

TABLE OF CONTENTS:

ACKNOWLEDGEMENT	3
INTRODUCTION	4
CHAPTER I STRESS TENSOR DERIVED FROM THE TEXTURE	6
CHAPTER II THE STRESS-STRAIN INTERPOLATION	10
A Introduction	10
B N-dimensional space geometry	10
1 Domain limit vectors	10
2 Definition of a vector basis into the domain	11
3 Use of the new basