/m
	Module de transformation ($d\sigma/d\epsilon$)	213.92 MPa / m/m
	Module d'élasticité en martensite (E_2)	24600.0 MPa / m/m
Pente d'instabilité (P_{inst})	100.294 MPa / m/m	

Tableau 4.2 Données pour la validation partielle - modèle mécanique (contrainte = 128 MPa). Matériaux : monocristaux Cu -16 at.% Zn - 15 at.% Al

Catégories	Données	Valeur
Géométrie	Longueur de l'éprouvette (l)	-
	Diamètre de l'éprouvette (d)	0.00234 m
Chargement thermomécanique	Contrainte appliquée (σ)	128 MPa
	Température ambiante (T_{AMB})	-25.0 °C
Matériau	Température M_s	-65.0 °C
	Température M_f	-74.0 °C
	Température A_s	-68.5 °C
	Température A_f	-61.0 °C
	Contrainte d'induction de mart. (σ_{Mf})	100.4 MPa
	Contrainte fin d'induction d'aust. (σ_{Af})	90.36 MPa
	Pente $d\sigma/dT$	2.51 MPa/°C
	Déformation de transformation (ϵ_{trans})	8.8 %
	Module d'élasticité en austénite (E_1)	24600.0 MPa / m/m
	Module de transformation ($d\sigma/d\epsilon$)	213.92 MPa / m/m
	Module d'élasticité en martensite (E_2)	24600.0 MPa / m/m
Pente d'instabilité (P_{inst})	100.294 MPa / m/m	

La figure 4.1 permet de constater que le modèle respecte bien le comportement mécanique de l'alliage.

4.1.2 Modèle thermique

Il s'agit maintenant de vérifier les calculs d'échauffement des échantillons AMF. Les courbes utilisées proviennent d'un essai d'échauffement par effet Joule d'une tige en TiNi (composition inconnue) effectué à l'Institut de recherche d'Hydro-Québec [38]. Comme, aucune information n'était donnée sur la température d'équilibre atteinte dans la tige dans les conditions de l'essai, on négligera ici les pertes thermiques aux extrémités de l'échantillon. Le coefficient de correction a donc été considéré nul. La figure 4.2 présente la courbe de la température en fonction du temps quand la tige est chauffée par effet joule pour deux valeurs différentes du courant, soit 50A et 75A. Les données de l'essai pour chacune des courbes sont fournies aux tableaux 4.3 et 4.4.

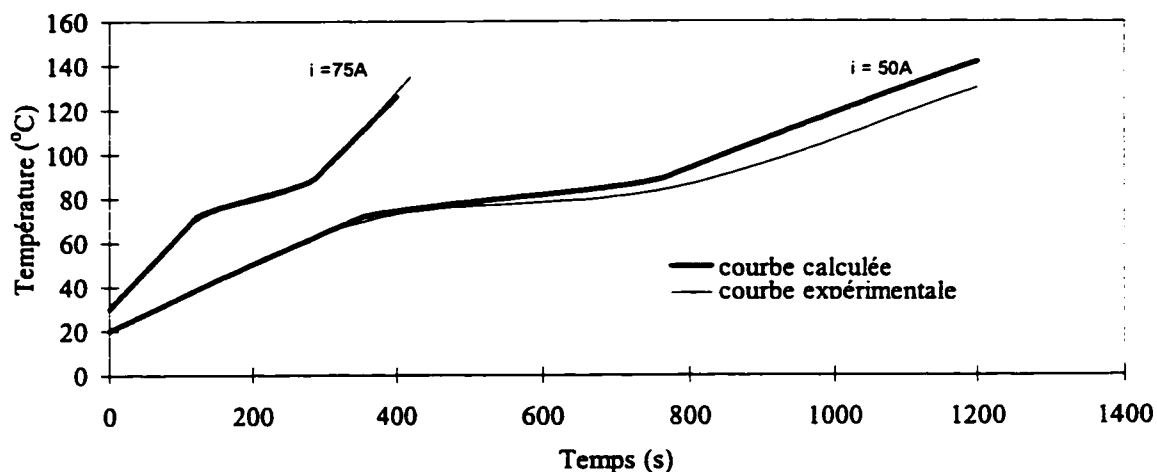


Figure 4.2 Comparaison de courbes calculées de la température en fonction du temps avec des résultats expérimentaux.

Tableau 4.3 Données pour la validation partielle - modèle thermique (courant = 75 A).
Matériaux : TiNi de composition inconnue.

Catégories	Données	Valeur	
Géométrie	Longueur de l'éprouvette (l)	0.3 m	
	Diamètre de l'éprouvette (d)	0.008 m	
Chargement thermomécanique	Contrainte appliquée (σ)	0.0 Mpa	
	Température ambiante (T_{AMB})	30.0 °C	
	Coefficient de convection (h)	3.5 W/m ² °C	
	Intensité du courant (i)	75 A	
Matériau	Température M_c	90 °C	
	Température M_r	72 °C	
	Température A_c	72 °C	
	Température A_r	90 °C	
	Coefficients d'approximation de la courbe de calorimétrie au chauffage (A, B, D)		-9200.0
			9200.0
			700.0
	Coefficients d'approximation de la courbe de calorimétrie au refroidissement (A, B, D)		-9200.0
			9200.0
		700.0	
Masse volumique (ρ)		6500 kg/m ³	
Chaleur spécifique (C_p)		700 J/kg°C	
Résistivité électrique (ρ_{elec})		0.000000725 Ωm	

Tableau 4.4 Données pour la validation partielle - modèle thermique (courant = 50 A).
Matériaux : TiNi de composition inconnue.

Catégories	Données	Valeur	
Géométrie	Longueur de l'éprouvette (l)	0.3 m	
	Diamètre de l'éprouvette (d)	0.008 m	
Chargement thermomécanique	Contrainte appliquée (σ)	0.0 MPa	
	Température ambiante (T_{AMB})	20.0 °C	
	Coefficient de convection (h)	3.5 W/m ² °C	
	Intensité du courant (i)	50 A	
Matériau	Température M_i	90 °C	
	Température M_r	72 °C	
	Température A_i	72 °C	
	Température A_r	90 °C	
	Coefficients d'approximation de la courbe de calorimétrie au chauffage (A, B, D)		-9200.0
			9200.0
			700.0
	Coefficients d'approximation de la courbe de calorimétrie au refroidissement (A, B, D)		-9200.0
			9200.0
		700.0	
Masse volumique (ρ)	6500 kg/m ³		
Chaleur spécifique (C_p)	700 J/kg°C		
Résistivité électrique (ρ_{elec})	0.000000725 Ωm		

La figure 4.2 permet de constater que le calcul du temps effectué par le modèle thermique est très satisfaisant pour un courant de 75A. Le résultat moins satisfaisant pour un courant de 50A pourrait être imputable aux pertes par conduction causées par l'augmentation du temps d'échauffement.

4.2 Comparaison des résultats du modèle couplé avec l'expérience

À cette étape-ci du développement du logiciel, seul le cas de l'actuateur à fil droit soumis à une contrainte constante et sans butée est abordé. Pour valider le modèle, on procède de la façon suivante : Premièrement, on effectue la caractérisation du matériau afin d'obtenir les données nécessaires aux calculs mécaniques et thermiques. Deuxièmement, on fait des essais d'échauffement par courant électrique à contrainte constante dans le but

d'obtenir les courbes contrainte-déformation-température en fonction du temps. Finalement, on compare les courbes expérimentales obtenues lors des essais avec les courbes calculées par le modèle. Les annexes I et II présentent respectivement des essais de validation pour des alliages cuivreux et des alliages de TiNi (ITI-001-30 et F21951B de l'TREQ) qui n'ont pas donné de résultats concluants. En ce qui a trait à cette section du rapport, les résultats d'une validation complète du modèle pour des fils de TiNi (lot 950315A) fournis par Shape Memory Applications Inc. sont présentés.

4.2.1 Caractérisation des échantillons

La première étape lors de la validation d'un modèle consiste à caractériser les échantillons utilisés afin de connaître tous les paramètres nécessaires à l'application du modèle. Généralement, deux essais différents sont nécessaires pour effectuer une caractérisation complète. Le premier est un essai de DSC (Differential Scanning Calorimetry), et le second un essai de traction. Cependant, comme il est mentionné à l'annexe II, pour obtenir de bonnes propriétés superélastiques avec un alliage de TiNi, il est nécessaire que ce dernier ne subisse qu'un revenu partiel après écrouissage afin de laisser des contraintes résiduelles dans le matériau. Un tel traitement ne permet pas l'utilisation de la DSC de façon directe. Il faudrait en effet faire subir un traitement de revenu au matériau pour le libérer de ces contraintes afin d'obtenir des résultats cohérents de DSC. Cependant, lors d'un tel traitement, les propriétés du matériau sont altérées et les températures de transformation ne sont plus les mêmes. Il serait éventuellement intéressant de vérifier, dans un scénario semblable, si l'évolution de la chaleur spécifique ainsi que l'hystérésis en température restent identique même si les températures de transformation sont modifiées. Dans un tel cas, en paramétrisant la courbe de chaleur spécifique en fonction de la température, il serait possible d'obtenir la courbe de DSC du matériau étudié même si ce dernier a dû subir un traitement d'écrouissage et de revenu partiel.

En ce qui nous concerne, ces travaux n'étant pas encore complétés, nous avons utilisé les données d'un essai de DSC effectué sur un autre alliage de TiNi. En effet, selon Airoidi, Rivolta et Turco [40], l'énergie de transformation des alliages de TiNi, pour une composition chimique similaire, dépend des températures de transformation selon une relation linéaire. Il est donc possible de déterminer les coefficients qui modélisent la courbe de la chaleur spécifique en fonction de la température en paramétrisant une courbe pour un alliage connu, puis de modifier le niveau d'énergie à l'aide de la relation donnée par Airoidi, Rivolta et Turco. À cette fin, il est nécessaire de fournir au modèle thermique deux nouveaux paramètres, soit les valeurs de l'hystérésis de température au chauffage et au refroidissement.

Essai de traction

Les essais de traction sur les fils de TiNi 950315A ont été effectués avec une machine Instron. Pour chaque essai, il faut obtenir une transformation complète. À partir d'une courbe superélastique complète, il est possible de déterminer le module d'élasticité en austénite E_1 , le module de transformation $d\sigma/d\varepsilon$, le module d'élasticité en martensite E_2 , la pente d'instabilité P_{inst} et la déformation de transformation ε_{trans} (figure 2.4). Un minimum de deux essais à des températures différentes et supérieures à A_f doivent être effectués pour obtenir le taux de variation de la contrainte en fonction de la température $d\sigma/dT$, la contrainte d'induction de martensite à la température du milieu ambiant σ_{Ms} et la contrainte de fin d'induction d'austénite à la température ambiante σ_{Af} (figure 2.5). De plus, afin que le logiciel indique s'il y a début de déformation plastique, il faut déterminer aussi la contrainte de plasticité du matériau S_y . Cependant, l'exactitude de la valeur de S_y dépend du nombre d'essais effectués. Ce dernier type d'essai étant destructif et sachant que le calcul de la plasticité ne représente pas une priorité, on peut se limiter à un seul essai ou simplement supposer une valeur de S_y assez élevée pour ne pas entraver le reste des calculs.

La première étape consiste donc à obtenir une courbe de traction stable pour l'échantillon. L'annexe III donne le cycle caractéristique de stabilisation pour l'échantillon fourni par le fabricant, tandis que la figure 4.3 présente la courbe superélastique stabilisée obtenue avec l'appareil de traction Instron à l'IREQ.

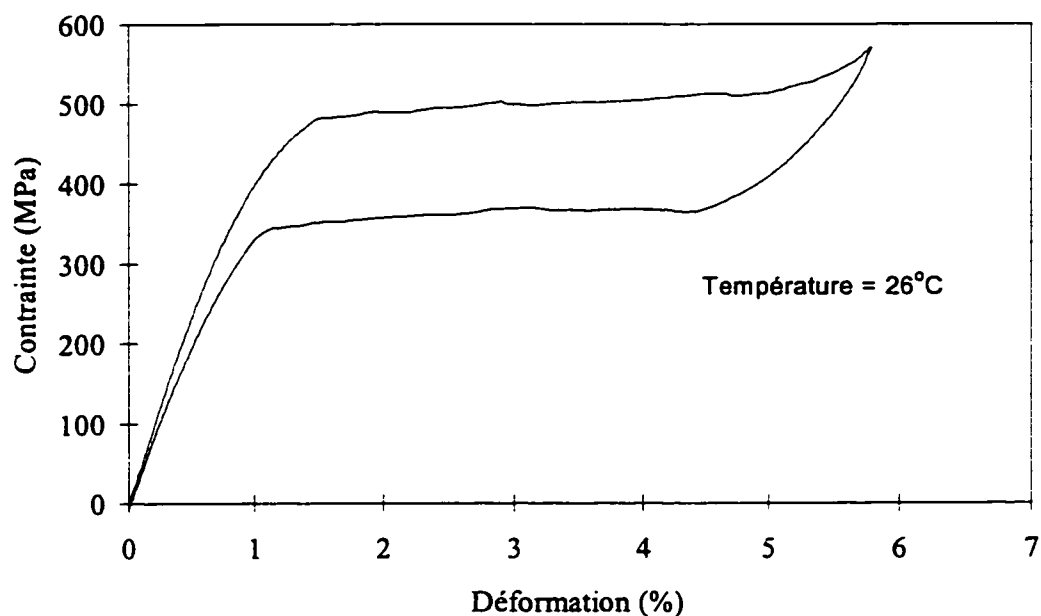


Figure 4.3 Courbe superélastique de l'échantillon de TiNi traité.

On remarque que la courbe possède toutes les caractéristiques recherchées pour la validation du logiciel. Il faut donc obtenir des courbes de traction à différentes températures de façon à construire le domaine superélastique dans lequel le logiciel devra opérer. La figure 4.4 présente les courbes obtenues pour des températures successives de 23,9°C, 30,0°C, 49,2°C et 60,2°C. Ces courbes vont permettre de déterminer les différentes propriétés de l'alliage qui sont nécessaires pour construire le domaine superélastique utilisé par le modèle.

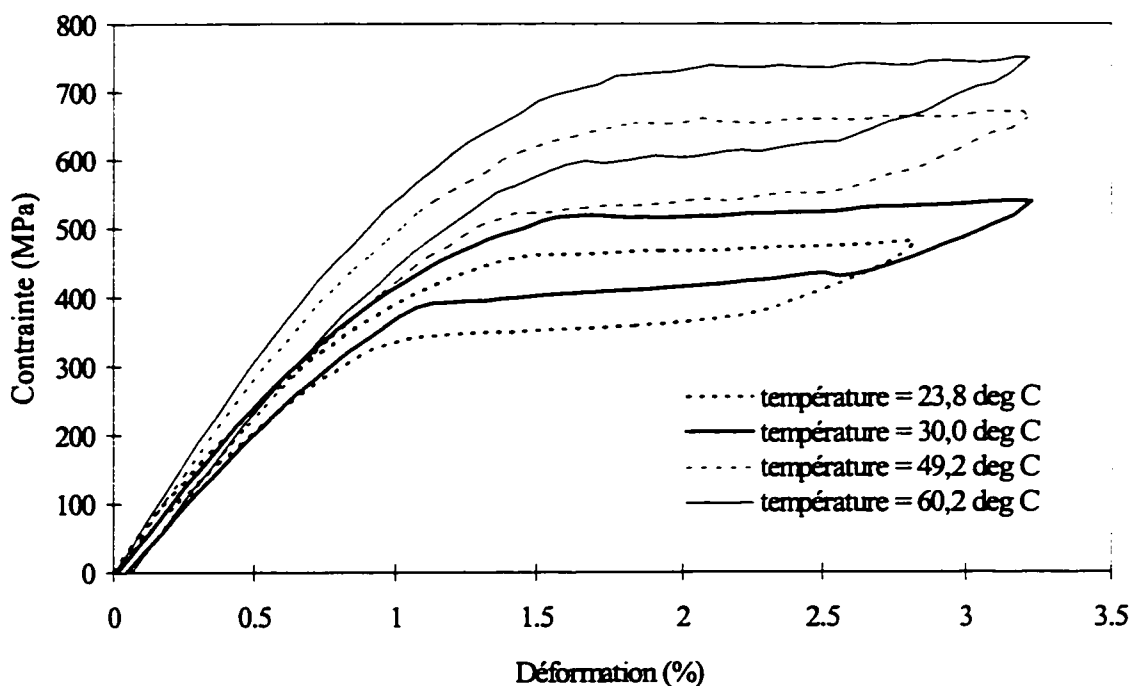


Figure 4.4 Courbes de traction à différentes températures du TiNi traité.

Le tableau 4.5 présente les données utilisées pour la validation du modèle : les paramètres tirés des courbes superélastiques présentées ci-dessus, les données de DSC tirées de la littérature, auxquels il faut ajouter les paramètres de l'expérience et les informations sur le matériau, c'est-à-dire l'intensité du courant (i), le coefficient de convection (h), la distance entre les mors (qui est la longueur active de l'échantillon), la température ambiante (T_{AMB}), la valeur du poids qui induit la contrainte dans l'échantillon, le diamètre de l'échantillon (d), la résistivité électrique du matériau (ρ_{elec}) ainsi que sa masse volumique (ρ) et sa chaleur spécifique (C_p).

Tableau 4.5 Données pour la validation du modèle. Matériaux : TiNi 950315A de la compagnie SMA inc.

Catégories	Données	Valeur	
Géométrie	Longueur de l'éprouvette (l)	0.16 m	
	Diamètre de l'éprouvette (d)	0.0004 m	
Chargement thermomécanique	Contrainte appliquée (σ)	560.4 MPa	
	Température ambiante (T_{AMB})	23.8 °C	
	Coefficient de convection (h)	6 W/m ² °C	
	Intensité du courant (i)	0.85 A	
	Facteur de correction (Q)	0.0199 W/°C	
Matériau	Température M_s	-42.84 °C	
	Température M_f	-47.54 °C	
	Température A_s	-28.73 °C	
	Température A_f	-24.02 °C	
	Coefficients d'approximation de la courbe de calorimétrie au chauffage (A, B, D)		-9200.0
			9200.0
			0.0
	Coefficients d'approximation de la courbe de calorimétrie au refroidissement (A, B, D)		-9200.0
			9200.0
			0.0
	Contrainte d'induction de mart. (σ_{Ms})	483.15 MPa	
	Contrainte fin d'induction d'aust. (σ_{Af})	346.73 MPa	
	Pente $d\sigma/dT$	7.25 MPa / °C	
	Déformation de transformation (ϵ_{trans})	4.585 %	
	Module d'élasticité en Austénite (E_s)	41878.3 MPa / m/m	
	Module de transformation ($d\sigma/d\epsilon$)	743.73 MPa / m/m	
	Module d'élasticité en martensite (E_m)	41878.3 MPa / m/m	
Pente d'instabilité (P_{inst})	-2399.62 MPa / m/m		
Masse volumique (ρ)	6450 kg/m ³		
Chaleur spécifique (Cp)	838.0 J/kg°C		
Résistivité électrique (ρ_{elec})	0.0000009 Ohm*m		

L'étape suivante consiste à mesurer la déformation en fonction du temps pour l'actuateur à fil droit à l'aide de la chambre de fatigue électrique et thermique à contrainte constante, afin d'obtenir les données nécessaires à la validation du modèle couplé.

4.2.2 Essai de validation

Comme il a déjà été décrit au chapitre 2, chaque essai consiste à cycler thermiquement une ou plusieurs fois un échantillon appliquée à l'aide d'un courant électrique auquel une charge constante est. La charge appliquée sert à induire une contrainte supérieur à σ_{Mf} dans l'échantillon afin que celui-ci soit martensitique à la température de l'essai. Les essais sont effectués dans la chambre de fatigue électrique et thermique à contrainte constante développée à l'Institut de Recherche d'Hydro-Québec. Le tableau 4.6¹ et la figure 4.5¹ présentent respectivement les données techniques et un schéma de l'appareil.

Tableau 4.6 Paramètres de la chambre de fatigue de l'IREQ.

Description	Plage
Géométrie des échantillons	fil ou ressort
Application de la charge	fil : tension ressort : compression
Nombre d'échantillon	3
Température	-184°C à +315°C
Courant	0 A à 100 A
Charge constante	0 kg à 100 kg
Nombre de points de mesure de tension	minimum de 3
Nombre de points de mesure de température	9 (3/échantillon) minimum
Amplitude de déplacement mesurable	±15 mm
Taux de chauffage maximum	0,83°C/sec
Précision du contrôleur PID	0,02°C
Volume interne	0,0198 m ³
Réservoir d'azote liquide	36,4 kg
Vitesse d'acquisition de données	300/sec

¹ Ces figures et tableaux ont été fournis par l'Institut de Recherche d'Hydro-Québec.

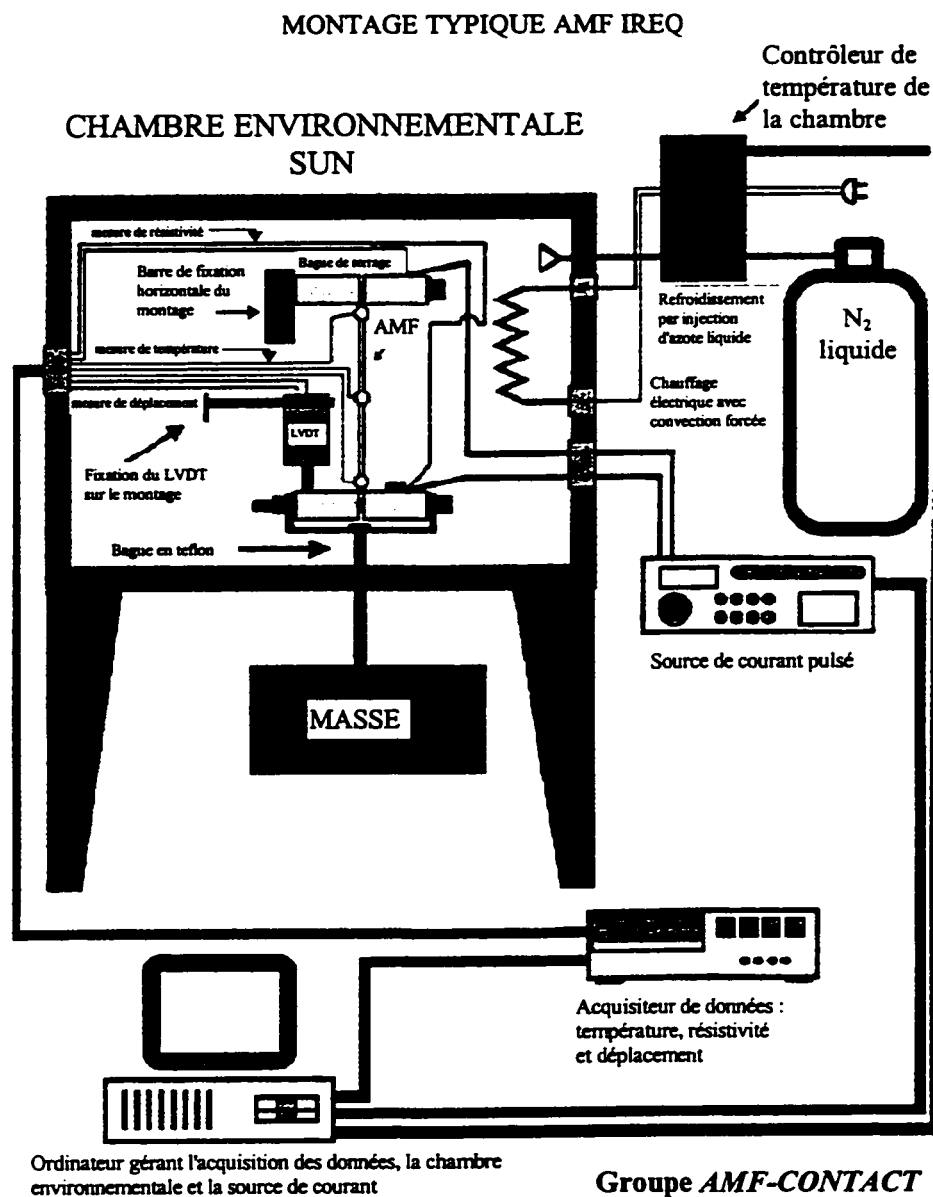


Figure 4.5 Schéma de la chambre de fatigue de l'IREQ.

La chambre (figures 4.5 et 4.6) possède les dimensions approximatives d'un four à micro-ondes. L'échauffement des échantillons peut se faire soit par courant électrique ou par convection forcée à partir de la température ambiante. Le refroidissement peut être naturel, ou forcé à l'aide d'une source d'azote liquide. Tous les accessoires sont fixés sur

une structure interne autoportante qui s'insère dans la chambre. Cette structure est fabriquée en Invar, ce qui limite les erreurs causées par la dilatation thermique des supports.

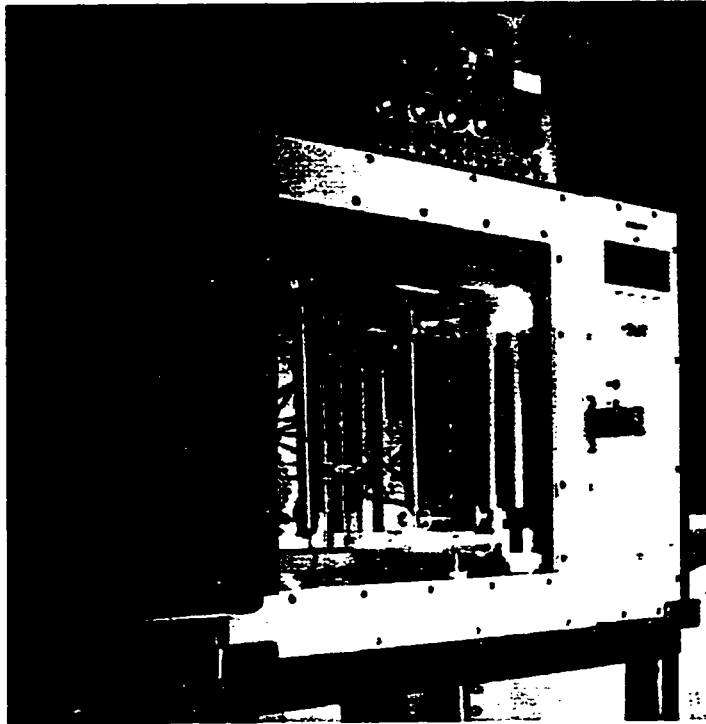


Figure 4.6 Photographie de la chambre de fatigue de l'IREQ.

Les échantillons sont coincés entre les faces de deux cylindres insérés sur un boulon (figures 4.5 et 4.7a). Ce type de fixation permet d'isoler électriquement les mors du montage en insérant un tube entre les cylindres et le boulon ainsi que deux rondelles à l'extrémité des cylindres. Un LVDT fixé sur les mors permet de lire l'allongement de l'échantillon pendant les essais.

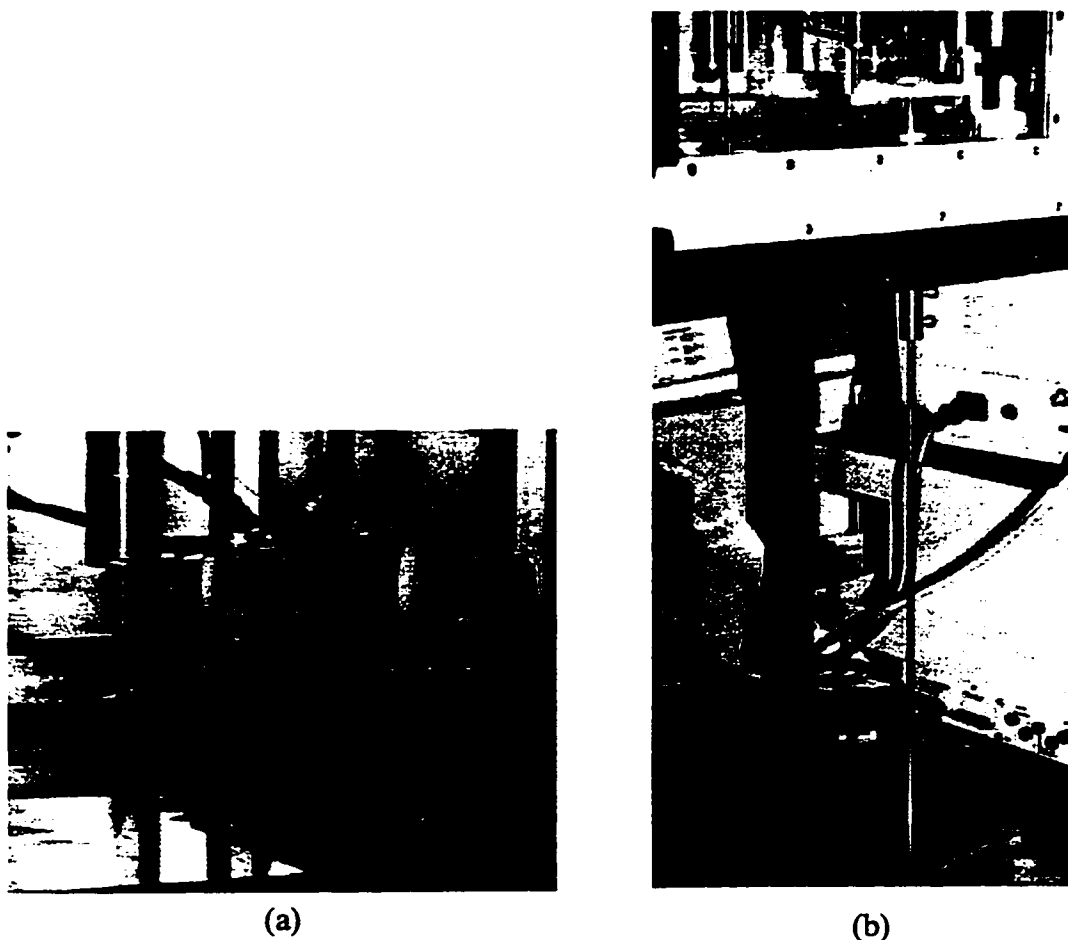


Figure 4.7 Fixation de l'échantillon dans la chambre de fatigue de l'IREQ et support de la charge.

La contrainte est appliquée en déposant un jeu de poids sur un plateau fixé sur le mors inférieur de l'échantillon (figures 4.5 et 4.7b). La chambre est contrôlée par un programme de gestion créé à l'aide du logiciel Vee-Test. Tel qu'illustré à la figure 4.8, ce programme permet d'obtenir simultanément les informations suivantes :

- (1) température des thermocouples en fonction du temps;
- (2) déplacement du LVDT en fonction de la température;
- (3) tension aux bornes du fil en fonction du temps;
- (4) courant en fonction du temps.

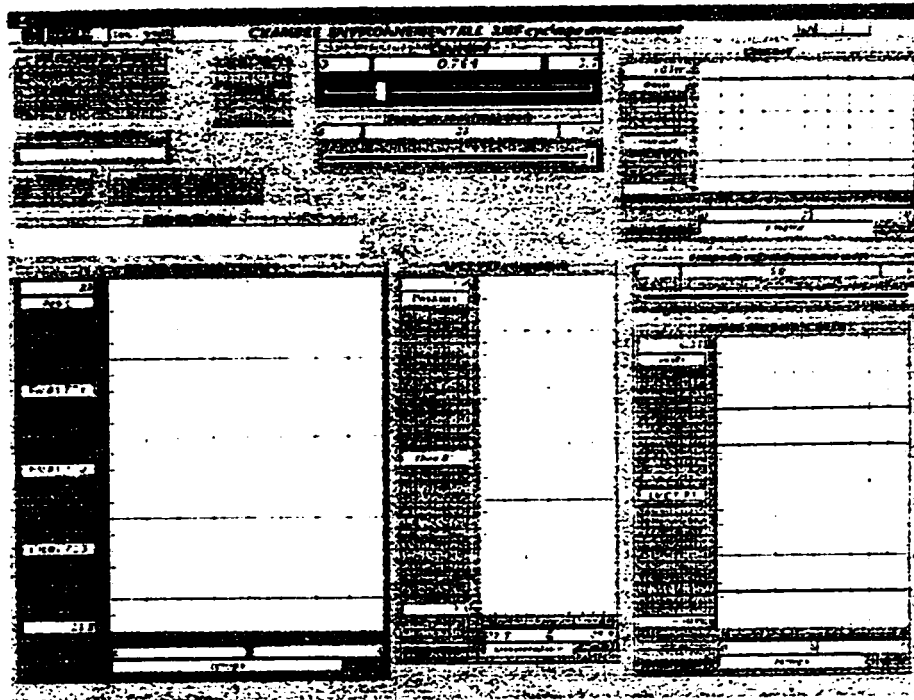


Figure 4.8 Interface de sortie du système d'acquisition de la chambre de fatigue de l'IREQ.

4.2.3 Analyse des résultats

Deux modes d'échauffement sont utilisés, soit l'échauffement par courant électrique et l'échauffement par convection. En effet, l'inertie thermique de la colle des thermocouples étant trop grande, la lecture de la température lors de l'échauffement par courant électrique est erronée. Le mode d'échauffement de l'échantillon ne modifiant pas son comportement *température-déformation*, il est possible de chauffer lentement l'échantillon par une élévation de la température dans la chambre environnementale et ainsi contourner le problème causé par l'inertie thermique de la colle des thermocouples.

Échauffement par courant électrique

La figure 4.9 présente une comparaison entre la courbe expérimentale déformation-temps et celle calculée avec et sans le coefficient de correction. La courbe a été obtenue

pour une contrainte de 560,4 MPa, ce qui correspond à une contrainte légèrement supérieure à σ_{MP}

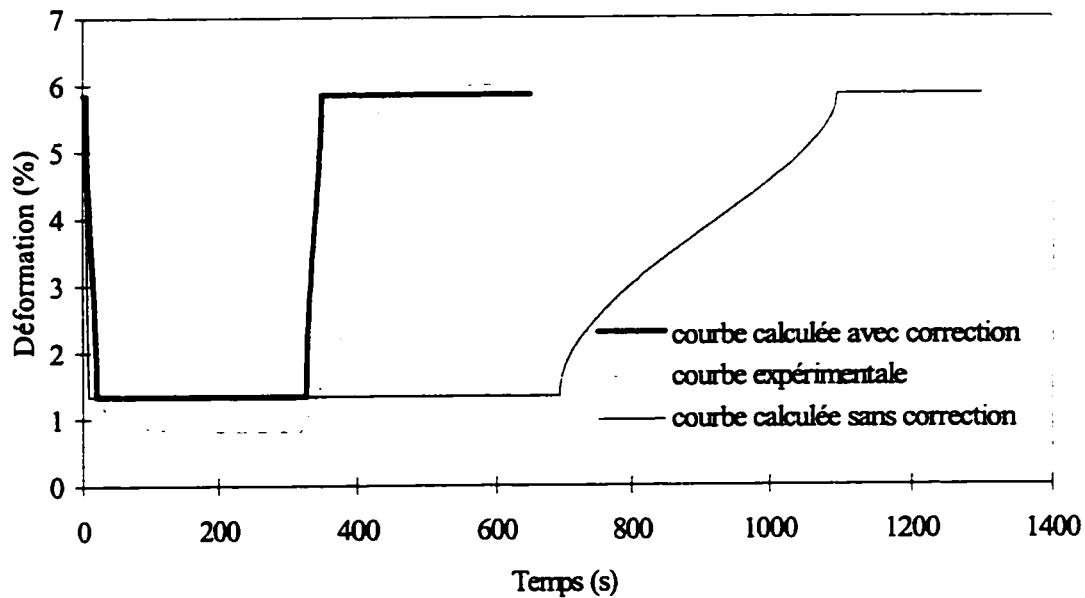


Figure 4.9 Déformation en fonction du temps (contrainte constante de 560,4 MPa).

On s'aperçoit que l'utilisation d'un coefficient de correction est justifiée. En effet, le modèle thermique sans coefficient de correction sous-estime les pertes, ce qui se traduit sur le graphique par une sous-estimation du temps de transformation au chauffage et une surestimation du temps de transformation au refroidissement. Ceci est confirmé par le fait que le refroidissement débute beaucoup plus tard pour le modèle sans correction, ce qui veut dire que pour un même temps de chauffage, la température atteinte par le modèle sans correction est beaucoup plus élevée.

La différence entre le temps calculé par le modèle avec un coefficient de correction et le temps de l'essai peut s'expliquer par le fait que l'on suppose que les mors qui tiennent le fil restent à la température ambiante T_{∞} . Ceci tend en effet à surestimer les pertes évaluées par le coefficient de correction. Bien que le résultat obtenu soit satisfaisant, il

reste cependant deux points à discuter : la différence entre les déformations de transformation du modèle et de l'essai, et la validité du coefficient de correction.

La différence entre les déformations de transformation peut s'expliquer de deux façons. Tout d'abord, les simplifications dans la construction du modèle affectent la précision du calcul et, comme nous l'avons vu lors de la validation partielle, il faut s'attendre à une légère différence entre les deux déformations de transformation. De plus, le modèle utilise une méthode de calcul des plans du domaine superélastique qui considère que la déformation de transformation est constante. Dans ce cas, si la pente du plan de martensite élastique n'est pas égale à celle de l'austénite élastique, il se créera des discontinuités entre les plans. Si l'on désire éliminer ces discontinuités, il faut supposer que le module d'Young de la martensite est égal à celui de l'austénite. Ceci a pour effet de surestimer ou de sous-estimer la déformation de transformation selon que le module réel de la martensite est plus élevé ou moins élevée que celui de l'austénite. Dans notre cas, en observant la figure 4.3, on s'aperçoit que le module d'Young de la martensite est moins élevée que celui de l'austénite, ce qui explique que la déformation de transformation calculée soit sous-estimée.

Afin de vérifier la validité du coefficient de correction, des essais à différents niveaux de contrainte ont été effectués. Les figures 4.10 et 4.11 présentent respectivement les courbes expérimentales et calculées de la déformation en fonction du temps à différents niveaux de contraintes. En comparant ces deux figures, nous sommes en mesure de vérifier si l'évolution du modèle en fonction de la contrainte correspond à l'expérience.

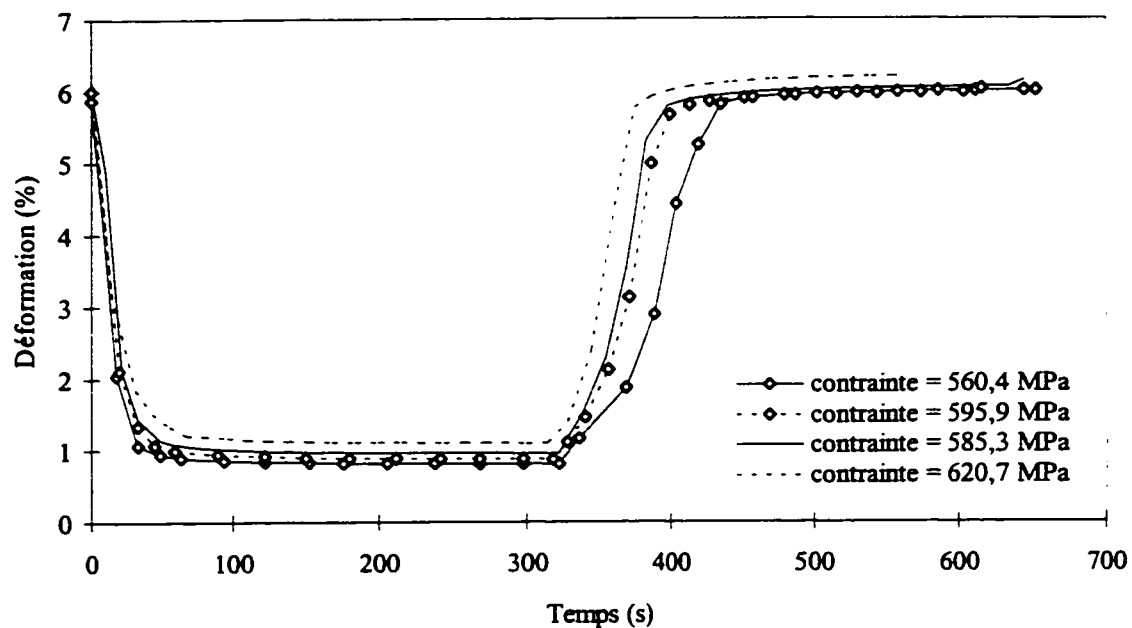


Figure 4.10 Déformation expérimentale en fonction du temps pour différents niveaux de contrainte.

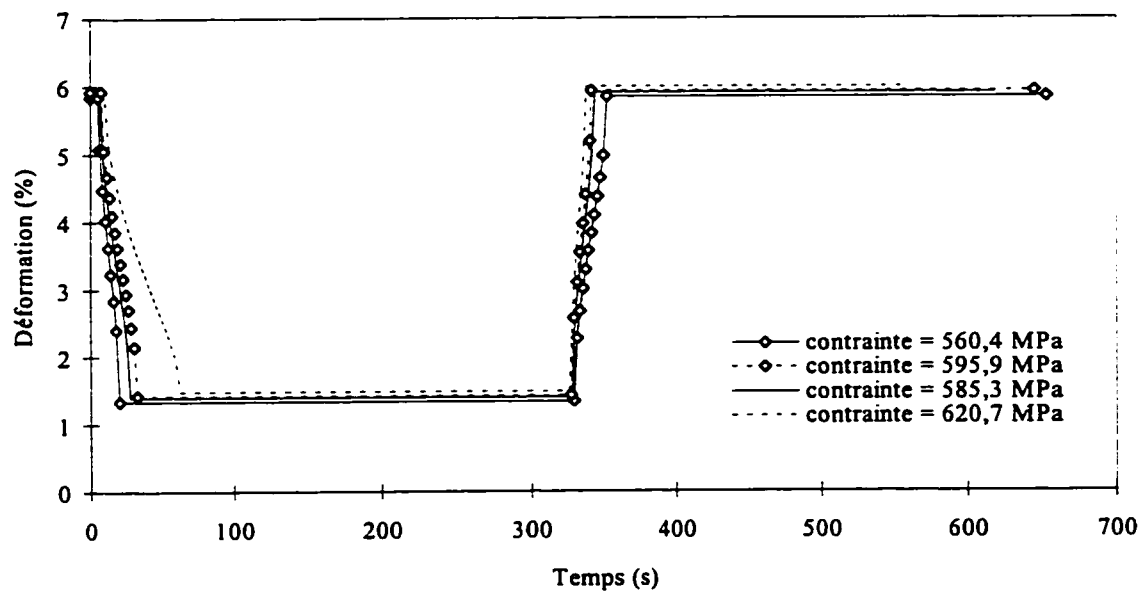


Figure 4.11 Déformation calculée en fonction du temps pour différents niveaux de contrainte.

En comparant les figures 4.10 et 4.11, on remarque que le modèle suit une évolution similaire aux essais, c'est-à-dire que plus la contrainte augmente, plus le temps nécessaire à la transformation vers l'austénite lors du chauffage est élevé et plus celui requis pour la transformation martensitique au refroidissement est court. Ceci est normal car plus la température de l'échantillon est élevée, plus les pertes sont importantes. De plus, l'écart relatif entre les courbes à différentes contraintes et celle à 560,4 MPa est le même pour le modèle que pour les essais. Il faut également mentionner que malgré le fait qu'au chauffage la courbe pour la contrainte de 620,7 MPa semble s'écarter anormalement des autres, on remarque, en se référant à la figure 4.12, que le temps total que prend la transformation pour s'accomplir est sensiblement le même pour le modèle et pour l'essai.

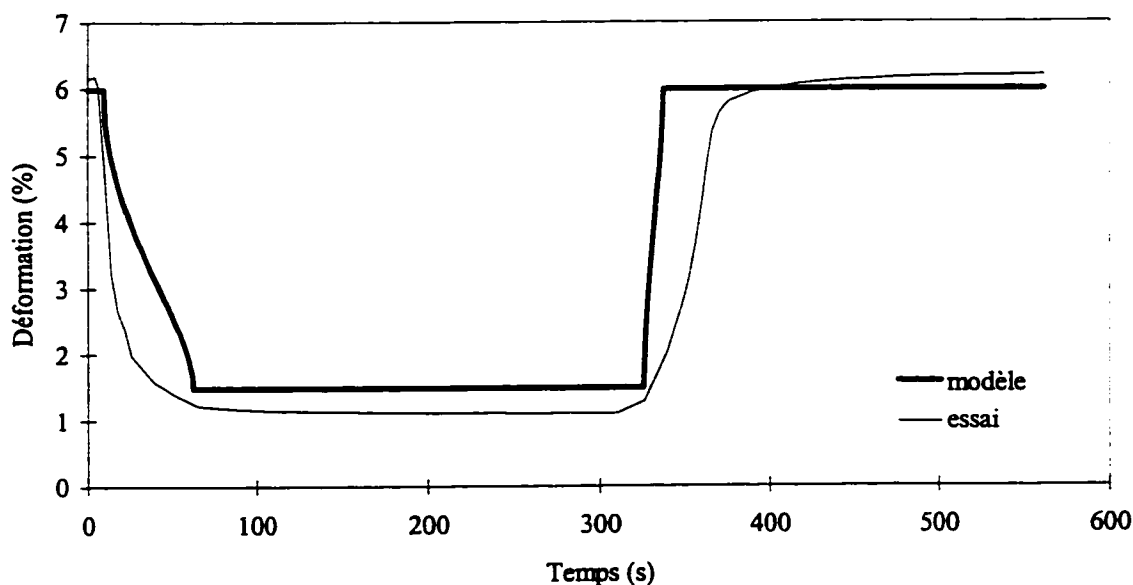


Figure 4.12 Déformation en fonction du temps (contrainte constante de 620,7 MPa).

La différence de parcours entre les deux courbes s'explique par le fait que plus la contrainte est élevée, plus la transformation a lieu à une température élevée, et comme

l'indique la figure 4.4, plus la température de transformation est élevée et plus la transition entre la transformation et les états élastiques est graduelle. Ainsi, au chauffage par exemple, la transformation s'effectuera rapidement au début, mais plus lentement au fur et à mesure qu'elle progressera, ce que le modèle n'est pas en mesure de reproduire. En effet, dans le modèle bilinéaire, la transition entre la transformation et les états élastiques est toujours ponctuelle et le temps de la transformation est toujours calculé selon la courbe théorique de l'évolution de la chaleur spécifique qui est supposée symétrique par rapport à la température médiane de la transformation.

La figure 4.13 présente les courbes calculées de la température et de la déformation en fonction du temps. L'étude de cette figure permet de vérifier que l'évolution de la déformation correspond avec celle de la température, c'est-à-dire que la déformation se produit seulement lors de la transformation de phase. Cette dernière peut être identifiée sur la figure 4.14 par la présence d'un plateau lors de l'échauffement et du refroidissement, car la transformation de phase est un phénomène endothermique au chauffage et exothermique au refroidissement.

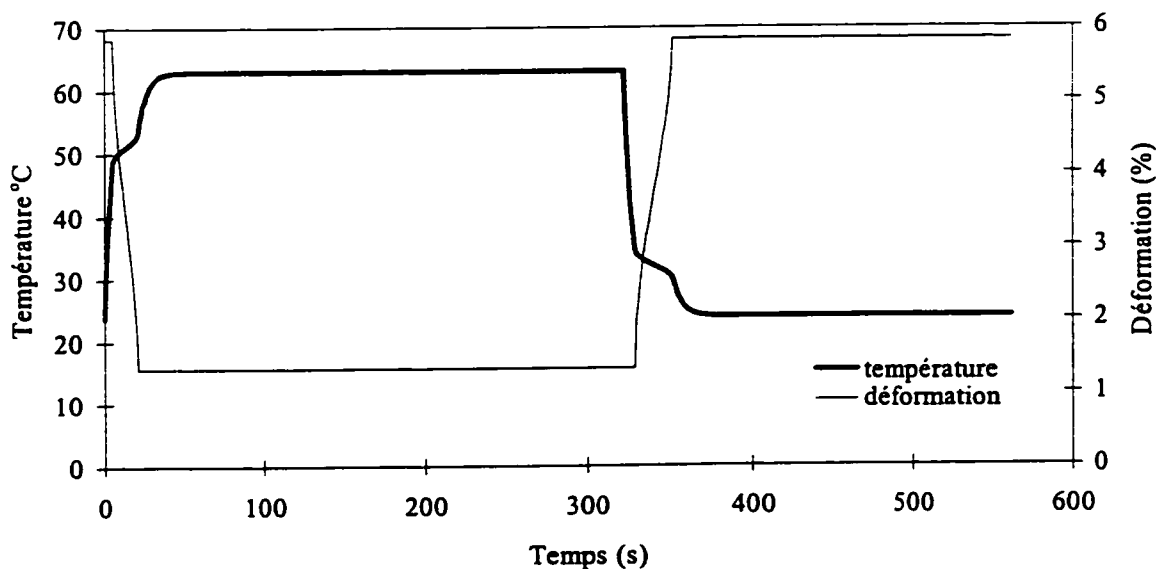


Figure 4.13 Température et déformation en fonction du temps.

Échauffement par convection

Pour valider le module de calcul mécanique, nous avons recours à l'échauffement de l'échantillon par convection, c'est à dire que l'échauffement est produit par une augmentation progressive de la température de l'enceinte thermique. Cette approche est utilisée pour un échantillon très fin car l'inertie thermique de la colle liant les thermocouples à l'échantillon empêche ces derniers de lire la température réelle de l'échantillon à un instant donné. La figure 4.14 présente le cycle superthermique obtenu pour les échantillons de TiNi dans la chambre environnementale. Cette figure confirme les résultats obtenus lors de la validation partielle et démontre que le modèle reproduit avec exactitude le comportement thermomécanique de l'échantillon.

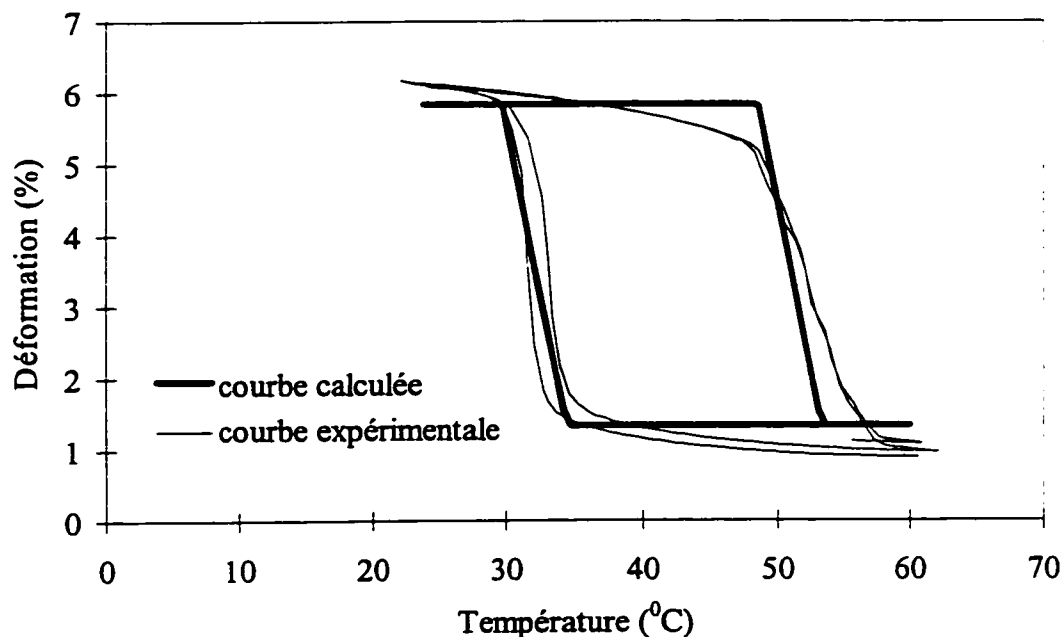


Figure 4.14 Déformation en fonction de la température pour la validation du modèle.

CONCLUSION

Le modèle de comportement bilinéaire des AMF présenté dans ce document s'avère être un modèle simple et rapide qui permet de prévoir les temps de réaction d'un actuateur AMF à fil droit avec une bonne précision dans le cas où ce dernier est soumis à une contrainte constante. Ces résultats permettent de supposer que le modèle est également fiable dans le cas où la contrainte est variable (contre-ressort et/ou butée). Cependant, cette approche comporte certaines limites. En ce qui concerne le module mécanique, le module d'Young de la martensite doit être supposé identique à celui de l'austénite, sinon il se crée une discontinuité entre les plans du domaine superélastique. Cette limitation a pour conséquence de sous-estimer la déformation de transformation. Dans le cas du modèle thermique, il n'est pas encore possible de calculer des sous-cycles lorsque la contrainte est variable. De plus, les temps de transformation de phase sont sous-estimés lors du refroidissement. Cette dernière limite est principalement imputable au fait que le coefficient de correction Q utilisé pour tenir compte des pertes par conduction ainsi que le coefficient de convection h sont considérés constants.

À la lumière des résultats présentés ici, on peut affirmer qu'il serait avantageux que le calcul du temps ne soit plus indépendant du calcul mécanique de façon à éliminer les limites inhérentes au modèle thermique. De plus, une telle modification faciliterait l'implantation d'un éventuel module de calcul éléments finis qui permettrait de traiter des cas de géométrie plus complexes, tout en tenant compte de la conduction lors des calculs thermiques.

RÉFÉRENCES

- [1] PATOOR E. et BERVEILLER M., Les alliages a memoire de forme, Hermès, Paris, 1990.
- [2] FUNAKUBO H., Shape memory alloys, Gordon & Breach, Amsterdam, 1987
- [3] HAWKINS L.E., VICK B. et ROGERS C.A., An Investigation of the Transient Thermal Response of a Shape Memory Alloy Composite Beam, AIAA-90-0941-CP, pp. 1972-78, 1990.
- [4] LIANG C. et ROGERS C.A., Behaviour of Shape Memory Alloy Actuators Embedded in Composites, ICCM Proceedings of the 8th International Conference on Composite Materials, 1990.
- [5] BAZ A., IMAM K. et MCCOY J., Active Vibration Control of Flexible Beams Using Shape Memory Actuators, source inconnue, 1989.
- [6] PATOOR E., BENSALAH M.O., EBERHARDT A. et BERVEILLER M., Détermination du comportement thermomécanique des alliages à mémoire de forme par optimisation d'un potentiel thermodynamique, à paraître dans la Revue de Métallurgie, novembre 1992.
- [7] PATOOR E., BENSALAH M.O., TROCHU F. et TERRIAULT P., Minimisation quadratique du potentiel thermodynamique pour une transformation martensitique thermoélastique, communication privée.
- [8] PATOOR E., Modélisation thermomécanique du comportement des matériaux et structures à mémoire de forme, Actes du congrès Technologie et Mise en Oeuvre des Matériaux à Mémoire de Forme, Montréal (Canada), juin 1993.

- [9] PATOOR E., EBERHARDT A. et BERVEILLER M., Thermomechanical Behaviour of Shape Memory Alloys, Arch. Mech., 40, 5-6, pp. 775-794, 1988.
- [10] PATOOR E., EBERHARDT A. et BERVEILLER M., An Integral Equation for the Polycrystalline Thermomechanical Behaviour of Shape Memory Alloys, Proceedings of MECAMAT - International Seminar on the Inelastic Behaviour of Solids : Models and Utilization, Besançon (France), 1988.
- [11] PATOOR E., EBERHARDT A. et BERVEILLER M., Micromechanical Modelling of Superelasticity in Shape Memory Alloys, à paraître dans Pitman Research Notes in Mathematics, avril 1993.
- [12] SUN Q.P. et HWANG K.C., Micromechanics Modelling for the Constitutive Behaviour of Polycrystalline Shape Memory Alloys (part I and II), J. Mech. Phys. Solids, Vol. 41, No. 1, pp. 1-33, 1993.
- [13] LEXCELLENT C. et LICHT C., Some Remarks on the Modelling of the Thermomechanical Behaviour of Shape Memory Alloys, Colloque C4, supplément au Journal de Physique III, Vol. 1, novembre 1991.
- [14] RANIECKI B. et LEXCELLENT C., RL-Models of Pseudoelasticity and their Specification for some Shape Memory Solids, Eur. J. Mech., A/Solids, 13, No. 1, pp. 21-50, 1994.
- [15] LECLERCQ S., LEXCELLENT C., TOBUSHI H. et LIN P.H., Thermodynamical Modelling of Recovery Stress Associated with R-Phase Transformation in TiNi Shape Memory Alloys, à paraître dans Mat. Trans. JIM.
- [16] LECLERCQ S., LEXCELLENT C. et GELIN J.C., Utilisation de la méthode des éléments finis pour l'étude du comportement pseudoélastique de structures en

- alliage à mémoire de forme, Actes du 11ème Congrès Français de Mécanique, Vol. 4, pp.173-176, 1993.
- [17] RANIECKI B., LEXCELLENT C. et TANAKA K., Thermodynamic Models of Pseudoelastic Behaviour of SMA, Arch. Mech., 44, 3, pp. 261-284, 1992.
- [18] BOURDON G. et LEXCELLENT C., Modélisation thermodynamique du comportement cyclique d'alliages à mémoire de forme TiNi et CuZnAl sollicités en traction ondulée, Journal de Physique IV, Colloque C3, supp. au Journal de Physique III, Vol. 4, pp.145-150, février 1994.
- [19] GANGHOFFER J.F., DENIS S., GAUTIER E., SIMON A., SIMONSSON K. et SJÖSTRÖM S., Micromechanical Simulation of a martensitic Transformation by Finite Elements, Journal de Physique IV, Colloque C4, supp. au Journal de Physique III, Vol. 1, pp. 77-82, nov. 1991.
- [20] COLLINS C. et LUSKIN M., Computational Results for Phase Transitions in Shape Memory Materials, source inconnue.
- [21] TANAKA K., A Thermomechanical Sketch of Shape Memory Effect: One-dimensional Tensile Behaviour, Res. Mech., 18, pp. 251-263, 1986.
- [22] TANAKA K., A Phenomenological Description on Thermomechanical Behaviour of Shape Memory Alloys, J. Pressure Vessel Technology, Vol. 112, pp.158-163, mai 1990.
- [23] TOBUSHI H., TANAKA K., KIMURA K., HORI T. et SAWADA T., Stress-Strain-Temperature Relationship Associated with the R-Phase Transformation in TiNi Shape Memory Alloy, JSME International Journal, Series I, Vol. 35, No. 3, pp. 278-284, 1992.

- [24] TOBUSHI H., IWANAGA H., TANAKA K., HORI T. et SAWADA T., Stress-Strain-Temperature Relationship of TiNi Shape Memory Alloy Suitable for Thermomechanical Cycling, JSME International Journal, Series I, Vol. 35, No. 3, pp. 271-277, 1992.
- [25] BRINSON L.C. et LAMMERING R., Finite Element Analysis of the Behaviour of Shape Memory Alloys and their Applications, Int. J. Solids & Structures, Vol. 30, No. 23, pp. 3261-3280, 1993.
- [26] BOYD J.G. et LAGOUDAS D.C., Thermomechanical Response of Shape Memory Composites, J. of Intelligent Material Systems & Structures, Vol. 5, pp. 333-346, mai 1994.
- [27] MÜLLER I. et XU H., On the Pseudo-Elastic Hysteresis, Acta Metall. Mater., Vol. 39, No. 3, pp. 263-271, 1991.
- [28] BRANDON D. et ROGERS R.C., Constitutive Laws for Pseudo-Elastic Materials, J. of Intell. Mater. Syst. and Struct., Vol. 3, pp. 255-267, avril 1992.
- [29] PATOOR E., EBERHARDT A. et BERVEILLER M., Potentiel pseudoélastique et plasticité de transformation martensitique dans les mono- et polycristaux métalliques, Acta Metall., Vol. 35, No. 11, pp. 2779-2789, 1987.
- [30] GILLET Y., PATOOR E. et BERVEILLER M., Mechanical Engineering for Shape Memory Alloys, Proceedings of the International Conference on Martensitic Transformations, 1992.
- [31] GILLET Y., PATOOR E. et BERVEILLER M., Sollicitation non homogène de poutre superélastique avec hystérésis, Congrès Français de Mécanique, septembre 1993, 153-156.

- [32] GILLET Y., PATOOR E. et BERVEILLER M., Beam theory Applied to Shape Memory Alloys, SMST'94, Asilomar, CA (USA), pp. 169-174, 1994.
- [33] ORTIN J., Partiel Hysteresis Cycles in Shape-Memory Alloys: Experiments and Modelling, Journal de Physique IV, Colloque C4, supp. au Journal de Physique III, Vol. 1, pp. 65-70, novembre 1991.
- [34] TERRIAULT P. et TROCHU F., Modélisation par krigeage de la relation contrainte-déformation isothermique d'un alliage à mémoire de forme, Rapport interne, École Polytechnique, génie mécanique, 1994.
- [35] TERRIAULT P., MEUNIER M.A. et TROCHU F., Application of Dual Kriging to Construct a General Phenomenological Material Law for Shape Memory Alloys, à paraître.
- [36] MEUNIER M.A., TROCHU F. et CHARBONNIER P., Modeling of Thermomechanical Fatigue Behaviour in Shape Memory Alloys Using Dual Kriging, Materials & Design, Vol. 17 No. 3, pp.133-139, 1996.
- [37] MEUNIER M.-A., Caractérisation expérimentale, modélisation thermomécanique et étude de la fatigue de pièces en alliages à mémoire de forma (AMF), mémoire de M. Sc. A., École Polytechnique de Montréal, 1995
- [38] TROCHU F. et Brailovski V., Analyse des temps de déclenchement et de réarmement d'actuateurs électriques en AMF, rapport technique projet CDT P1968, École Polytechnique de Montréal, 1995
- [39] PATOOR E., EBERHARDT A. et BERVEILLER M., Potentiel pseudoélastique et plasticité de transformation martensitique dans les mono et polycristaux métalliques, Laboratoire de Physique et Mécanique des Matériaux, U.A. CNRS 1215, Institut Supérieur de Génie Mécanique et Productique, France, 1986

- [40] AIROLDI G., RIVOLTA B. et TURCO C., Heats of transformations as a function of thermal cycling in NiTi alloys, Proceedings of the International Conference On Martensitic Transformation, The Japan Institute of Metals, 1986, pp.691-696
- [41] FUNAKUBO, H., Shape Memory Alloys, Université de Tokyo, Gordon and Breach Science Publishers, New York Londre Paris Montreux Tokyo Melbourne, 1984

ANNEXE I

ESSAI DE VALIDATION POUR L'ÉCHANTILLON CUIVREUX

Cet essai de validation consiste à modéliser le comportement thermomécanique d'une tige d'alliage cuivreux. Notez que l'échantillon a d'abord été stabilisé thermiquement par un cyclage en température entre 20°C et 100°C. Tel que nous l'avons déjà mentionné, la première étape consiste à caractériser l'échantillon. Les essais de DSC ont été réalisés sur un appareil Dupont 2000, jumelé au système d'acquisition Thermal Analyser 9900, tandis que les essais de traction ont été effectués sur un appareil JJ M30K.

La courbe de DSC présentée à la fin de cette annexe (figure I.5) permet de déterminer les températures de transformation et les coefficients de l'équation cubique représentant l'évolution de la chaleur spécifique en fonction de la température qui est utilisée dans le calcul de la température de l'échantillon lors de la transformation de phase (chapitre 2, section 2.2).

Tableau I.1 Températures de transformation.

M_r	M_s	A_r	A_s
62,9°C	74,1°C	82,3°C	94,3°C

À partir de ces informations, il est possible de réaliser les essais de traction dans le domaine superélastique pour différentes températures. Pour chaque température, deux essais de traction sont effectués. Cependant, lors de l'élévation de température, l'échantillon reste fixé aux mors de l'appareil, ce qui a induit de légères précontraintes

dans l'échantillon. La figure I.1 permet de constater que l'échantillon ne nécessite pas de cyclage mécanique afin de stabiliser son comportement.

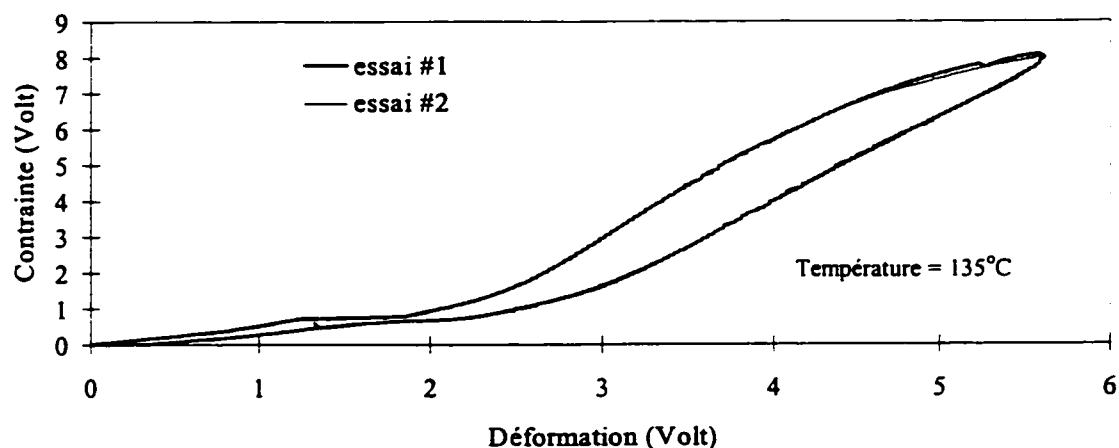


Figure I.1 Influence du cyclage mécanique sur le comportement superélastique de l'alliage cuivreux.

La figure I.2 présente la courbe de traction obtenue à 107°C. Bien que le comportement superélastique soit clairement identifiable, la présence d'un premier plateau pose certaines questions: est-ce imputable au glissement de l'échantillon sur ses mors, à une mauvaise orientation de celui-ci, ou alors à une première transformation de phase?

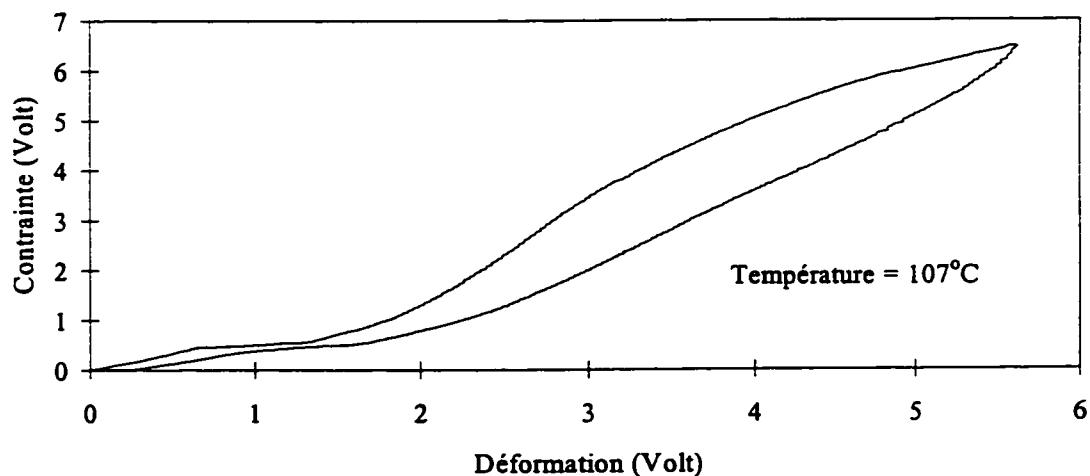


Figure I.2 Courbe superélastique de l'alliage cuivreux.

La figure I.3 présente trois courbes superélastiques de l'alliage cuivreux à des températures différentes. La figure 4 présente un agrandi des plateaux observés sur la figure I.3 pour le chargement seulement. Ces deux figures permettent de conclure que le plateau est imputable à une première transformation de phase. En effet, bien qu'il y ait un léger glissement des courbes causé par la précontrainte, on s'aperçoit que la progression du plateau en fonction de la température élimine l'hypothèse du glissement car celui-ci se produit pour une contrainte et une déformation différentes. De plus, on retrouve également un plateau lors de la décharge, ce qui confirme que le phénomène observé n'est pas dû au glissement.

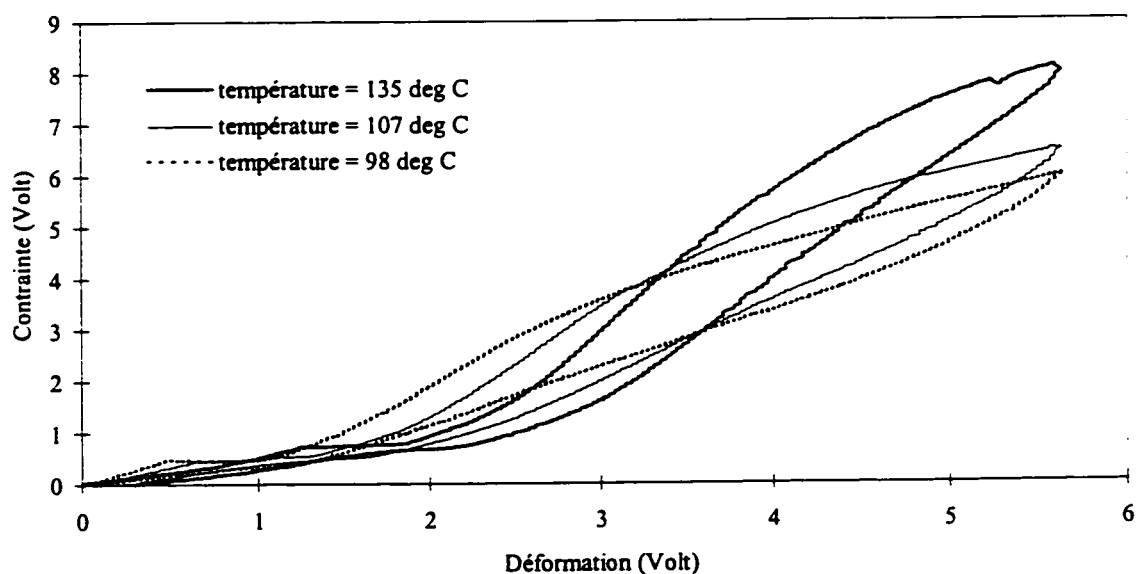


Figure I.3 Courbes superélastiques de l'alliage cuivreux.

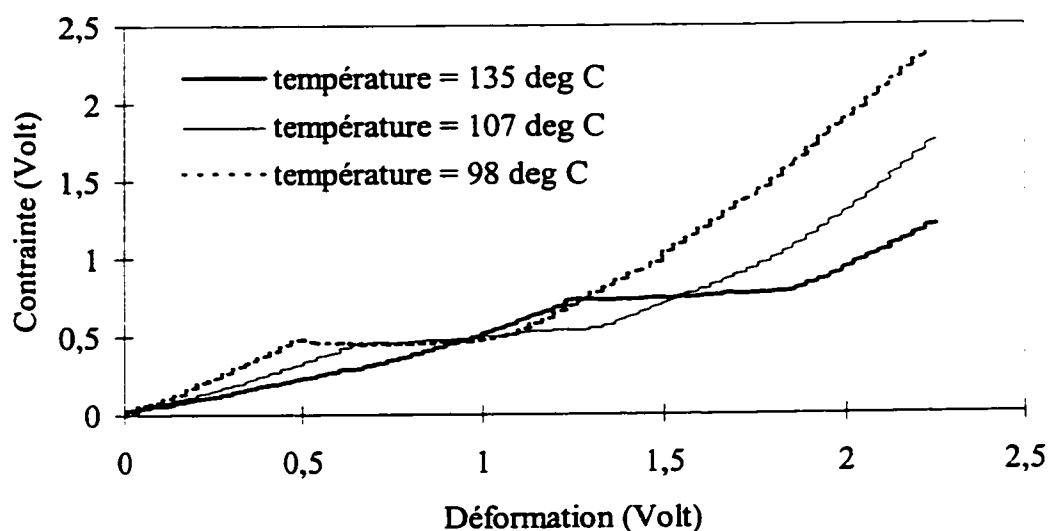


Figure I.4 Courbes superélastiques partielles de l'alliage cuivreux au chargement.

Sachant que les mêmes résultats ont été obtenus avec un deuxième échantillon, on peut éliminer aussi l'hypothèse d'une mauvaise orientation de l'échantillon. Il ne reste donc que l'hypothèse de la transformation de phase. Cette double transformation est imputable à la transformation de la martensite en une autre sorte de martensite lors de l'élévation de la contrainte [41]. Il n'est pas possible de modéliser ce type d'alliage avec un modèle bilinéaire, il serait nécessaire d'utiliser un modèle krigé [36]. Il faut donc envisager l'utilisation d'échantillons présentant un comportement plus simple, tel que les TiNi.

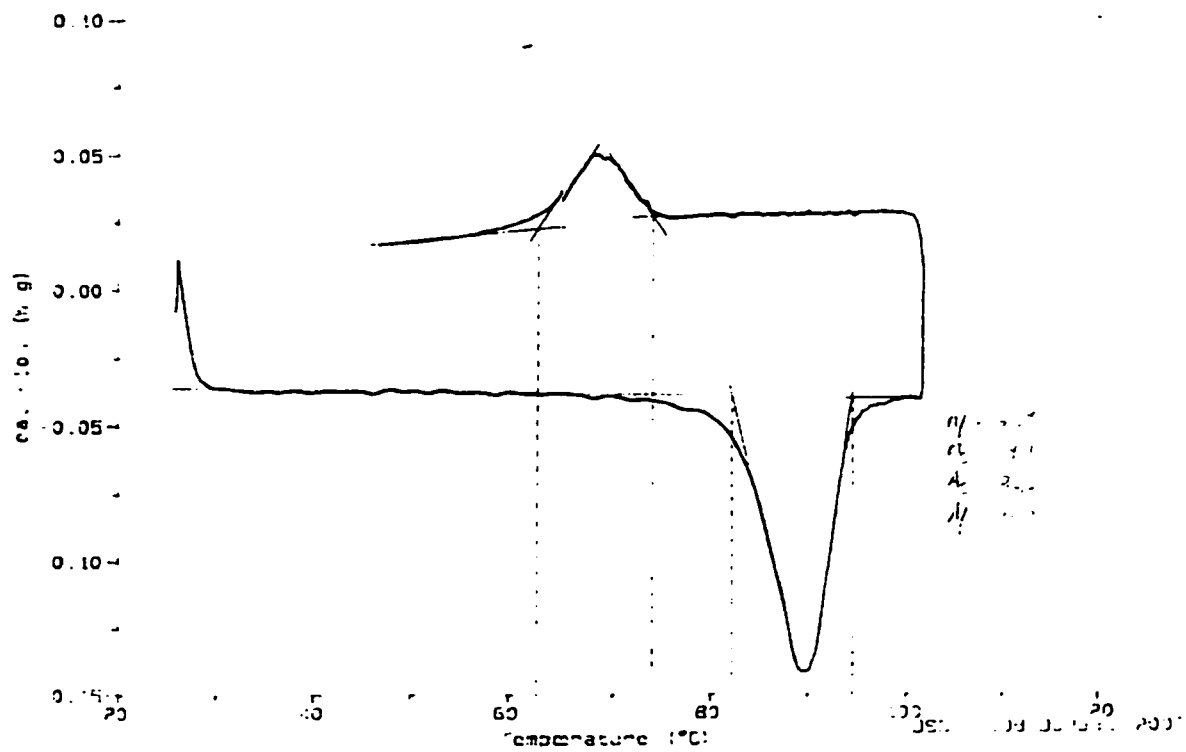


Figure I.5 Courbe de DSC de l'échantillon cuivreux.

ANNEXE II

ESSAI DE VALIDATION POUR L'ÉCHANTILLON DE TITANE-NICKEL NON TRAITÉ

Lors de cet essai de validation, deux types d'échantillons fournis par l'IREQ ont été étudiés. Le tableau II.1 présente les caractéristiques de ces deux types d'échantillons.

Tableau II.1 Caractéristiques des échantillons de TiNi non traités.

	F21951B	ITI-001-30
diamètre (pouce)	0,039	0,01
As théorique (°C)	45	30
traitement thermique	mise en solution à 600 °C pendant 10 minutes et refroidissement à l'air	mise en solution à 600°C pendant 10 minutes et refroidissement à l'air

Un dispositif a dû être fabriqué afin de traiter thermiquement les échantillons. Le but de ce dispositif est de traiter plusieurs échantillons à la fois, tout en s'assurant qu'ils subissent le même niveau de contrainte.

Description du dispositif de traitement thermique

Le dispositif se compose de deux parties principales: le système de mise en charge et le support des échantillons. Ce dernier sert à maintenir une légère contrainte dans les fils lors du traitement thermique. Le système de mise en charge permet d'appliquer la même contrainte à chacun des fils qui seront traités simultanément. Ce dernier se compose de trois pièces : le support (figure II.1), le tendeur (figure II.2) et la vis (figure II.3).

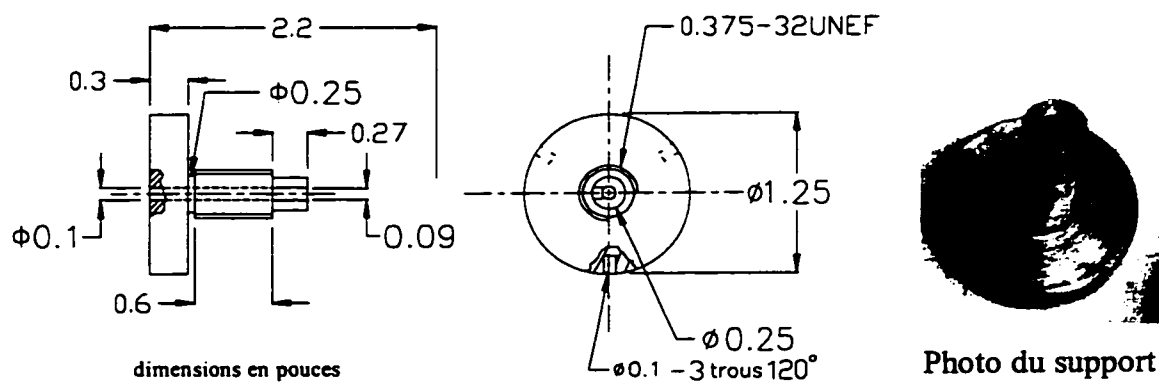


Figure II.1 Dessin de détail du support.

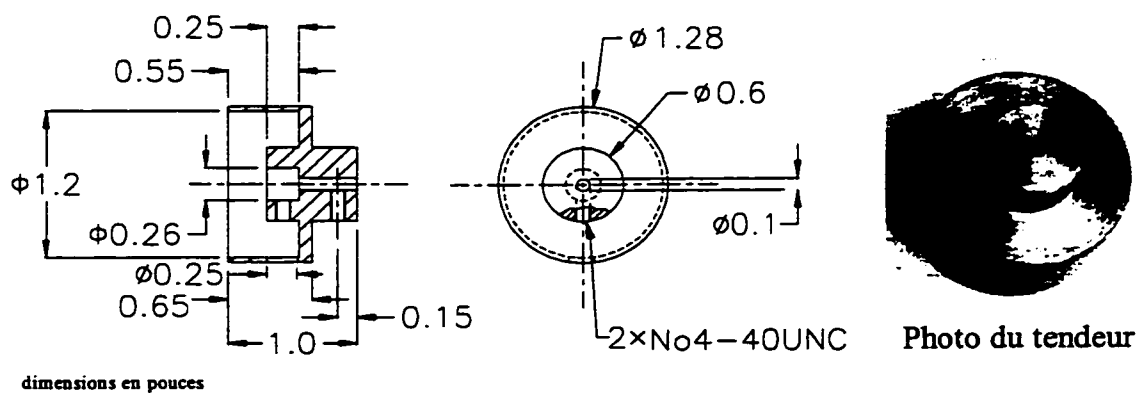


Figure II.2 Dessin de détail du tendeur.

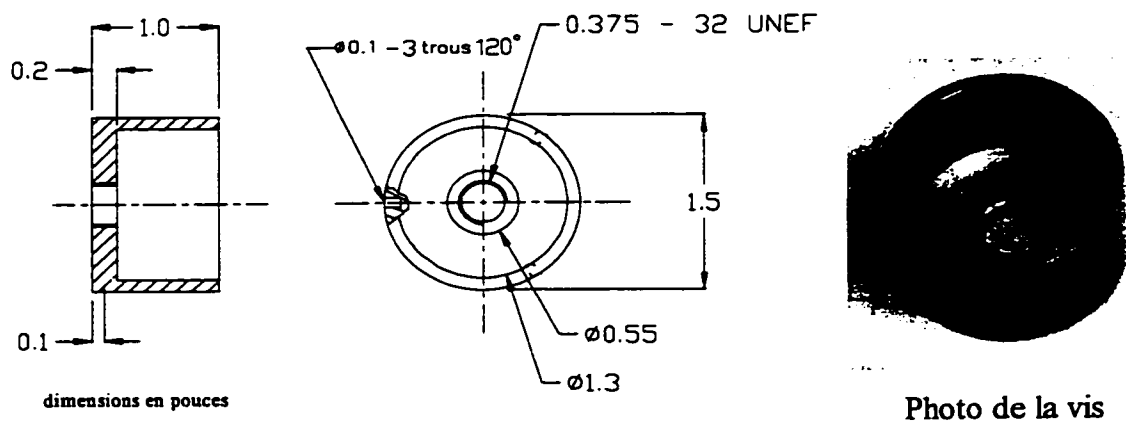


Figure II.3 Dessin de détail de la vis.

Le fil passe dans le trou situé au centre du support (point 1 sur la figure II.4), puis il est fixé au tendeur à l'aide d'une vis (point 2 sur la figure II.4). La vis insérée au point 3 permet au tendeur d'être solidaire au support empêchant ainsi qu'il y aie de la torsion dans le fil AMF. On applique une tension sur le fil en tournant la vis qui pousse sur le tendeur par l'intermédiaire d'un ressort.

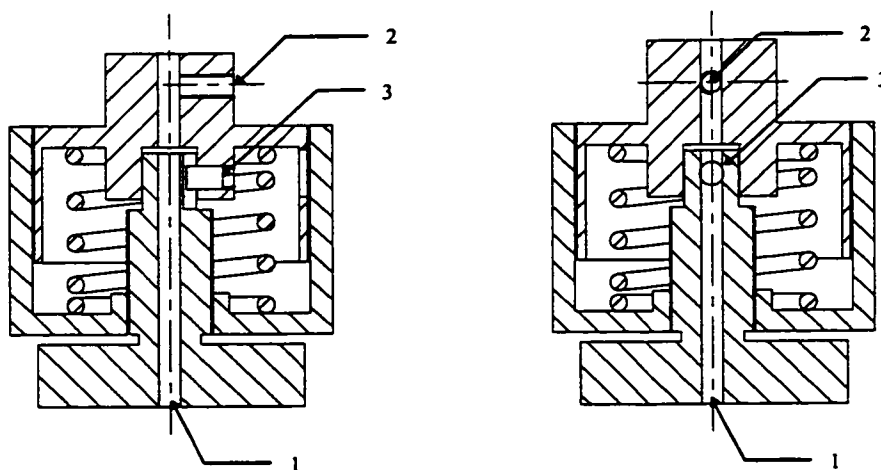


Figure II.4 Dessin d'assemblage du dispositif de mise en charge.

Le support des échantillons est composé de trois pièces : 2 plateaux (figure II.5) et une tige. Le premier plateau est fixé sur la tige à l'aide de deux boulons (figure II.6), tandis que le deuxième est restreint dans son mouvement dans une direction, c'est-à-dire qu'un boulon l'empêche de se rapprocher du premier plateau. Cependant il peut s'en éloigner de façon à ne pas limiter la dilatation des fils lors du traitement thermique.

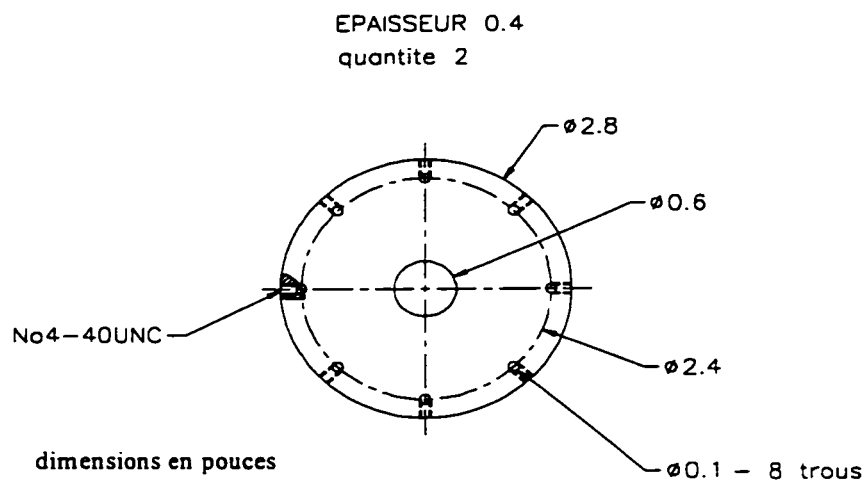


Figure II.5 Dessin de détail des plateaux.

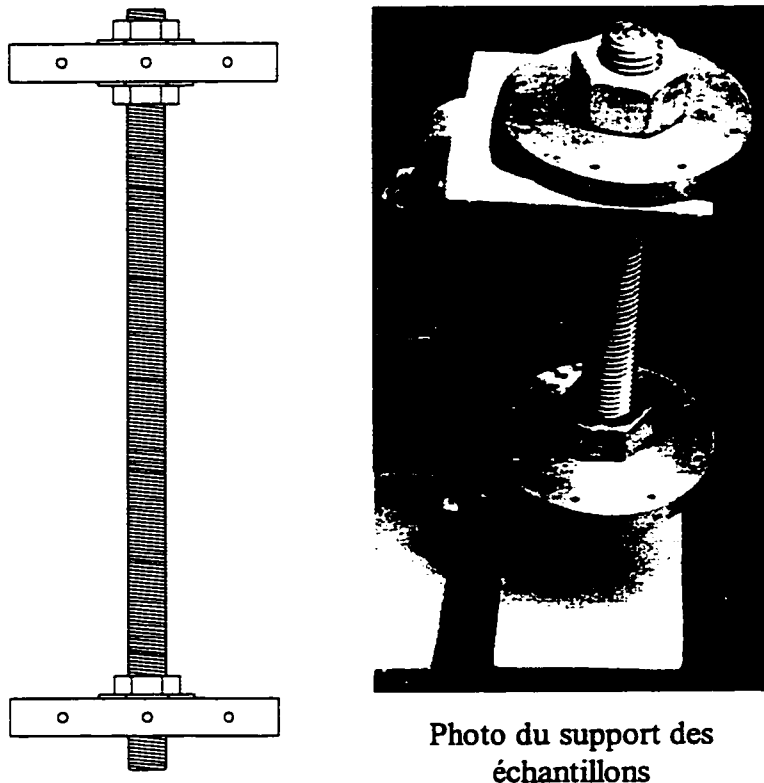


Photo du support des échantillons

Figure II.6 Dessin d'assemblage du support des échantillons.

Fonctionnement du dispositif de traitement thermique

La première étape consiste à fixer le fil sur le plateau mobile, puis de passer son autre extrémité à travers le second plateau. Cette extrémité du fil est ensuite fixée au dispositif de serrage de la façon expliquée précédemment. Ce dernier est alors actionné afin d'étirer le fil. Le fil est ensuite fixé au plateau et le dispositif de serrage peut être retiré. En appliquant le même nombre de tour de vis pour chaque fil, on s'assure qu'ils subissent la même contrainte initiale.

Cependant, ces échantillons n'ont pas été utilisés pour la validation du logiciel. En effet, les résultats obtenus lors de la caractérisation n'ont pas été concluants. Les figures II.7 à

II.9 présentent trois courbes de DSC (appareil Dupont 2000, jumelé au système d'acquisition Thermal Analyser 9900) obtenues pour l'échantillon de TiNi F21951B. On remarque qu'aucune transformation inverse n'est apparente et que les essais ne sont pas reproductibles. De plus en étudiant les courbes de tractions présentées aux figures II.10 et II.11 pour les échantillons de TiNi F21951B (appareil JJ M30K) et de TiNi ITI-001-30 (appareil MTS 810) respectivement, on remarque qu'il n'y a pas de domaine superélastique.

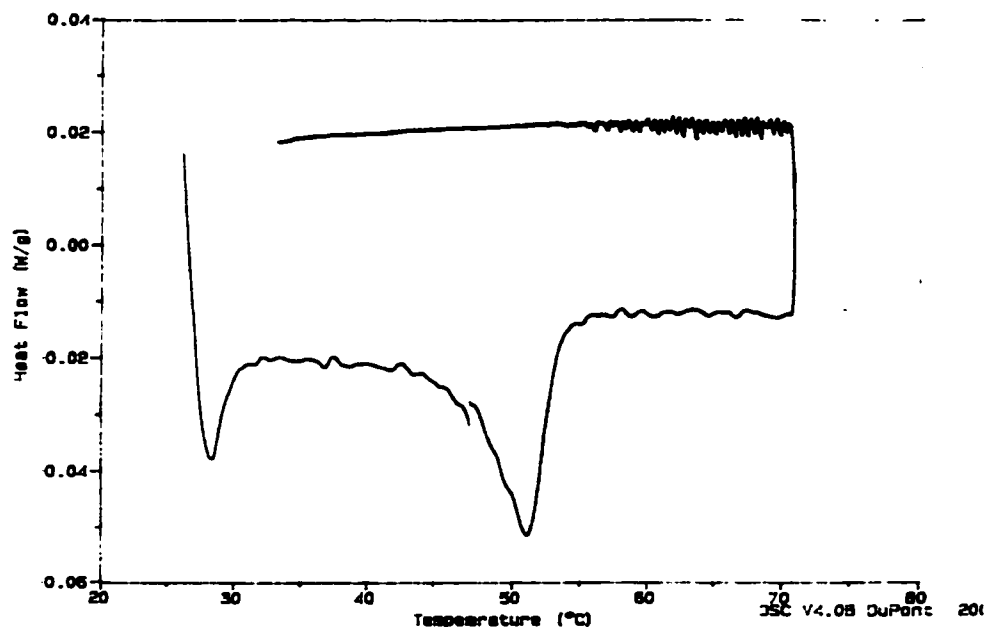


Figure II.7 Courbe de DSC pour l'échantillon de TiNi F21951B (1^{er} essai).

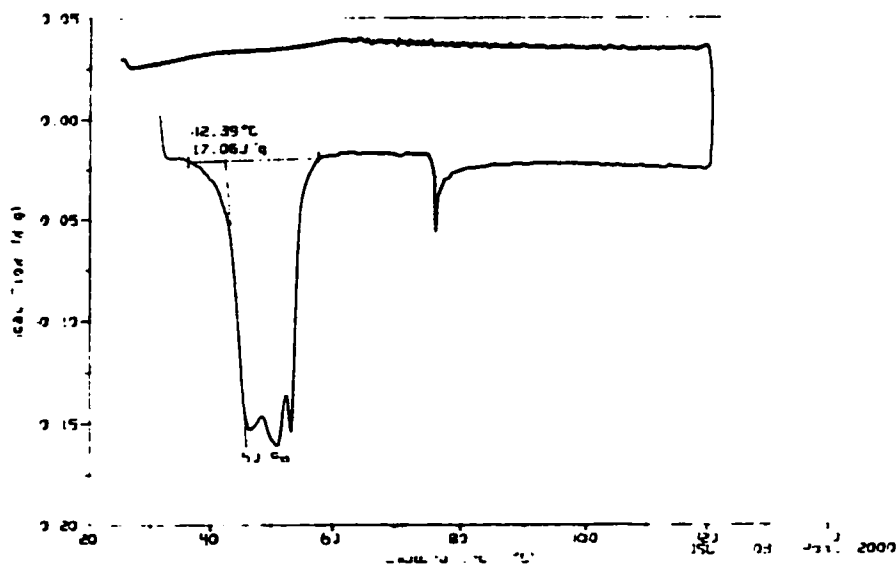


Figure II.8 Courbe de DSC pour l'échantillon de TiNi F21951B (2^{ème} essai).

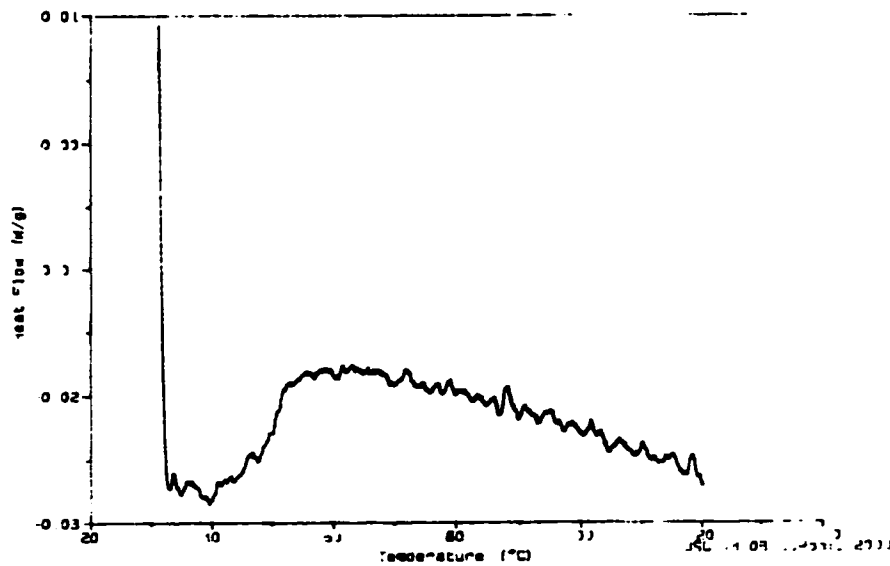


Figure II.9 Courbe de DSC pour l'échantillon de TiNi F21951B - 3^{ème} essai.

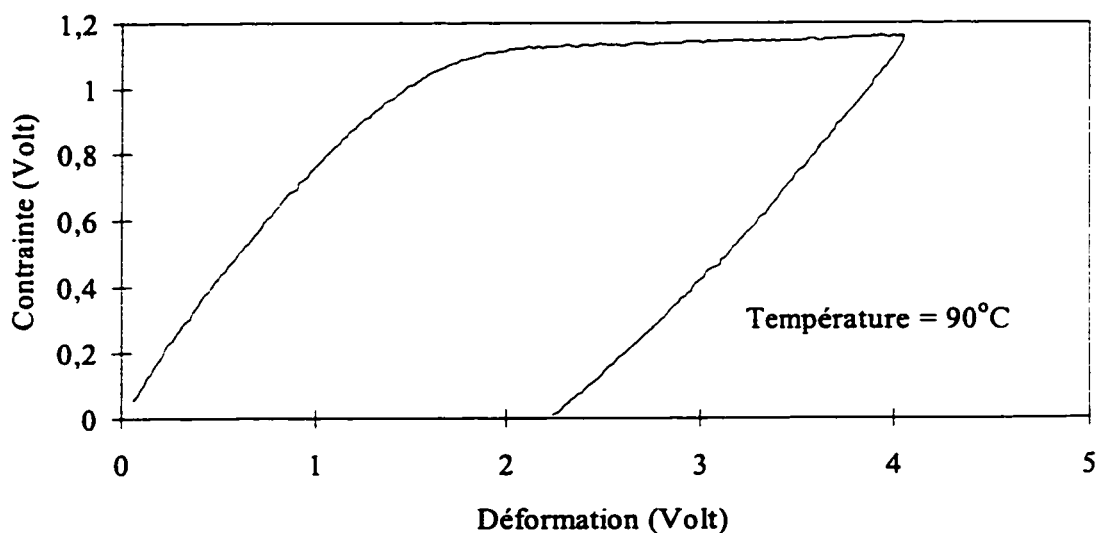


Figure II.10 Courbe de traction pour l'échantillon de TiNi F21951B à 90°C.

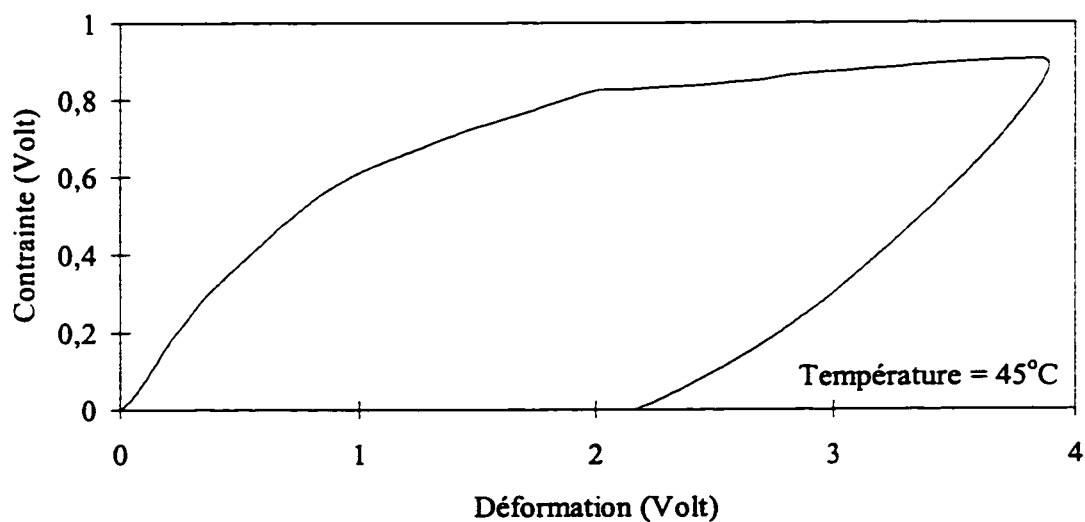
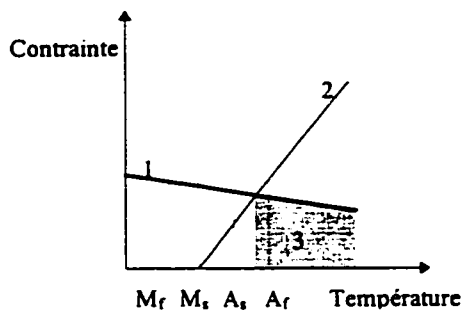


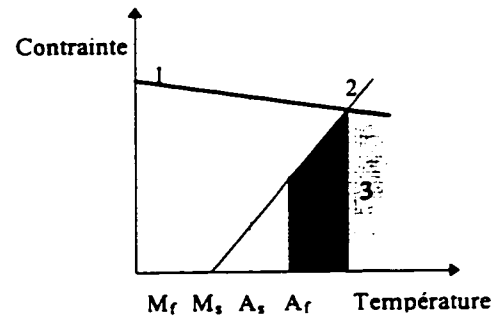
Figure II.11 Courbe de traction pour l'échantillon de TiNi ITI-001-30 à 45°C.

Les problèmes rencontrés lors des différents essais sur les fils de TiNi de l'IREQ sont imputables à la plastification de l'échantillon avant même que la transformation martensitique ait lieu, ce qui, par conséquent, empêche le phénomène de superélasticité

de se manifester. Effectivement, l'étude des figures II.12 et II.13 démontre clairement que si la contrainte de plastification ("slip strain") de l'échantillon n'est pas assez élevée, l'échantillon se plastifie avant que la transformation soit complétée. Ceci provoque alors une déformation permanente de l'échantillon, comme cela fut déjà observé dans les courbes de traction obtenues aux figures II.10 et II.11.



- 1- contrainte de plastification
- 2- contrainte d'induction de martensite
- 3- zone de plastification
- Pas de zone de superélasticité



- 1- contrainte de plastification
- 2- contrainte d'induction de martensite
- 3- zone de plastification
- 4- zone de superélasticité

Figure II.12 σ_{plast} plus petite que σ_{M_s} .

Figure II.13 σ_{plast} plus grande que σ_{M_s} .

Il s'agit donc d'élever la contrainte de plasticité. Pour ceci, trois facteurs sont à considérer:

- la température de recuit ("annealing temperature")
- la température de revenu ("aging temperature")
- la concentration de Ni

Bien que le facteur de concentration de Ni soit hors de notre contrôle, on peut malgré tout mentionner que plus le contenu en Ni est élevé, plus on favorise la formation de précipités lors du revenu, ce qui aide à augmenter la contrainte de plastification. En effet, cette augmentation est imputable à la difficulté de glissement des réseaux complexes de dislocations internes ainsi créés. De la même façon, les deux autres

facteurs contribuent à augmenter cette contrainte soit en provoquant un durcissement par réarrangement thermique des dislocations créées préalablement par un travail à froid dans le cas du premier, ou bien en créant un durcissement par précipitation dans le cas du second.

Malheureusement, il n'existe pas dans la littérature de modèles qui pourraient être utilisés pour prévoir ces deux températures critiques en fonction de l'alliage de TiNi que nous désirons traiter. Cependant, chaque auteur donne sa "recette", dont les plus fréquentes sont :

- (a) recuit à 1273 K pendant 1 heure suivi d'un refroidissement graduel;
- (b) traitement en solution à 1273 K pendant 1 heure suivi d'une trempe à l'eau glacée puis vieillissement à 673 K pendant 1 heure suivi d'une trempe à l'eau glacée;
- (c) recuit à 673 K pendant 1 heure suivi d'une trempe à l'eau glacée immédiatement après un écrouissage à froid.

Le traitement (c) permet un réarrangement thermique des dislocations créées par le travail à froid, tandis que le traitement (b) crée un durcissement par précipitation. Ces deux traitements contribuent à arrêter le mouvement des dislocations internes et donc à augmenter la contrainte de plastification de l'échantillon. Il est donc possible de contourner notre problème, mais les traitements thermiques nécessaires s'avèrent trop complexes pour les ressources dont nous disposons. Il a donc été décidé d'utiliser des échantillons déjà traités.

ANNEXE III

COURBE CARACTÉRISTIQUE DE STABILISATION DE
L'ÉCHANTILLON DE TITANE-NICKEL

La figure III.1 présente la courbe caractéristique de stabilisation de l'échantillon de TiNi 950315 A obtenue par le manufacturier (Shape Memory Applications Inc.).

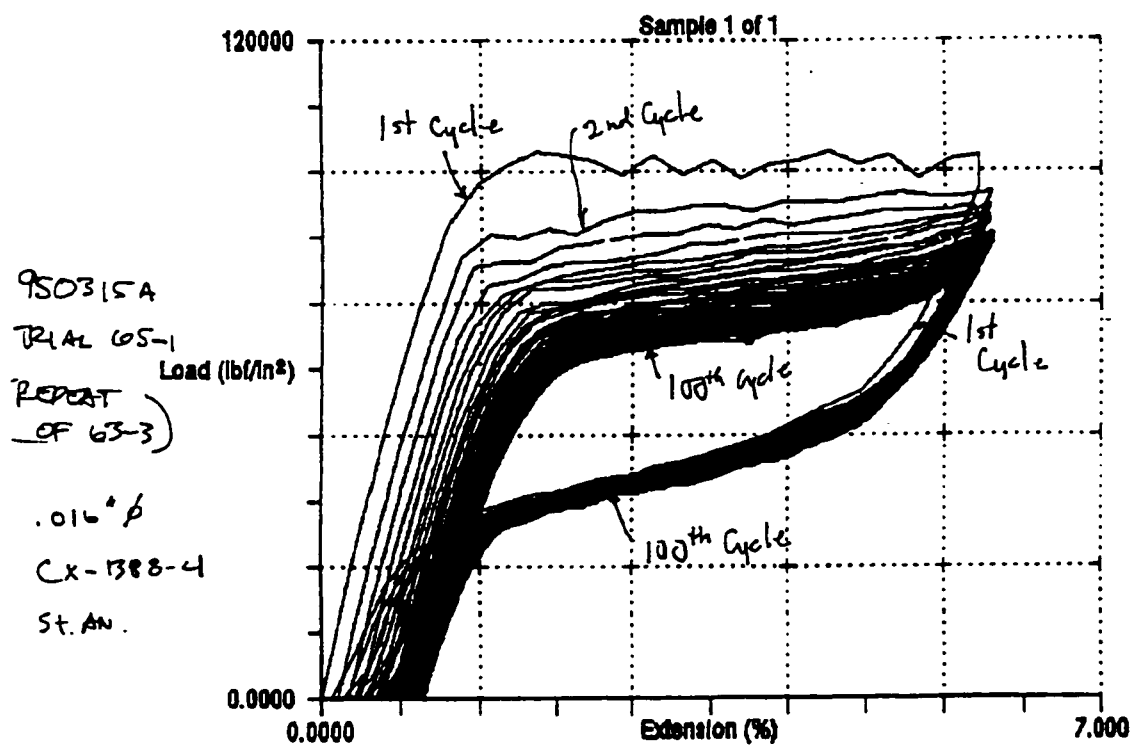


Figure III.1 Courbe de stabilisation de l'échantillon de TiNi traité.

ANNEXE IV

FORMATS DES FICHIERS DES LIBRAIRIES INFORMATIQUES

Cette annexe présente les formats de tous les fichiers utilisés par les librairies de modélisation. Pour chaque fichiers, la fonction de lecture ou d'écriture correspondante est donnée. Une documentation détaillée des fonctions est disponible à l'annexe V.

Fichier des données de l'AMF

Fonction : TYP_LireFichierAmfData

#IDENT Amf théorique pour validation	// identification de l'AMF
#TEMPERATURES 10.0 10.0 50.0	// Tref, Tmin et Tmax
#DSC -74.0 -65.0 -68.5 -61.0	// M_p , M_s , A_s , A_f
#PENTES 24600.0 213.92 24600.0 100.2946964	// Aust, transfo, mart, insta
#SIGMATRANSFO 188.25 178.21	// sigma M_s , sigma A_f
#LIMITES 8.8 350	// défo de transfo, S_y
#DSDT 2.51	// pente (sigma / Temp)

Fichier de chargement pour contrainte variable

Fonction : TYP_LireFichierSeqTemperature

#SEQT 0.05 3	// incrément de température, nombre de points
10.0	// première température
45.0	
10.05	// dernière température

Fichier de chargement pour contrainte constante

Fonction : TYP_LireFichierSeqTemp_ContDefo

```
#SEQTC 2.0 1.0 4 // incrément Temp, incrément contr, nb de points
10.0 0.0 // premier points -> température, contrainte
10.0 245.0
45.0 245.0
10.01 245.0 // dernier point -> température, contrainte
```

Le cas de la contrainte constante est un cas particulier de chargement. La librairie peut traiter les cas plus généraux suivants:

```
#SEQTC 0.1 0.25 5 et #SEQTD 0.01 0.05 5
49.0 0.0 20.0 0.0
49.0 7.0 20.0 2.0
22.0 7.0 20.0 0.01
49.0 7.0 20.0 6.0
49.0 0.1 20.0 0.01
```

où SEQTC : séquence de chargement en température et contrainte.

SEQTD : séquence de chargement en température et déformation.

Fichier du contre-ressort

Fonction : CAL_LireFichierCRess et CAL_EcrireCRess

```
#CRDF 3 // nombre de points
0.0 0.0 // premier point -> déflexion, force
0.4 0.0000436
0.5 0.0001308 // dernier point -> déflexion, force
```


Fichier des paramètres de l'expérience

Fonction : THR_LireFichierDataThermique

```

#GEOMETRIE 0.3 0.0008 // longueur et diamètre
#PROPRIETES 0.000000725 6500 700 300 // résistivité, rho, Cp, h
#COEFFICIENTS -9200 9200 700 -9200 9200 700 // coef chauff, coef refroid
#HYSTERESIS 20.0 20.0 // hysteresis en température
#COURANT 75 // courant électrique
#BUTEE 0.05 // défo AMF lors de butée
#CORR 0.02 // facteur de correction

```

Fichier de chargement - temps des échanges thermiques imposés

Fonction : THR_LireFichierSeqTemps

```

#CONTROLE 1.0 0.00001 // incrément de la contrainte, erreur de Newton
#CONTRAINTE 560.4 // contrainte initiale
#TEMPS 322.0 330.0 // temps de chauffage, temps de refroidissement

```

Fichier de la courbe déflexion-force-température d'un ressort hélicoïdal AMF

Fonction : RES_EcrireCourbeDFT

```

#IDENT mem d sens ass.
#DFT 4
0.0000000 0.0000000 0.0000000 // 1er points déflexion-force-température
0.3838281 1.0000000 23.0000000
0.7676562 2.0000000 23.0000000
1.1514844 3.0000000 23.0000000 // dernier points

```

Fichier de la courbe τ - γ obtenue par calcul inverse sur un ressort hélicoïdal AMF

Fonction : RES_EcrireCourbeTGpourRessort

```
#IDENT  identification du ressort           // identification du ressort
#TEMP 20.000000                          // température de l'essai
#DSC -24.0000 -18.0000 -11.0000 -6.0000  // température de transformation
#GEOM -24.0000 -18.0000 -11.0000 -6.0000 // géométrie du ressort
```

#TGC 5

```
0.0000000 0.0000000 // 1er point déformation-contraintes au chargement
0.0003746 5.3663243
0.0004060 5.8053281
0.0004400 6.2708493
0.0004580 6.5144318 // dernier point au chargement
```

#TGD 4

```
0.0000000 0.0000000 // 1er point au déchargement
0.0003822 5.4746740
0.0004060 5.8082924
0.0004489 6.4021242 // dernier point au déchargement
```

Fichier de données d'un ressort hélicoïdal AMF

Fonction : RES_LireFichierRessort

```
#GEOM 3.4 0.8 6 15 // dia moyen, dia spire, nb spires actives,
                    // hauteur austénitique de charge nulle
#TEMP 20 // température de l'essai
#IDENT identification du ressort // identification du ressort
#DSC -24 -18 -11 -6 // température de transformation
```

```

#DFC 6
0      0      // 1er point déflexion-force au chargement
0.71111111 2.04395604
1.31111111 3.26373626
2.42222222 4.375
5.6     6.13186813
8.04444444 7.15384615 // dernier point déflexion-force au chargement

```

```

#DFD 5
0      0      // 1er point déflexion-force au chargement
0.33333333 0.98901099
1.2     2.57142857
2.24444444 3.42857143
8.04444444 7.15384615 // dernier point déflexion-force au déchargement

```

Fichier de la distribution de τ - γ sur le rayon d'une tige en torsion

Fonction : CAL_EcrireDistributionTauGammaSurRayon

nombre d'éléments 3 $r=1.5$

r	T	Gamma	Tau
0	20.0	0.0000000	0.0000000
0.5	20.0	0.0003746	5.3663243
1.0	20.0	0.0004060	5.8053281
1.5	20.0	0.0004400	6.2708493

ANNEXE V

DOCUMENTATION DES LIBRAIRIES DE MODÉLISATION BILINÉAIRE DU COMPORTEMENT THERMOMÉCANIQUE DES AMF

On retrouve dans cette annexe la documentation des librairies informatiques de modélisation bilinéaire du comportement thermomécanique des alliages à mémoire de forme. On s'intéresse premièrement à la présentation générale des différentes librairies. Les cinq sections suivantes présentent la documentation sommaire des types et des fonctions de chacune des librairies. Finalement, la dernière section présente la documentation détaillée de tous les types et fonctions classés en ordre alphabétique selon chaque librairie.

1. PRÉSENTATION DES LIBRAIRIES

Il y a en tout cinq librairies : **typamf_l.lib**¹, **matamf_l.lib**, **calamf.lib**, **thramf_l.lib** et **resamf_l.lib**. La librairie **typamf_l.lib** (*typ* pour type) contient les fonctions qui permettent de gérer les différents types de variables communs à toutes les librairies. Les fonctions de calcul de la loi de comportement de l'AMF sont contenues dans la librairie **matamf_l.lib** (*mat* pour matériau). Ces deux librairies constituent le coeur du modèle et doivent systématiquement être appelées par tous les programmes utilisant le modèle bilinéaire. La librairie **calamf_l.lib** (*cal* pour calcul) contient toutes les fonctions de calcul de sollicitation mécanique simple : traction uniaxiale d'une pièce de section constante, torsion d'une tige cylindrique, système de butée et/ou de contre-ressort pour le cas de la traction et chargement en température (échauffement et refroidissement). Cette librairie est

¹ Le suffixe "_l" est utilisé pour noter la nature linéaire de la loi de comportement du matériau.

en fait l'outil qui permet à un usager d'utiliser le modèle bilinéaire afin de calculer la réponse d'un AMF à un cas de sollicitation thermomécanique donné.

Les deux autres bibliothèques sont des exemples de modules complémentaires qui peuvent être facilement ajoutés aux bibliothèques de base pour agrandir l'éventail des cas traités par le modèle. La bibliothèque **thramf_l.lib** (*thr* pour thermique) permet de calculer le temps d'échauffement par effet joule d'un actuateur AMF à fil droit, tandis que la bibliothèque **resamf_l.lib** (*res* pour ressort) est utilisée pour le calcul de ressort AMF hélicoïdaux.

À chaque bibliothèque est associée un fichier d'en-tête du même nom qui contient la déclaration des fonctions présentes dans la bibliothèque. Le fichier de **typamf_l.h** présente également la déclaration des types généraux² de variables spécifiques au modèle. Chaque fichier d'en-tête des bibliothèques complémentaires doit contenir la déclaration des types de variables qui leur sont spécifiques. De plus, le nom de toutes les fonctions appartenant à une bibliothèque commence par les trois premières lettres de cette bibliothèque afin de faciliter la lecture du code source d'un programme utilisant les bibliothèques. Chaque bibliothèque peut être compilée individuellement afin de simplifier la mise à jour de ces dernières et des programmes qui les utilisent. Cependant, **calamf_l.lib** fait directement appel aux fonctions de **matamf_l.lib**. Il est donc toujours nécessaire d'appeler cette dernière lors de l'utilisation de **calamf_l.h**. Les bibliothèques complémentaires **thramf_l.lib** et **resamf_l.lib** ne possèdent pas un tel lien de dépendance et pourraient donc être utilisées avec un autre modèle de calcul thermomécanique d'AMF. L'utilisation de la bibliothèque **typamf_l.lib** pourrait également être évitée, mais toutes les fonctions de gestion des types des bibliothèques seraient perdues. Notez cependant que le fichier **typamf_l.h** doit toujours être appelé car il contient la déclaration de types qui sont utilisés par toutes les bibliothèques.

² On parle ici de types généraux car chaque bibliothèque peut posséder ses propres types en ce qui concerne son fonctionnement interne mais elle doit utiliser les types définis dans **typamf_l.h** pour l'échange d'informations avec les autres bibliothèques

2. TYPAMF_L.LIB : LIBRAIRIE DE GESTION DES TYPES

La librairie `typamf_l.lib` contient les fonctions qui permettent de gérer les types généraux. Cette section présente chaque type de variables ainsi que les fonctions nécessaires à leur utilisation. Deux groupes de types sont identifiés : les types utilisés pour le calcul de la loi de comportement et ceux utilisés pour la manipulation des points ainsi que des courbes *contrainte-déformation-température* de chargement et de réponse d'un AMF. Notez que seuls les types nécessitant des pointeurs possèdent des fonctions *Créer* et *Détruire*. Toutes les variables d'un type qui ne possède pas ces deux sortes de fonctions ne devraient pas être déclarées sous forme de pointeur.

De plus, la librairie `typamf_l` contient une fonction qui permet de retourner la valeur absolue d'une variable de type *double* (nombre réel double précision) car toutes les variables numériques utilisées par la librairie sont du type *double*. Cette fonction est nécessaire car les librairies standards de C n'offrent que des fonctions qui retournent la valeur absolue d'un *int* (nombre entier) ou d'un *float* (nombre réel simple précision).

- **ValAbs** calcule la valeur absolue d'un nombre.

2.1 Types pour le calcul de la loi de comportement

Les types présentés dans cette section sont utilisés pour la mémorisation et la manipulation des propriétés de l'AMF ainsi que des points définissant les plans du domaine superélastique et de toutes les informations concernant l'histoire thermomécanique de l'AMF lors d'un chargement.

2.1.1 Type *AmfData*

Le type *AmfData* est utilisé pour les variables qui contiennent l'information sur les propriétés thermomécaniques de l'AMF : températures de transformation, pentes des plans principaux du domaine superélastique, etc.

Fonctions associées :

- **TYP_CreerAmfData** réserve l'espace mémoire nécessaire à une variable de type *AmfData* et initialise ses différents paramètres à 0.
- **TYP_DetruireAmfData** libère l'espace mémoire réservé pour les données de l'AMF.
- **TYP_LireFichierAmfData** lit, dans un fichier, les données thermomécaniques de l'alliage et les assigne à une variable de type *AmfData* en plus de construire les trois plans principaux.

2.1.2 Type *Plan*

Le Type *Plan* est utilisé pour les variables qui contiennent les points qui définissent un plan de chargement du domaine superélastique linéarisé.

Fonction associée :

- **TYP_CopierPlan** crée une copie d'un plan contrainte-déformation-température.

2.1.3 Type *TypeDePlan*

Le type *TypeDePlan* est utilisé pour les variables qui indiquent la nature du plan sur lequel se trouve le point *contrainte-déformation-température* courant. Ces variables permettent l'identification de l'état de sollicitation de l'AMF et de suivre la progression des transformations de phase successives de l'AMF. La figure V.1 illustre les différents plans définis par le modèle bilinéaire.

Fonctions associées :

Aucune

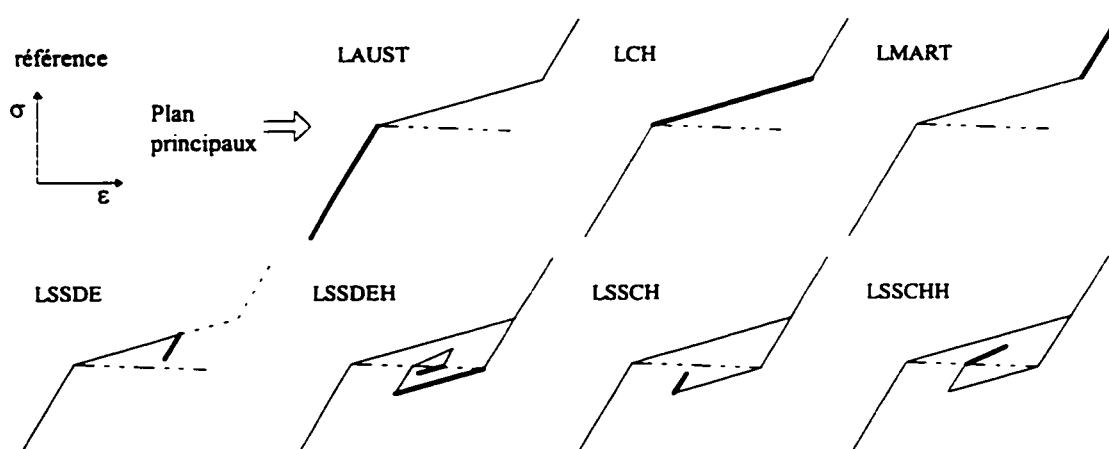


Figure V.I Plans de chargement définis dans le modèles bilinéaire

2.1.4 Type *Noeud*

Le type *Noeud* est utilisé pour les variables qui contiennent l'information sur l'état de sollicitation thermomécanique en un point d'un échantillon. Lorsque la contrainte est constante sur la section un seul noeud est nécessaire. Dans le cas contraire, il faut utiliser une série de noeuds pour représenter la section mise sous contrainte. Cette approche est notamment utilisée lors du calcul de la torsion d'une tige.

Fonctions associées :

- **TYP_CreerNoeud** crée un noeud et l'initialise (paramètres = 0 et plan de chargement = LAUST).
- **TYP_DetruireNoeud** libère l'espace mémoire réservé pour un noeud.
- **TYP_DetruirePlansNoeud** libère l'espace mémoire d'un noeud qui est réservé pour les plans représentant son état thermomécanique.
- **TYP_ModifierNbPlansNoeud** modifie le nombre de plans utilisés pour représenter l'état thermomécanique d'un noeud.
- **TYP_CopierNoeud** crée une copie d'un noeud.
- **TYP_CopierSerieDeNoeud** crée une copie d'une série de noeuds.

2.2 Types pour la gestion des points et des courbes

Les types présentés dans cette section sont utilisés pour la mémorisation et la manipulation des points et des courbes *contrainte-déformation-température* qui représentent les chargements thermomécaniques imposés ainsi que ceux qui représentent la réponse d'un AMF à ces chargements.

2.2.1 Type *Point*

Le type *Point* est utilisé pour les variables qui contiennent un point dans un espace 3D. Généralement, ces variables sont utilisées pour les points contrainte-déformation-température d'un essai mécanique.

Fonctions associées :

- **TYP_CopierPoint** crée une copie d'un point

2.2.2 Type *Courbe*

Le type *Courbe* est utilisé pour les variables qui contiennent une courbe constituée de points de type *Point*.

Fonctions associées :

- **TYP_CreerCourbe** réserve l'espace mémoire pour une courbe.
- **TYP_DetruireCourbe** libère l'espace mémoire réservé pour une courbe.
- **TYP_InverserCourbe** inverse l'ordre de la liste des points d'une courbe.
- **TYP_EcrireCourbe** écrit les points d'une courbe dans un fichier.
- **TYP_InterpolerPointSurCourbe** interpole un point sur une courbe.

2.2.3 Type *PointChn*

Le type *PointChn* est utilisé pour les variables qui contiennent les points τ - γ caractérisant l'alliage d'une tige en torsion. Ces variables contiennent également les points moment-angle-température qui caractérisent le comportement de la tige. Le suffixe *Chn* identifie ce type de point comme étant utilisé à l'intérieur d'une liste chaînée (*CourbeChn*).

Fonctions associées :

Aucune.

2.2.4 Type *CourbeChn*

Le type *CourbeChn* est utilisé pour les variables qui contiennent une courbe sous forme de liste chaînée constituée de points de type *PointChn*.

Fonctions associées :

- **TYP_CreerCourbeChn** crée une référence à la liste des points de la courbe chaînée.
- **TYP_DetruireCourbeChn** libère l'espace mémoire réservé pour une courbe chaînée.
- **TYP_AjouterUnPointChnSurCourbeChn** crée un nouveau point et l'ajoute à la fin d'une courbe chaînée.
- **TYP_InterpolerPointChnSurCourbeChn** interpole un point sur une courbe chaînée.
Le paramètre interpolé est la déformation pour une contrainte et une température connues.

2.2.5 Le type *SeqTP*

Le type *SeqTP* est utilisé pour les variables qui contiennent la liste des points température-(contrainte ou déformation) de chargement³ imposé à l'AMF.

³ Le terme *chargement* désigne la séquence complète de chargement-déchargement imposée à l'AMF.

Fonctions associées :

- **TYP_CreerSeqTP** réserve l'espace mémoire pour une variable de type SeqTP. Cependant, le nombre de points de la séquence de chargement n'étant pas connus lors de la création de la variable, la valeur *néant* (NULL) est attribuée aux deux pointeurs de types *double** qui pointent sur les séquences de chargement.
- **TYP_DetruireSeqTP** libère l'espace mémoire réservé pour la séquence de chargement.
- **TYP_LireFichierSeqTemp_ContDefo** lit dans un fichier et calcul la séquence de chargement qui doit être imposée et l'assigne à une variable de type *SeqTP*.
- **TYP_LireFichierSeqTemperature** lit dans un fichier et calcul la séquence de chargement en température lorsque l'essai est à contrainte variable et l'assigne à une variable de type *SeqTP*.
- **TYP_ModifierSeqTP** réserve l'espace pour les pointeurs aux séquences de chargement d'une variable de type *SeqTP*. Si la variable contient déjà une séquence de chargement, celle-ci est détruite et remplacée par la nouvelle.

3. MATAMF_L.LIB : LIBRAIRIE DE CALCUL DE LA LOI DE COMPORTEMENT

Les fonctions contenues dans cette librairie permettent de créer les différents plans du domaine superélastique construit par le modèle (voir figure V.1), ainsi que de calculer la réponse d'un AMF pour un incrément donné de température et/ou de contrainte ou de déformation.

- **MAT_CalculerPlanLSSCH** calcule les 4 points qui définissent un plan LSSCH (sous-surface de chargement sans hystérésis).
- **MAT_CalculerPlanLSSCHH** calcule les 4 points qui définissent un plan LSSCHH (sous-surface de chargement avec une possibilité d'hystérésis).

- **MAT_CalculerPlanLSSDE** calcule les 4 points qui définissent un plan LSSDE (sous-surface de déchargement sans hystérésis).
- **MAT_CalculerPlanLSSDEH** calcule les 4 points qui définissent un plan LSSDEH (sous-surface de déchargement avec une possibilité d'hystérésis).
- **MAT_InterpolerPointSurDroite** interpole un point sur une section droite de l'espace située entre deux points connus.
- **MAT_InterpolerPointSurPlan** interpole un point sur un plan pour une température et soit une contrainte ou une déformation connue.
- **MAT_CalculerReponseUnNoeud** calcule la réponse d'un noeud soumis à une variation de température et de contrainte ou de déformation.

4. CALAMF_L.LIB : LIBRAIRIE DE CALCUL DE CHARGEMENT SIMPLE

La librairie de calcul de chargement simple contient les fonctions qui permettent de calculer la réponse d'un AMF à un chargement simple : échauffement d'un échantillon AMF, torsion sur une tige et traction sur une pièce de section constante (contrainte uniforme sur la section). Pour ce dernier type de chargement, le mouvement de la pièce peut être restreint par une butée et/ou un contre-ressort. En réalité, les fonctions de cette librairie ne font que calculer et appliquer une séquence de chargement à l'AMF, la réponse de ce dernier (interpolation sur les plans du domaine superélastique) étant calculée par les fonctions de la librairie de calcul de la loi de comportement **matamf_l.lib**. Les fonctions de cette librairie peuvent être regroupés en trois catégories : fonctions de calcul de la traction avec ou sans butée et/ou contre-ressort, fonctions de calcul de la torsion d'une tige cylindrique et fonctions d'entrée/sortie des données.

4.1 Fonctions de calcul de la traction

Les fonctions présentées ici permettent de calculer la réponse d'un AMF soumis à un chargement de traction uniaxiale ou à une élévation de température. Elles sont cependant limitées au cas où la contrainte est uniforme dans la section de l'AMF.

- **CAL_CalculerDefoContrCte** calcule, pour un seul noeud, la déformation associée à chaque point de chargement pour un essai à contrainte imposée constante (avec ou sans butée).
- **CAL_CalculerDefoContrNCte** calcule, pour un seul noeud, la déformation associée à chaque point du chargement pour un essai à contrainte imposée variable, c'est-à-dire avec un contre-ressort (avec ou sans butée).
- **CAL_TrouverPtEquiCRessUnNoeud** calcule la contrainte générée dans l'AMF lorsque ce dernier est en équilibre avec un contre-ressort.

4.2 Fonctions de calcul de la torsion

Cette catégorie de fonctions peut encore être subdivisée en deux sous-groupes : les fonctions de *calcul inverse* de torsion qui permettent de trouver la courbe τ - γ de l'AMF à partir d'une courbe moment-angle représentant la réponse d'une tige en torsion [1], puis les fonctions de calcul de torsion en tant que tel qu'on appellera les fonctions de *calcul direct* [2] de la torsion. Notez que dans les deux cas, l'angle de torsion est un angle unitaire (rad/mm) [3].

4.2.1 Calcul inverse

- **CAL_CalculerTauSurRayonExt** calcule la contrainte de cisaillement sur le rayon extérieur d'une tige soumise à de la torsion en supposant une distribution linéaire du cisaillement sur la section du rayon.

- **CAL_EstimerCourbeTGpourMA** estime la courbe τ - γ à partir de la courbe moment-angle par un calcul inverse.

4.2.2 Calcul direct

- **CAL_CalculerMTorsionSurAnneau** calcule le moment de torsion sur un anneau de la section d'une tige cylindrique.
- **CAL_CalculerMTorsionResultantDeTau** calcule le moment de torsion résultant de la répartition non-linéaire du cisaillement sur le rayon d'une tige en torsion.
- **CAL_ImposerDeltaMTorsionDeltaT** calcule la réponse d'une tige AMF soumise à un incrément du moment de torsion et de la température.
- **CAL_ImposerDeltaAngleDeltaT** calcule la réponse d'une tige AMF soumise à un incrément de l'angle de torsion et de la température.
- **CAL_CalculerCourbeTauGamma** calcule la courbe τ - γ d'une tige cylindrique soumise à une séquence de chargement en torsion.

4.3 Fonctions d'entrée/sortie des données

Il existe en fait qu'une seule fonction dans cette librairie qui permet de conserver les points qui caractérisent la réponse d'un AMF à un chargement donné :

- **CAL_EcrireDistributionTauGammaSurRayon** écrit les points τ - γ - T des noeuds sur la section d'une tige en torsion pour un état de sollicitation donné.

Les autres fonctions d'entrée/sortie se divisent en deux groupes : les fonctions de gestion d'un contre-ressort et les fonctions d'affichage graphique qui ne permettent pas de conserver les points dans un fichier.

4.3.1 Gestion d'un contre-ressort

Ce que nous appelons ici un contre-ressort est en fait tout dispositif qui s'oppose au mouvement de l'AMF avec un effort croissant en fonction de la déflexion. Ces fonctions sont conservées dans la librairie de calcul de chargement simple car le contre-ressort est utilisé comme modificateur du chargement imposé à un AMF. Le cas où le ressort est l'échantillon AMF à modéliser est traité par la librairie `resamf_1.lib`.

- `CAL_LireFichierCress` lit, dans un fichier, les points déflexion-force qui caractérisent le comportement d'un contre-ressort.
- `CAL_LireCress` lit les points déflexion-force qui caractérisent le comportement d'un contre-ressort par le biais d'un dialogue avec l'utilisateur.
- `CAL_LireCteCress` lit la constante de rigidité d'un contre-ressort linéaire par le biais d'un dialogue avec l'utilisateur.
- `CAL_EcrireCress` écrit, dans un fichier, les points déflexion-force qui caractérisent le comportement d'un contre-ressort.

4.3.2 Affichage graphique

Les fonctions d'affichage permettent de visualiser les résultats de calculs à l'aide du logiciel GNUPLOT. Cependant, si les courbes à afficher nécessitent un espace mémoire trop grand (supérieur à 64K), les fonctions d'affichage sont inopérantes. Dans un tel cas, il faut écrire les résultats de calculs dans un fichier de type texte et utiliser un autre logiciel, *EXCEL* en est un bon exemple.

- `CAL_AfficherCourbeCharge3D` affiche la courbe *contrainte-déformation-température* à l'aide du logiciel GNUPLOT (ce dernier doit se situer dans le répertoire courant).
- `CAL_AfficherCourbesCharge2D` affiche la projection de la courbe *contrainte-déformation-température* sur les plans *contrainte-déformation*,

courbe *contrainte-température* et courbe *déformation-température* à l'aide du logiciel GNUPLOT (ce dernier doit se situer dans le répertoire courant).

5. THRAMF_L.LIB : LIBRAIRIE DE CALCUL D'ÉCHANGE THERMIQUE

Cette librairie contient les fonctions utilisées pour le calcul du temps des échanges thermiques lorsqu'un actuateur AMF à fil droit est chauffé par effet joule et refroidit par convection naturelle. Cette librairie fait appel à des nouveaux types de variables utilisées pour contenir et traiter l'information nécessaire au calcul des échanges thermiques. La première partie de cette section présente ces nouveaux types de données, tandis que la seconde présente la documentation des fonctions de la librairies.

5.1 Types des variables

Les types de variables utilisées dans cette librairie ne nécessitent aucune fonction de gestion car les variables qui utilisent ces types n'ont pas à être des pointeurs.

5.1.1 Type Coefficient

Le type *Coefficient* est utilisé pour les variables qui contiennent les coefficients de l'équation quadratique, représentant l'évolution de la chaleur spécifique en fonction de la température (au chauffage et au refroidissement), utilisés pour modéliser la transformation de phase.

5.1.2 Type CtePhysique

Le type *CtePhysique* est utilisé pour les variables qui contiennent les propriétés physiques de l'AMF.

5.1.3 Type *GeomFil*

Le type *GeomFil* est utilisé pour les variables qui contiennent les données géométriques de l'actuateur AMF.

5.1.4 Type *Parametre*

Le type *Parametre* est utilisé pour les variables qui contiennent les paramètres d'un essai d'échauffement par effet Joule d'un fil droit AMF.

5.2 Fonctions

Les fonctions de cette librairie permettent de calculer le temps de réponse d'un actuateur AMF à fil droit à un échauffement par effet Joule. La séquence d'échauffement peut être spécifiée par une température à atteindre ou encore par un temps d'échauffement et de refroidissement. Les fonctions sont regroupées en deux sous-groupes : les fonctions d'entrée/sortie et les fonctions de calcul.

5.2.1 Entrée/sortie des données

- **THR_LireFichierDataThermique** lit, dans un fichier, les données sur les paramètres de l'expérience et les assigne à une variable de type *Parametre*.
- **THR_LireFichierSeqTemps** lit un fichier contenant les données de chargement lorsque le temps des échanges thermiques est imposé.
- **THR_AfficherCourbeDefoTemps** affiche la courbe *déformation-temps* à l'aide du logiciel GNUPLOT (ce dernier doit se situer dans le répertoire courant).
- **THR_EcritureFichierTCDt** écrit, dans un fichier, les points température-contrainte-déformation-temps qui caractérisent la réponse d'un fil AMF soumis à un chargement thermomécanique.

5.2.2 Calcul des échanges thermiques

- **THR_ConvertirSeqTemperatureEnSeqTemps** calcule la séquence de température à imposer à l'AMF à partir du temps des échanges thermiques. Dans ce cas-ci, contrairement au cas où la séquence de température est spécifiée, il ne peut y avoir qu'un seul cycle.
- **THR_ResoudreEquaEchTPhase** résous, par la méthode de Newton, l'équation du troisième degré utilisée dans le calcul du temps lors de la transformation de phase.
- **THR_CalculerEnergie** calcule la perte d'énergie sur un intervalle de temps lors de la transformation de phase.
- **THR_CalculerSeqTemps** calcule la séquence totale de chargement (contrainte et température) appliquée à l'échantillon quand la séquence est spécifiée par un temps d'application du courant.
- **THR_CalculerTemps** calcule le temps requis pour réaliser la séquence de chargement en température.

6. RESAMF_L.LIB : LIBRAIRIE DE CALCUL DE RESSORTS AMF HÉLICOÏDAUX

Les fonctions de cette librairie permettent de calculer la réponse d'un ressort AMF hélicoïdaux à un chargement force-déflexion et/ou à un chargement en température [4].

6.1 Type des variables

La librairie de calcul de ressort hélicoïdaux ne nécessite qu'un seul nouveau type de variable, soit le type *Ressort* qui est utilisé pour les variables qui contiennent les données du ressort

Fonctions associées :

- **RES_CreerRessort** réserve l'espace mémoire pour un ressort et initialise ses paramètres à 0. Cependant, le nombre de points des courbes de chargement et de

déchargement n'étant pas connus lors de la création de la variable, la valeur *néant* (NULL) est attribuée aux deux pointeurs de types *Courbe** qui pointent sur les courbes de chargement et de déchargement du ressort.

- **RES_DetruireRessort** libère l'espace mémoire réservé pour un ressort, incluant celui des deux courbes de type *Courbe*.
- **RES_ModifierRessortCh** réserve l'espace mémoire pour la courbe de chargement d'un ressort.
- **RES_ModifierRessortDe** réserve l'espace mémoire pour la courbe de déchargement d'un ressort.
- **RES_LireDataRessort** lit les données géométriques du ressort par le biais d'un dialogue avec l'utilisateur.
- **RES_LireFichierRessort** lit un fichier contenant les données mécaniques et géométriques d'un ressort et les assigne à une variable de type *Ressort*.

6.2 Fonctions

Les fonctions peuvent être séparées en deux groupes : les fonctions d'entrée/sortie et les fonctions de calcul.

6.2.1 Entrée/sortie des données

- **RES_EcrireCourbeTGpourRessort** écrit les données d'un ressort ainsi que les points de la courbe contrainte-déformation de l'alliage dans un fichier.
- **RES_EcrireCourbeDFT** écrit les points de la courbe déflexion-force-température d'un ressort dans un fichier.

6.2.2 Calcul de ressort

Les fonctions de calcul servent en fait qu'à convertir les résultats de calcul de torsion de la librairie **calamf_1.lib** en résultats de force-déflexion pour un ressort et vice-versa.

- **RES_ConvertirMomentEnForce** converti un moment en force
- **RES_ConvertirForceEnMoment** converti une force en moment.
- **RES_ConvertirDeflexEnAngle** transforme les données de déflexion d'un ressort en données d'angle (rad/mm).
- **RES_ConvertirAngleEnDeflex** transforme les données d'angle (rad/mm) d'un ressort en données de déflexion.
- **RES_ConvertirFDenMA** transforme les données de force-déplacement d'un ressort en données de moment-angle (N·m·rad/mm).
- **RES_ConvertirCourbeRessortFcteEnTcte** transforme une courbe *contrainte-déformation* à force constante en une courbe *contrainte-déformation* à température constante. Cette fonction est encore au stade expérimental et elle fonctionne plus ou moins bien avec les TiNi car elle considère les modules d'élasticité comme étant constants.

7. DOCUMENTATION DÉTAILLÉE DES TYPES ET FONCTIONS DES LIBRAIRIES

Cette section présente la documentation détaillée de tous les types et fonctions que nous retrouvons dans les librairies de modélisation bilinéaire du comportement thermomécanique des AMF. Pour chaque type, on retrouve sa définition et sa syntaxe. De la même façon, on retrouve la définition, syntaxe à utiliser, la signification des arguments à fournir ainsi que la signification de la valeur retournée de chaque fonction.

7.1 Gestion des types (typamf_1.lib)

7.1.1 Types

AmfData

Le type *AmfData* est utilisé pour les variables qui contiennent l'information sur les propriétés thermomécaniques de l'AMF : températures de transformation, pentes des

plans principaux du domaine superélastique, etc.

Déclaration :

```
typedef struct f_AmfData
{
    char    ident[100];    // identification de l'alliage
    double  Af;            // température de transformation Af
    double  As;            // température de transformation As
    double  Ms;            // température de transformation Ms
    double  Mf;            // température de transformation Mf
    double  Tref;          // température de l'essai
    double  Tmin;         // température min. pour la définition de la surface
    double  Tmax;         // température max. pour la définition de la surface
    double  a;            // sigma Ms à Tref
    double  b;            // sigma Af à Tref
    double  A;            // module de départ approché
    double  B;            // module de transformation approché
    double  C;            // module de martensite approché
    double  D;            // pente d'instabilité
    double  defomartmax;  // déformation maximale admissible
    double  Sy;           // contrainte maximale admissible
    double  dsdt;         // pente dans la plan contrainte-température
    Plan*   LAUST;        // ces trois plans sont calculés
    Plan*   LCH;          // à partir des données
    Plan*   LMART;        // précédentes
} AmfData;
```

Courbe

Le type *Courbe* est utilisé pour les variables qui contiennent une courbe constituée de points de type *Point*..

Déclaration :

```
typedef struct f_Courbe
{
    unsigned int  nbcpt;    // nombre de points
    Point*       cpt;      // pointeur à la liste des points
} Courbe;
```

CourbeChn

Le type *CourbeChn* est utilisé pour les variables qui contiennent une courbe sous forme de liste chaînée constituée de points de type *PointChn*.

Déclaration :

```
typedef struct f_CourbeChn
{
    unsigned int nbc;           // nombre de chaînons
    PointChn* init;           // pointeur au premier chaînon
    PointChn* dernier;        // pointeur au dernier chaînon
} CourbeChn;
```

Noeud

Le type *Noeud* est utilisé pour les variables qui contiennent l'information sur l'état de sollicitation thermomécanique en un point d'un échantillon. Lorsque la contrainte est constante sur la section un seul noeud est nécessaire. Dans le cas contraire, il faut utiliser une série de noeuds pour représenter la section mise sous contrainte. Cette approche est notamment utilisée lors du calcul de la torsion d'une tige.

Déclaration :

```
typedef struct f_Noeud
{
    int nbpl; //nombre de plans pour représenter la situation thermomécanique du
noeud
    Plan *pl; // pointeur aux surfaces de chargement-déchargement
    Point pt; // pointeur au dernier point du chargement-déchargement
} Noeud;
```

Plan

Le Type *Plan* est utilisé pour les variables qui contiennent les points qui définissent un plan de chargement du domaine superélastique linéarisé.

Déclaration :

```
typedef struct f_Plan
{
    Point    pt1;    // premier point de la surface
    Point    pt2;    // deuxième point de la surface
    Point    pt3;    // troisième point de la surface
    Point    pt4;    // quatrième point de la surface
    TypeDePlan type; // type de surface qui est défini
} Plan;
```

Point

Le type *Point* est utilisé pour les variables qui contiennent un point dans un espace 3D. Généralement, ces variables sont utilisées pour les points contrainte-déformation-température d'un essai mécanique.

Déclaration :

```
typedef struct f_Point
{
    double x;    // déformation - %
    double y;    // contrainte - Mpa
    double z;    //température - °C
} Point;
```

PointChn

Le type *PointChn* est utilisé pour les variables qui contiennent les points τ - γ caractérisant l'alliage d'une tige en torsion. Ces variables contiennent également les points moment-angle-température qui caractérisent le comportement de la tige. Le suffixe *Chn* identifie ce type de point comme étant utilisé à l'intérieur d'une liste chaînée (*CourbeChn*).

Déclaration :

```
typedef struct f_PointChn
{
    unsigned int no;           // numéro du point
    f_PointChn* next;        // pointeur au prochain point
    double tau;               // valeur de la contrainte
    double gamma;            // valeur de la déformation
    // fin fibre externe
    double moment;           // valeur du moment
    double anglunit;         // valeur de l'angle
    double temp;             // valeur de la température
}PointChn;
```

TypeDePlan

Le type *TypeDePlan* est utilisé pour les variables qui indiquent la nature du plan sur lequel se trouve le point *contrainte-déformation-température* courant. Ces variables permettent l'identification de l'état de sollicitation de l'AMF et de suivre la progression des transformations de phase successives de l'AMF.

Déclaration :

```
typedef enum f_TypeDePlan
{
    LCH,           // chargement
    LDE,           // déchargement
    LMART,         // martensite élastique
    LAUST,         // austénite élastique
    LSSCH,         // sous-surface de chargement
    LSSDE,         // sous-surface de déchargement
    LSSCHH,        // sous-surface de chargement avec hystérésis
    LSSDEH         // sous-surface de déchargement avec hystérésis
} TypeDePlan;
```

SeqTP

Le type *SeqTP* est utilisé pour les variables qui contiennent la liste des points *température-(contrainte ou déformation)* de chargement imposé à l'AMF.

Déclaration :

```
typedef struct f_SeqTP
{
  int NbEtape;      // nombre de points de chargement-déchargement
  int Param;       // 1 = contrainte  2 = déformation
  double* T;       // pointeur à la séquence de température
  double* P;       // pointeur à la séquence de contrainte ou déformation selon Param
} SeqTP;
```

7.1.2 Fonctions

TYP_AjouterUnPointChnSurCourbeChn

Fonction : La fonction crée un nouveau point et l'ajoute à la fin d'une courbe chaînée.

Syntaxe : `void TYP_AjouterUnPointChnSurCourbeChn(CourbeChn* base);`

Arguments : *base* pointe sur la courbe chaînée à laquelle le point doit être ajouté.

Valeur : Aucune, mais le contenu de *base* est modifié.
retournée

TYP_CopierNoeud

Fonction : La fonction crée une copie d'un noeud.

Syntaxe : `void TYP_CopierNoeud(Noeud* nd, const Noeud* nd_in);`

Arguments : *nd* pointe sur la copie du noeud.
nd_in pointe sur le noeud à copier.

Valeur : Aucune, mais *nd* est modifié.
retournée

TYP_CopierPlan

Fonction : La fonction crée une copie d'un plan contrainte-déformation-température.

Syntaxe : `void TYP_CopierPlan(Plan* copie, const Plan* ori);`

Arguments : *copie* pointe sur la copie de la surface.
ori pointe sur la surface à copier.

Valeur : Aucune, mais *copie* est modifié.
retournée

TYP CopierPoint

Fonction : La fonction crée une copie d'un point.

Syntaxe : *void TYP_CopierPoint(Point* copie, const Point*ori);*

Arguments : *copie* pointe sur la copie du point.

ori pointe sur le point à copier.

Valeur : Aucune.
retournée

TYP CopierSerieDeNoeud

Fonction : La fonction crée une copie d'une série de noeud.

Syntaxe : *void TYP_CopierSerieDeNoeud(Noeud* copy, const Noeud* ori, int nbn);*

Arguments : *copy* pointe sur les noeuds copiés.

ori pointe sur les noeuds originaux.

nbn indique le nombre de noeuds à copier.

Valeur : Aucune.
retournée

TYP CreerAmfData

Fonction : La fonction réserve l'espace mémoire nécessaire à une variable de type *AmfData* et initialise ses différents paramètres à 0.

Syntaxe : *AmfData* TYP_CreerAmfData(void);*

Arguments : Aucun.

Valeur : La fonction retourne un pointeur à la variable de type *AmfData* initialisée.
retournée

TYP CreerCourbe

Fonction : La fonction réserve l'espace mémoire pour une série de courbes.

Syntaxe : *Courbe* TYP_CreeCourbe(int nbcpt, int nbcourbe=1);*

Arguments : *nbcpt* contient le nombre de points que doit contenir la courbe.

nbcourbe contient le nombre de courbe (1 par défaut).

Valeur retournée : La fonction retourne un pointeur à la liste des points de la surface. Si *nbcpt* est égale ou inférieur à 0, le pointeur sera nul.

TYP CreerCourbeChn

Fonction : La fonction crée une référence à la liste des points de la courbe chaînée.

Syntaxe : *CourbeChn* TYP_CreerCourbeChn()*;

Arguments : Aucun.

Valeur retournée : La fonction retourne un pointeur à la courbe chaînée.

TYP CreerNoeud

Fonction : La fonction crée une série de noeuds. Chaque noeud est initialisé à :
paramètres = 0 et plan de chargement = LAUST.

Syntaxe : *Noeud* TYP_CreerNoeud(const AmfData* AmfD, int nbnoeud=1)*;

Arguments : *AmfD* pointe sur les données de l'AMF. Les données utilisées sont les coordonnées des points qui définissent les plans de chargement pointés par cette variable et qui sont attribués au noeud.
nbnoeud contient le nombre de noeud (1 par défaut).

Valeur retournée : La fonction retourne un pointeur au noeud.

TYP CreerSeqTP

Fonction : La fonction réserve l'espace mémoire pour une variable de type SeqTP. Cependant, le nombre de points de la séquence de chargement n'étant pas connus lors de la création de la variable, la valeur *néant* (NULL) est attribuée aux deux pointeurs de types *double** qui pointent sur les séquences de chargement.

Syntaxe : *SeqTP* TYP_CreerSeqTP(void)*;

Arguments : Aucun.

Valeur : La fonction retourne un pointeur à la séquence de chargement.
retournée

TYP DetruireAmfData

Fonction : La fonction libère l'espace mémoire réservé pour les données de l'AMF.

Syntaxe : `void TYP_DetruireAmfData(AmfData* AmfD);`

Arguments : *AmfD* pointe sur la variable à détruire.

Valeur : Aucune.
retournée

TYP DetruireCourbe

Fonction : La fonction libère l'espace mémoire réservé pour une série de courbes.

Syntaxe : `void TYP_DetruireCourbe(Courbe* in, int nbcourbe=1);`

Arguments : *in* pointe sur la courbe à détruire.

nbcourbe contient le nombre de courbe (1 par défaut).

Valeur : Aucune.
retournée

TYP DetruireCourbeChn

Fonction : La fonction libère l'espace mémoire réservé pour une courbe chaînée.

Syntaxe : `void TYP_DetruireCourbeChn(CourbeChn* base);`

Arguments : *base* pointe sur la liste des points de la courbe chaînée contrainte-déformation à détruire.

Valeur : Aucune.
retournée

TYP DetruireNoeud

Fonction : La fonction libère l'espace mémoire réservé pour un noeud.

Syntaxe : `void TYP_DetruireNoeud(Noeud* nd, int nbnoeud=1);`

Arguments : *nd* pointe sur le noeud à détruire.

nbnoeud contient le nombre de noeud (1 par défaut).

Valeur : Aucune.
retournée

TYP DetruirePlansNoeud

Fonction : La fonction libère l'espace mémoire d'un noeud qui est réservé pour les plans représentant son état thermomécanique.

Syntaxe : *void TYP_DetruirePlansNoeud(Noeud* nd_in);*

Arguments : *nd_in* pointe sur le noeud pour lequel l'espace mémoire est libéré

Valeur : Aucune.
retournée

TYP DetruireSeqTP

Fonction : La fonction libère l'espace mémoire réservé pour la séquence de chargement.

Syntaxe : *void TYP_DetruireSeqTP(SeqTP* tc);*

Arguments : *tc* pointe sur la séquence de chargement pour laquelle l'espace mémoire est libéré.

Valeur : Aucune.
retournée

TYP EcrireCourbe

Fonction : La fonction écrit les points d'une courbe dans un fichier.

Syntaxe : *void TYP_EcrireCourbe(Courbe* in);*

Arguments : *in* pointe sur la courbe à écrire dans un fichier.

Valeur : Aucune.
retournée

TYP InterpolerPointChnSurCourbeChn

Fonction : La fonction interpole un point sur une courbe chaînée. Le paramètre interpolé est la déformation pour une contrainte et une température connues.

Syntaxe : *double CAL_InterpolerTauPourGamma(CourbeChn* liste, double gamma, PointChn* depart);*

Arguments : *liste* pointe sur la liste des points de la courbe τ - γ .
gamma contient la valeur de γ .
depart pointe sur point à partir duquel l'interpolation débute.

Valeur retournée : La fonction retourne la valeur interpolée de τ .

TYP InterpolerPointSurCourbe

Fonction : La fonction interpole un point sur une courbe.

Syntaxe : *double TYP_InterpolerPointSurCourbe(Courbe* liste, double valeur, int param);*

Arguments : *liste* pointe sur la liste des points de la surface.
param indique le paramètre connu.
 0 = déformation → paramètre interpolé = contrainte.
 1 = contrainte → paramètre interpolé = déformation.
 2 = température → paramètre interpolé = déformation.
valeur contient la valeur de ce paramètre.

Valeur retournée : La fonction retourne la valeur du paramètre interpolé.

TYP InverserCourbe

Fonction : La fonction inverse l'ordre de la liste des points d'une courbe.

Syntaxe : *void TYP_InverserCourbe(Courbe* cbin);*

Arguments : *cbin* pointe la courbe à inverser.

Valeur retournée : Aucune.

TYP LireFichierAmfData

Fonction : La fonction lit, dans un fichier, les données thermomécaniques de l'alliage et les assigne à une variable de type *AmfData* en plus de construire les trois plans principaux.

Syntaxe : *int TYP_LireFichierAmfData(AmfData* AmfD);*

Arguments : *AmfD* pointe sur les données de l'AMF.

Valeur retournée : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.

TYP LireFichierSeqTemp ContDefo

Fonction : La fonction lit dans un fichier et calcul la séquence de chargement qui doit être imposée et l'assigne à une variable de type *SeqTP*.

Syntaxe : *int TYP_LireFichierSeqTemp_ContDefo(SeqTP* sequence);*

Arguments : *sequence* pointe sur la séquence de chargement.

Valeur retournée : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.

TYP LireFichierSeqTemperature

Fonction : La fonction lit dans un fichier et calcul la séquence de chargement en température lorsque l'essai est à contrainte variable et l'assigne à une variable de type *SeqTP*.

Syntaxe : *int TYP_LectureSeqTempL(SeqTP* sequence, double contr);*

Arguments : *contr* contient la valeur de la contrainte dans l'AMF lorsque l'équilibre est atteint entre ce dernier et le contre-ressort.

Sequence pointe sur la séquence de chargement.

Valeur retournée : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.

TYP ModifierNbPlansNoeud

Fonction : La fonction modifie le nombre de plans utilisés pour représenter l'état thermomécanique d'un noeud.

Syntaxe : *int TYP_ModifierNbPlansNoeud(Noeud* in, int deltanbpl);*

Arguments : *in* pointe sur le noeud pour lequel le nombre de plan est modifié.
deltanbpl indique le nombre de plan qui est ajouté (sa valeur peut être négative si les plans doivent être supprimés).

Valeur retournée : La fonction retourne -9 s'il y a une erreur, sinon elle retourne 0 et *in* est modifié.

TYP ModifierSeqTP

Fonction : La fonction réserve l'espace pour les pointeurs aux séquences de chargement d'une variable de type *SeqTP*. Si la variable contient déjà une séquence de chargement, celle-ci est détruite et remplacée par la nouvelle.

Syntaxe : *void TYP_ModifierSeqTP(SeqTP* seq, int param, int nbetape);*

Arguments : *seq* pointe sur la séquence de chargement.

param indique le type de chargement

0 = déformation

1 = contrainte)

nbetape contient le nombre de points de la séquence de chargement.

Valeur retournée : Aucune.

ValAbs

Fonction : La fonction calcule la valeur absolue d'un nombre.

Syntaxe : *double ValAbs(double a);*

Arguments : *a* contient la valeur du nombre à traiter.

Valeur retournée : La fonction retourne la valeur absolue de son argument.

7.2 Loi de comportement (matamf_1.lib)

7.2.1 Fonctions

MAT CalculerPlanLSSCH

Fonction : La fonction calcule les 4 points qui définissent un plan LSSCH (sous-surface de chargement sans hystérésis).

Syntaxe : *void MAT_CalculerPlanLSSCH(const AmfData* AmfD, const Point* pt, Plan* surf);*

Arguments : *AmfD* pointe sur les données de l'AMF.
pt pointe sur le point de départ utilisé pour générer la nouvelle surface.
surf pointe sur la surface générée.

Valeur retournée : Aucune, mais *surf* est modifié.

MAT CalculerPlanLSSCHH

Fonction : La fonction calcule les 4 points qui définissent un plan LSSCHH (sous-surface de chargement avec une possibilité d'hystérésis).

Syntaxe : *void MAT_CalculerPlanLSSCHH(Noeud* ni);*

Arguments : *ni* pointe sur le noeud pour lequel la surface est calculée.

Valeur retournée : Aucune, mais la valeur de *ni* est modifiée.

MAT CalculerPlanLSSDE

Fonction : La fonction calcule les 4 points qui définissent un plan LSSDE (sous-surface de déchargement sans hystérésis).

Syntaxe : *void MAT_CalculerPlanLSSDE(const AmfData* AmfD, const Point* pt, Plan* surf);*

Arguments : *AmfD* pointe sur les données de l'AMF.
pt pointe sur le point de départ utilisé pour générer la nouvelle surface.
surf pointe sur la surface générée.

Valeur : Aucune.
retournée

MAT CalculerPlanLSSDEH

Fonction : La fonction calcule les 4 points qui définissent un plan LSSDEH (sous-surface de déchargement avec une possibilité d'hystérésis).

Syntaxe : `void MAT_CalculerPlanLSSDEH(const AmfData* LAmfD, Noeud* ni);`

Arguments : *AmfD* pointe sur les données de l'AMF.
ni pointe sur le noeud pour lequel la surface est calculée.

Valeur : Aucune, mais la valeur de *ni* est modifiée.
retournée

MAT CalculerReponseUnNoeud

Fonction : La fonction calcule la réponse d'un noeud soumis à une variation de température et de contrainte ou de déformation.

Syntaxe : `int MAT_CalculerReponseUnNoeud(const AmfData* AmfD, Noeud* ndi, double DeltaTemp, double DeltaParam, int NumParam);`

Arguments : *AmfD* pointe sur les données de l'AMF.
ndi pointe sur le noeud pour lequel la surface est calculée.
DeltaTemp contient la valeur de l'incrément (ou du décrétement) de température.
NumParam indique le paramètre modifié.
0 = déformation
1 = contrainte
DeltaParam contient la valeur de l'incrément pour ce paramètre.

Valeur : La fonction retourne -9 si il y a une erreur, sinon elle retourne 0.
retournée

MAT InterpolerPointSurDroite

Fonction : La fonction interpole un point sur une section droite de l'espace située entre deux points connus.

Syntaxe : *int MAT_InterpolerPointSurDroite(const Point* pt1, const Point* pt2, int NoObs, double Obs, Point* retour);*

Arguments : *pt1* et *pt2* contiennent les points entre lesquels l'interpolation linéaire est effectuée.

NoObs indique paramètre connu.

0 = déformation

1 = contrainte

2 = température

Obs contient la valeur de ce paramètre.

retour pointe sur le point interpolé.

Valeur retournée : La fonction retourne 1 si le point interpolé est supérieur à *pt2* et elle retourne -1 s'il est inférieur à *pt1*. Dans les deux cas, aucune valeur n'est attribuée à *retour*. Si l'interpolation est réussie, la fonction retourne 0 et le point interpolé est affecté à *retour*.

MAT InterpolerPointSurPlan

Fonction : La fonction interpole un point sur un plan pour une température et soit une contrainte ou une déformation connue.

Syntaxe : *int MAT_InterpolerPointSurSurface(const Plan* surf, double temp, int NoObs, double Obs, Point* retour);*

Arguments : *surf* pointe sur la surface contrainte-déformation-température sur laquelle l'interpolation a lieu.

temp contient la valeur de la température du point interpolé.

NoObs indique le paramètre connu.

0 = déformation

1 = contrainte

Obs contient la valeur de ce paramètre.

retour pointe sur le point interpolé.

Valeur retournée : La fonction retourne 1 si le résultat de l'interpolation dépasse la surface vers le haut et elle retourne -1 si ce résultat dépasse la surface vers le bas, sinon

elle retourne 0 et le point interpolé est affecté à *retour*.

7.3 Chargement simple (calamf_1.lib)

7.3.1 Fonctions

CAL_AfficherCourbeCharge3D

Fonction : La fonction affiche la courbe *contrainte-déformation-température* à l'aide du logiciel GNUPLOT (ce dernier doit se situer dans le répertoire courant).

Syntaxe : `int CAL_AfficheCourbeCharge(double* Temp, double* contr, double* defo, AmfData* AMF, int NbPoints);`

Arguments : *Temp* pointe sur la séquence de température.

contr pointe sur la séquence de contrainte.

defo pointe sur la séquence de déformation.

AMF pointe sur les données de l'AMF (pour construire les plans).

NbPoints contient le nombre de points de la courbe.

Valeur retournée : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.

CAL_AfficherCourbesCharge2D

Fonction : La fonction affiche la projection de la courbe *contrainte-déformation-température* sur les plans *courbe contrainte-déformation*, *courbe contrainte-température* et *courbe déformation-température* à l'aide du logiciel GNUPLOT (ce dernier doit se situer dans le répertoire courant).

Syntaxe : `int CAL_AfficherCourbesCharge2D(double* Temp, double* contr, double* defo, int NbPoints);`

Arguments : *Temp* pointe sur la séquence de température.

contr pointe sur la séquence de contrainte.

defo pointe sur la séquence de déformation.

NbPoints contient le nombre de points de la courbe.

Valeur retournée : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.

CAL CalculerCourbeTauGamma

Fonction : La fonction calcule la courbe τ - γ d'une tige cylindrique soumise à une séquence de chargement en torsion.

Syntaxe : *int CAL_CalculerTorsionFil(const AmfData* AmfDcis, const double r, Noeud* ndelem, int nbel, CourbeChn* Rep);*

Arguments : *AmfDcis* pointe sur les données de l'AMF.

r contient la valeur du rayon de la tige.

ndelem pointe sur les noeuds auxquels la séquence de chargement est imposée.

nbel contient le nombre d'éléments utilisés pour la discrétisation.

Rep pointe sur la liste des points qui représentent la courbe τ - γ de l'alliage.

Valeur retournée : La fonction retourne -1 s'il y a une erreur dans la valeur de la déflexion et -9 s'il y a une erreur lors de l'imposition du chargement, sinon elle retourne 0.

CAL CalculerDefoContrCte

Fonction : La fonction calcule, pour un seul noeud, la déformation associée à chaque point de chargement pour un essai à contrainte imposée constante (avec ou sans butée).

Syntaxe : *int CAL_CalculerDefoContrCte(double* defo, TypeDePlan* surface, SeqTP* SequenceTC, const AmfData* AmfDcis, const double butee, const double contr);*

Arguments : *defo* pointe sur les déformations associées à chaque point de chargement.

surface pointe sur les surfaces associées à chaque point de chargement.

SequenceTC pointe sur la séquence de chargement.

AmfDcis pointe sur les données de l'AMF.

butee contient la valeur de la déformation de l'AMF lorsque ce dernier entre en contact avec la butée (*butee* = 0 s'il n'y a pas de butée).

contr contient la valeur de la contrainte à laquelle l'essai est effectué.

Valeur retournée : La fonction retourne -9 si il y a une erreur, sinon elle retourne 0.

CAL CalculerDefoContrNCte

Fonction : La fonction calcule, pour un seul noeud, la déformation associée à chaque point du chargement pour un essai à contrainte imposée variable, c'est-à-dire avec un contre-ressort (avec ou sans butée).

Syntaxe : *int CAL_CalculerDefoContrNCte(double* defo, TypeDePlan* surface, SeqTP* SequenceTC, const AmfData* AmfDcis, const double butee, const Courbe* ressort, const GeomFil* geom);*

Arguments : *defo* pointe sur les déformations associées à chaque point de chargement.
surface pointe sur les surfaces associées à chaque point de chargement.
SequenceTC pointe sur la séquence de chargement.
AmfDcis pointe sur les données de l'AMF.
butee contient la valeur de la déformation de l'AMF lorsque ce dernier entre en contact avec la butée (*butee* = 0 s'il n'y a pas de butée).
ressort pointe sur la liste des points déflexion-force qui caractérisent le contre-ressort.
geom pointe sur la géométrie de l'échantillon.

Valeur retournée : La fonction retourne -9 si il y a une erreur, sinon elle retourne 0..

CAL CalculerMTorsionResultantDeTau

Fonction : La fonction calcule le moment de torsion résultant de la répartition non-linéaire du cisaillement sur le rayon d'une tige en torsion.

Syntaxe : *double CAL_CalculerMTorsionResultantDeTau(const Noeud* elem, int nbed, const double r);*

Arguments : *elem* pointe sur les noeuds des éléments pour lesquels le moment est calculé.

nbed contient le nombre d'éléments.

r contient la valeur du rayon de la tige.

Valeur retournée : La fonction retourne la valeur du moment de torsion.

CAL CalculerMTorsionSurAnneau

Fonction : La fonction calcule le moment de torsion sur un anneau de la section d'une tige cylindrique.

Syntaxe : *double CAL_CalculerMTorsionSurAnneau(double r1, double r2, double tau1, double tau2);*

Arguments : *r1* contient la valeur du rayon intérieur de la section circulaire.

r2 contient la valeur du rayon extérieur de la section circulaire.

tau1 contient la valeur de la contrainte de cisaillement à *r1*.

tau2 contient la valeur de la contrainte de cisaillement à *r2*.

Valeur retournée : La fonction retourne la valeur du moment sur l'anneau.

CAL CalculerTauSurRayonExt

Fonction : La fonction calcule la contrainte de cisaillement sur le rayon extérieur d'une tige soumise à de la torsion en supposant une distribution linéaire du cisaillement sur la section du rayon.

Syntaxe : *double CAL_CalculerTauSurRayonExt(double r1, double r, double tau1, double deltm);*

Arguments : *r1* contient une valeur intermédiaire du rayon.

r contient la valeur du rayon extérieur.

tau1 contient la valeur de la contrainte de cisaillement à *r1*.

deltM contient la variation du moment entre *r1* et *r*.

Valeur : La fonction retourne la valeur de la contrainte de cisaillement.
retournée

CAL EcrireCress

Fonction : La fonction écrit, dans un fichier, les points déflexion-force qui caractérisent le comportement d'un contre-ressort.

Syntaxe : *int CAL_EcrireCress(Courbe* fdex);*

Arguments : *fdex* pointe sur la liste des points force-déflexion qui caractérisent le comportement du contre-ressort.

Valeur : La fonction retourne 0 s'il y a une erreur, sinon elle retourne 1.
retournée

CAL EcrireDistributionTauGammaSurRayon

Fonction : La fonction écrit les points τ - γ - T des noeuds sur la section d'une tige en torsion pour un état de sollicitation donné.

Syntaxe : *int CAL_EcrireCourbeTauGammaTRayon(const double r_ext, const Noeud* ndelem, int nbed);*

Arguments : *r_ext* contient la valeur du rayon extérieur de la tige.

ndelem pointe sur les noeuds de la tige.

nbed contient le nombre d'éléments utilisés pour la discrétisation.

Valeur : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.
retournée

CAL EstimerCourbeTGpourMA

Fonction : La fonction estime la courbe τ - γ à partir de la courbe moment-angle par un calcul inverse.

Syntaxe : *CourbeChn* CAL_EstimerCourbeTGpourMA(Courbe* dataMA, double r, int nbed);*

Arguments : *dataMA* pointe sur la liste des points moment-angle.

r contient la valeur du rayon de la tige.

nbed est le nombre d'éléments discrétisés.

Valeur retournée : La fonction retourne un pointeur à la liste des points de la courbe tau-gamma

CAL_ImposerDeltaAngleDeltaT

Fonction : La fonction calcule la réponse d'une tige AMF soumise à un incrément de l'angle de torsion et de la température.

Syntaxe : *int CAL_ImposerDeltaAngleDeltaT(const AmfData* AmfDcis,*
const double r, Noeud ndelem, int nbel,*
double deltanglunit, double DelT);

Arguments : *AmfDcis* pointe sur les données de l'AMF.

r contient la valeur du rayon de la tige.

ndelem pointe sur les noeuds des éléments auxquels l'incrément du moment et de la température sont imposés.

nbel contient le nombre d'éléments utilisés pour la discrétisation

deltanglunit contient la valeur de l'incrément de l'angle unitaire (rad/mm)

DelT contient la valeur de l'incrément de température.

Valeur retournée : La fonction retourne -9 s'il y a une erreur, sinon elle retourne 0. Chaque noeud de *ndelem* contient le nouveau point *moment-angle-température* représentant son nouvel état thermodynamique.

CAL_ImposerDeltaMTorsionDeltaT

Fonction : La fonction calcule la réponse d'une tige AMF soumise à un incrément du moment de torsion et de la température.

Syntaxe : *int CAL_ImposerDeltaMTorsionDeltaT(const AmfData* AmfDcis,*
const double r, Noeud ndelem, int nbel, double deltmoment,*

double pente_init, double DelT);

Arguments : *AmfDcis* pointe sur les données de l'AMF.

r contient la valeur du rayon de la tige.

ndelem pointe sur les noeuds des éléments auxquels l'incrément du moment et de la température sont imposés.

nbel contient le nombre d'éléments utilisés pour la discrétisation

deltmoment contient la valeur de l'incrément du moment

pente_init contient la valeur de la pente moment/angle_unitaire (rad/mm) initiale.

DelT contient la valeur de l'incrément de température.

Valeur retournée : La fonction retourne -9 s'il y a une erreur, sinon elle retourne 0. Chaque noeud de *ndelem* contient le nouveau point *moment-angle-température* représentant son nouvel état thermodynamique.

CAL LireCress

Fonction : La fonction lit les points déflexion-force qui caractérisent le comportement d'un contre-ressort par le biais d'un dialogue avec l'utilisateur.

Syntaxe : *int CAL_LireCress(Courbe* fdex);*

Arguments : *fdex* pointe sur la liste des points force-déflexion qui caractérisent le comportement du contre-ressort.

Valeur retournée : La fonction retourne 0 s'il y a une erreur, sinon elle retourne 1.

CAL LireCteCress

Fonction : La fonction lit la constante de rigidité d'un contre-ressort linéaire par le biais d'un dialogue avec l'utilisateur.

Syntaxe : *void CAL_LireCteCress(double* k);*

Arguments : *k* pointe sur la valeur de la constante de rigidité du ressort.

Valeur retournée : Aucune.

CAL LireFichierCress

Fonction : La fonction lit, dans un fichier, les points déflexion-force qui caractérisent le comportement d'un contre-ressort.

Syntaxe : *int CAL_LireFichierCress(Courbe* fdex);*

Arguments : *fdex* pointe sur la courbe représentant le comportement du contre-ressort.

Valeur retournée : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.

CAL TrouverPtEquiCressUnNoeud

Fonction : La fonction calcule la contrainte générée dans l'AMF lorsque ce dernier est en équilibre avec un contre-ressort.

Syntaxe : *double CAL_TrouverPtEquiCressUnNoeud(double temp, GeomFil* geom, AmfData* AmfDcis, Courbe* ressort);*

Arguments : *temp* contient la valeur de la température initiale de l'essai.

geom pointe sur la géométrie de l'échantillon.

AmfDcis pointe sur les données de l'AMF.

ressort pointe sur les points déflexion-force caractérisant le contre-ressort.

Valeur retournée : La fonction retourne la valeur de la contrainte.

7.4 Échanges thermiques (thramf_1.lib)

7.4.1 Types

Coefficient

Le type *Coefficient* est utilisé pour les variables qui contiennent les coefficients de l'équation quadratique, représentant l'évolution de la chaleur spécifique en fonction de la température (au chauffage et au refroidissement), utilisés pour modéliser la transformation de phase.

Déclaration :

```
typedef struct f_Coefficient
{
double A;          // coef. de l'équation cubique - contrainte variable
double B;          // coef. de l'équation cubique - contrainte variable
double D;          // coef. de l'équation cubique - contrainte variable
double AdscC;     // coef. de l'équation cubique - DSC chauffage
double BdscC;     // coef. de l'équation cubique - DSC chauffage
double DdscC;     // coef. de l'équation cubique - DSC chauffage
double AdscR;     // coef. de l'équation cubique - DSC refroid.
double BdscR;     // coef. de l'équation cubique - DSC refroid.
double DdscR;     // coef. de l'équation cubique - DSC refroid.
} Coefficient;
```

CtePhysique

Le type *CtePhysique* est utilisé pour les variables qui contiennent les propriétés physiques de l'AMF.

Déclaration :

```
typedef struct f_CtePhysique
{
double As;          // température de début de transformation inverse
double Af;          // température de fin de transformation inverse
double Ms;          // température de début de transformation directe
double Mf;          // température de fin de transformation directe
double rho_elec;    // résistivité électrique
double rho;         // masse volumique
double Cp;          // chaleur spécifique
double hyst_c;      // hystérésis en température au chauffage
double hyst_r;      // hystérésis en température au refroid.
} CtePhysique;
```

GeomFil

Le type *GeomFil* est utilisé pour les variables qui contiennent les données géométriques de l'actuateur AMF.

Déclaration :

```
typedef struct f_GeomFil
{
    double d;        // diamètre du fil
    double l;        // longueur du fil
    double A;        // aire latérale du fil
    double V;        // volume du fil
} GeomFil;
```

Parametre

Le type *Parametre* est utilisé pour les variables qui contiennent les paramètres d'un essai d'échauffement par effet Joule d'un fil droit AMF.

Déclaration :

```
typedef struct f_Parametre
{
    double resist;   // résistance électrique
    double perte;   // pertes thermiques
    double i;        // courant électrique
    double h;        // coefficient de convection
    double delta_t; // incrément de temps
    double butee;   // défo. de l'AMF quand il entre en contact avec la butée
    double cor;     // Coefficient de correction pour la conduction
} Parametre;
```

7.4.2 Fonctions

THR AfficherCourbeDefoTemps

Fonction : La fonction affiche la courbe *déformation-temps* à l'aide du logiciel GNUPLOT (ce dernier doit se situer dans le répertoire courant).

Syntaxe : `int THR_AfficherCourbeDefoTemps(double* defo, double* temps, int NbPoints);`

Arguments : *defo* pointe sur la séquence de déformation.

temps pointe sur la séquence de temps.

NbPoints contient le nombre de points de la courbe.

Valeur retournée : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.

THR CalculerEnergie

Fonction : La fonction calcule la perte d'énergie sur un intervalle de temps lors de la transformation de phase.

Syntaxe : *double THR_CalculerEnergie(double *temps2, double energie, Parametre *param, double *t);*

Arguments : *temps2* pointe sur le temps corrigé lorsque l'énergie de transformation devient nulle.

energie contient la valeur de l'énergie de transformation restante.

param pointe sur les paramètres de l'expérience.

t pointe sur le temps calculé.

Valeur retournée : La fonction retourne la nouvelle valeur de l'énergie.

THR CalculerSeqTemps

Fonction : La fonction calcule la séquence totale de chargement (contrainte et température) appliquée à l'échantillon quand la séquence est spécifiée par un temps d'application du courant.

Syntaxe : *void THR_CalculerSeqTemps(double contr, double *temperature, int sizeoftemp, int sizeofcharge, SeqTP* sequence);*

Arguments : *contr* contient la valeur de la contrainte à laquelle l'essai est effectué.

temperature pointe sur la séquence de température.

sizeoftemp contient la grandeur de la séquence de température.

sizeofcharge contient la grandeur de la séquence de contrainte.

sequence pointe sur la séquence de chargement.

Valeur : Aucune.
retournée

THR CalculerTemps

Fonction : La fonction calcule le temps requis pour réaliser la séquence de chargement en température.

Syntaxe : *double* THR_CalculerTemps(SeqTP* SequenceTC, TypeDePlan* surface, Parametre* param, CtePhysique* amf, GeomFil* geom, Coefficient* coef, AmfData* AmfDcis);*

Arguments : *SequenceTC* pointe sur la séquence de chargement.
surface pointe sur la liste des surfaces associées à chaque point de chargement.
param pointe sur les paramètres de l'expérience.
amf pointe sur les propriétés de l'AMF.
geom pointe sur la géométrie de l'échantillon.
coef pointe sur les Coefficients de l'équation cubique.
AmfDcis pointe sur les données de l'AMF.

Valeur : La fonction retourne un pointeur à la liste contenant le temps calculé pour
retournée chaque point de chargement

THR ConvertirSeqTemperatureEnSeqTemps

Fonction : La fonction calcule la séquence de température à imposer à l'AMF à partir du temps des échanges thermiques. Dans ce cas-ci, contrairement au cas où la séquence de température est spécifiée, il ne peut y avoir qu'un seul cycle.

Syntaxe : *int THR_ConvertirSeqTemperatureEnSeqTemps(Parametre *param, CtePhysique *amf, GeomFil* geom, Coefficient *coef, double *temps, double* temperature, double T8, double eps, double chauF, double refroid, int sizeofcharge);*

Arguments : *param* pointe sur les paramètres de l'expérience.

amf pointe sur les propriétés de l'AMF.

geom pointe sur la géométrie de l'échantillon.

coef pointe sur les coefficients de l'équation cubique.

temps pointe sur la liste du temps associé à chaque point de chargement.

temperature pointe sur la liste de la température associée à chaque point de chargement.

T8 contient la valeur de la température ambiante.

eps contient la valeur de la tolérance utilisée pour la résolution de l'équation cubique par la méthode de Newton.

chauf contient la valeur du temps d'échauffement.

refroid contient la valeur du temps de refroidissement.

sizeofcharge contient la valeur de la grandeur du tableau des contraintes.

Valeur retournée : La fonction retourne le nombre total de points de chargement.

THR_EcritureFichierTCDt

Fonction : La fonction écrit, dans un fichier, les points température-contrainte-déformation-temps qui caractérisent la réponse d'un fil AMF soumis à un chargement thermomécanique.

Syntaxe : *int THR_EcritureFichierTCDt(double* Temp, double* contr, double* defo, double* temps);*

Arguments : *Temp* pointe sur la séquence de température.

contr pointe sur la séquence de contrainte.

defo pointe sur la séquence de déformation.

temps pointe sur la séquence de temps.

Valeur retournée : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.

THR LireFichierDataThermique

Fonction : La fonction lit, dans un fichier, les données sur les paramètres de l'expérience et les assigne à une variable de type *Parametre*.

Syntaxe : *int THR_LireFichierDataThermique(CtePhysique* amf, GeomFil* geom, Parametre* param, Coefficient* coef);*

Arguments : *amf* pointe sur les propriétés de l'AMF.
geom pointe sur la géométrie de l'échantillon.
param pointe sur les paramètres de l'expérience.
coef pointe sur les coefficients de l'équation de C_p .

Valeur retournée : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.

THR LireFichierSeqTemps

Fonction : La fonction lit un fichier contenant les données de chargement lorsque le temps des échanges thermiques est imposé.

Syntaxe : *int THR_LireFichierSeqTemps(Parametre *param, CtePhysique *amf, double *T8, double *contr, double *eps, double *chauf, double *refroid, int *sizeoftemp, double dsdt);*

Arguments : *param* pointe sur les paramètres de l'expérience.
amf pointe sur les propriétés de l'AMF.
T8 contient la valeur de la température ambiante.
contr contient la valeur de la contrainte à laquelle l'essai est effectué.
eps contient la valeur de la tolérance utilisée pour la résolution de l'équation cubique par la méthode de Newton.
chauf contient la valeur du temps d'échauffement.
refroid contient la valeur du temps de refroidissement.
sizeoftemp pointe sur la grandeur du tableau contenant la séquence de température.
dsdt contient la valeur de la pente contrainte/température.

Valeur retournée : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.

THR_ResoudreEquaEchTPhase

Fonction : La fonction résous, par la méthode de Newton, l'équation du troisième degré utilisée dans le calcul du temps lors de la transformation de phase.

Syntaxe : *double THR_ResoudreEquaEchTPhase(Coefficient* coef, Parametre* param, double eps, double *Te, double V, double rho, double deltaTrans, double hyst, double Tfintrans, double fraction);*

Arguments : *coef* pointe sur les Coefficients de l'équation cubique.

param pointe sur les paramètres de l'expérience.

eps contient la valeur de la tolérance utilisée pour la résolution de l'équation cubique par la méthode de Newton.

Te pointe sur les deux dernières températures calculées.

V contient la valeur du volume de l'échantillon.

rho contient la valeur de la masse volumique de l'échantillon.

deltaTrans contient la valeur de l'hystérésis en température du modèle.

hyst contient la valeur théorique de l'hystérésis en température.

Tfintrans contient la valeur de la température finale de transformation.

fraction contient la valeur de la fraction d'énergie disponible pour la transformation de phase.

Valeur retournée : La fonction retourne la température calculée en fonction du temps et des conditions de l'essai.

7.5 Ressorts AMF hélicoïdaux.

7.5.1 Types

Ressort

Le type *Ressort* est utilisé pour les variables qui contiennent les données du ressort.

Déclaration :

```
typedef struct f_Ressort
{
  char ident[100];      // identification du ressort
  double temp;         // C
  double D;            // mm diamètre moyen
  double d;            // mm diamètre du fil
  double n;            // nombre de spires
  double h0;           // hauteur austénitique charge nulle
  double Mf;           // C en cas d'omission -400
  double Ms;           // C en cas d'omission -400
  double As;           // C en cas d'omission -400
  double Af;           // C en cas d'omission -400
  Courbe* Ch;          // Courbe Tau Gamma de l'amf pour le chargement
  Courbe* De;          // Courbe Tau Gamma de l'amf pour le déchargement
} Ressort;
```

7.5.2 Fonctions

RES ConvertirAngleEnDeflex

Fonction : La fonction transforme les données d'angle (rad/mm) d'un ressort en données de déflexion.

Syntaxe : *double RES_ConvertirAngleEnDeflex(const Ressort* resso, double ddAU);*

Arguments : *resso* pointe sur les données du ressort.

ddAU contient la valeur de l'angle unitaire (rad/mm).

Valeur retournée : La fonction retourne la valeur de la déflexion.

RES ConvertirCourbeRessortFcteEnTcte

Fonction : La fonction transforme une courbe *contrainte-déformation* à force constante en une courbe *contrainte-déformation* à température constante. Cette fonction est encore au stade expérimental et elle fonctionne plus ou moins bien avec les TiNi car elle considère les modules d'élasticité comme étant constants.

Syntaxe : *void RES_ConvertirCourbeRessortFcteEnTcte(Courbe* cin, double Kressaust, double dFdT, double Tref);*

Arguments : *cin* pointe sur la liste des points de la surface force-déflexion-température.

Kressaust contient la valeur de la constante du ressort en phase austénitique.

dFdT contient la valeur de la pente force-température.

Tref contient la valeur de la température de référence.

Valeur : Aucune, mais *cin* est modifié.
retournée

RES_ConvertirDeflexEnAngle

Fonction : La fonction transforme les données de déflexion d'un ressort en données d'angle (rad/mm).

Syntaxe : *double RES_ConvertirDeflexEnAngle(const Ressort* resso, double ddeflex);*

Arguments : *resso* pointe sur les données du ressort.

ddeflex contient la valeur de la déflexion.

Valeur : La fonction retourne la valeur de l'angle (rad/mm).
retournée

RES_ConvertirFDenMA

Fonction : La fonction transforme les données de force-déplacement d'un ressort en données de moment-angle (N·m·rad/mm).

Syntaxe : *void RES_ConvertirFDenMA(Ressort* rin);*

Arguments : *rin* pointe sur le ressort à être modifié.

Valeur : Aucune, mais le contenu de *rin* est modifié.
retournée

RES_ConvertirForceEnMoment

Fonction : La fonction converti une force en moment.

Syntaxe : *double CAL_ConvertirForceEnMoment(double R, double Force);*

Arguments : *R* contient la valeur du rayon.

Force contient la valeur de la force

Valeur retournée : La fonction retourne la valeur du moment.

RES_ConvertirMomentEnForce

Fonction : La fonction converti un moment en force

Syntaxe : *double CAL_ConvertirMomentEnForce(double R, double Moment);*

Arguments : *R* contient la valeur du rayon.

Moment contient la valeur du moment.

Valeur retournée : La fonction retourne la valeur de la force.

RES_CreerRessort

Fonction : La fonction réserve l'espace mémoire pour un ressort et initialise ses paramètres à 0. Cependant, le nombre de points des courbes de chargement et de déchargement n'étant pas connus lors de la création de la variable, la valeur *néant* (NULL) est attribuée aux deux pointeurs de types *Courbe** qui pointent sur les courbes de chargement et de déchargement du ressort.

Syntaxe : *Ressort* RES_CreerRessort(void);*

Arguments : Aucun.

Valeur retournée : La fonction retourne un pointeur à un ressort.

RES_DetruireRessort

Fonction : La fonction libère l'espace mémoire réservé pour un ressort, incluant celui des deux courbes de type *Courbe*.

Syntaxe : *void RES_DetruireRessort(Ressort* in);*

Arguments : *in* pointe sur le ressort à détruire.

Valeur : Aucune.
retournée

RES EcrireCourbeDFT

Fonction : La fonction écrit les points de la courbe déflexion-force-température d'un ressort dans un fichier.

Syntaxe : *int RES_EcrireCourbeDFT(Ressort* ress, CourbeChn* Rep);*

Arguments : *ress* pointe sur le ressort.

Rep pointe sur la liste des points de la courbe.

Valeur : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.
retournée

RES EcrireCourbeTGpourRessort

Fonction : La fonction écrit les données d'un ressort ainsi que les points de la courbe contrainte-déformation de l'alliage dans un fichier.

Syntaxe : *int RES_EcrireCourbeTGpourRessort(Ressort* out, CourbeChn* ch, CourbeChn* de);*

Arguments : *out* pointe sur le ressort qui a été modélisé.

ch pointe sur la liste des points de la courbe contrainte-déformation lors du chargement.

de pointe sur la liste des points de la courbe contrainte-déformation lors du déchargement.

Valeur : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.
retournée

RES LireDataRessort

Fonction : La fonction lit les données géométriques du ressort par le biais d'un dialogue avec l'utilisateur.

Syntaxe : *int RES_LireDataRessort(Ressort* in);*

Arguments : *in* pointe sur les données du ressort.

Valeur : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.
retournée

RES LireFichierRessort

Fonction : La fonction lit un fichier contenant les données mécaniques et géométriques d'un ressort et les assigne à une variable de type *Ressort*.

Syntaxe : *int RES_LectureFichierIntrant(Ressort* rin);*

Arguments : *rin* pointe sur le ressort.

Valeur : La fonction retourne 0 si il y a une erreur, sinon elle retourne 1.
retournée

RES ModifierRessortCh

Fonction : La fonction réserve réserve l'espace mémoire pour la courbe de chargement d'un ressort.

Syntaxe : *void RES_ModifierRessortCh(Ressort* res, int nbpoints);*

Arguments : *res* pointe sur le ressort.

nbpoints contient le nombre de points de la courbe.

Valeur : Aucune.
retournée

RES ModifierRessortDe

Fonction : La fonction réserve l'espace mémoire pour la courbe de déchargement d'un ressort.

Syntaxe : *void RES_ModifierRessortDe(Ressort* res, int nbpoints);*

Arguments : *res* pointe sur le ressort.

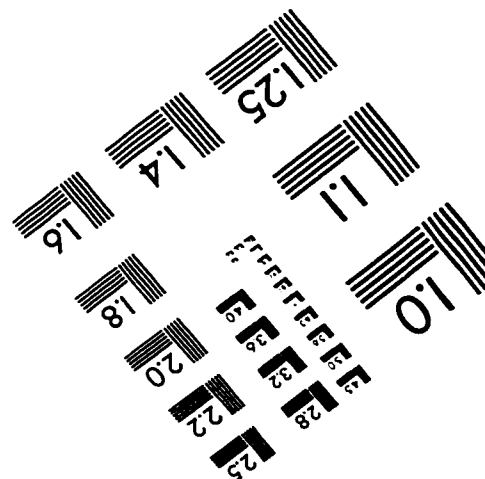
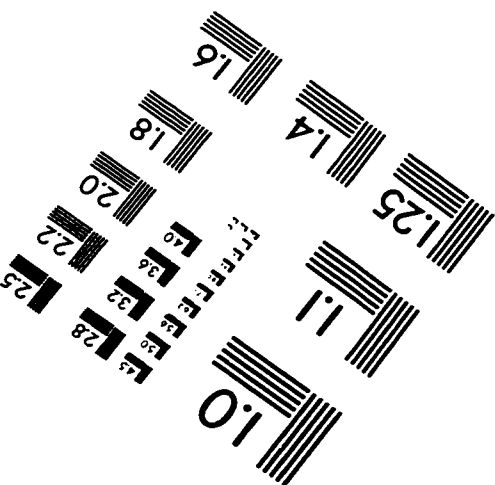
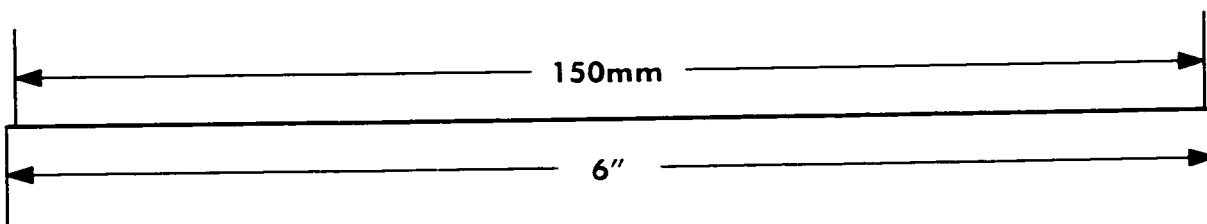
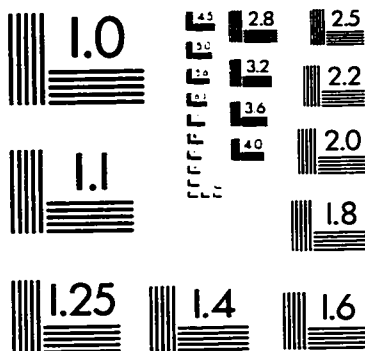
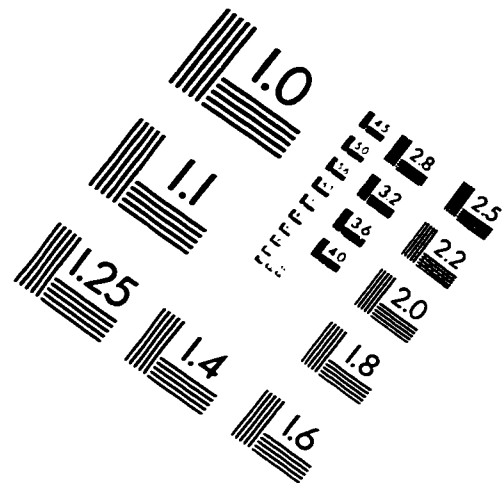
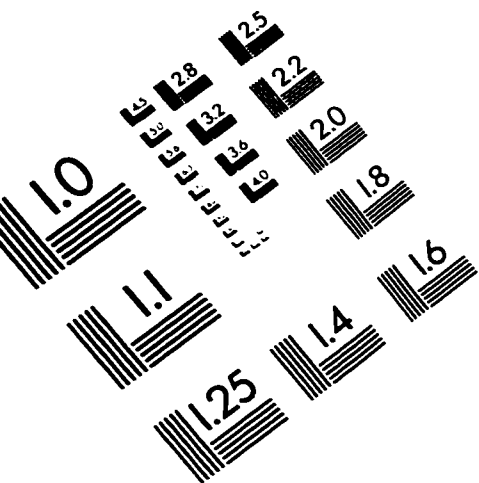
nbpoints contient le nombre de points de la courbe.

Valeur : Aucune.
retournée

Références

- [1] MEUNIER M.-A., Caractérisation expérimentale, modélisation thermomécanique et étude de la fatigue de pièces en alliages à mémoire de (AMF), Mémoire de maîtrise, École Polytechnique de Montréal, 1995, p.194
- [2] MEUNIER M.-A., Caractérisation expérimentale, modélisation thermomécanique et étude de la fatigue de pièces en alliages à mémoire de (AMF), Mémoire de maîtrise, École Polytechnique de Montréal, 1995, p.187
- [3] MEUNIER M.-A., Caractérisation expérimentale, modélisation thermomécanique et étude de la fatigue de pièces en alliages à mémoire de (AMF), Mémoire de maîtrise, École Polytechnique de Montréal, 1995, p.183
- [4] MEUNIER M.-A., Caractérisation expérimentale, modélisation thermomécanique et étude de la fatigue de pièces en alliages à mémoire de (AMF), Mémoire de maîtrise, École Polytechnique de Montréal, 1995, p.179

IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc
 1653 East Main Street
 Rochester, NY 14609 USA
 Phone: 716/482-0300
 Fax: 716/288-5989

© 1993, Applied Image, Inc.. All Rights Reserved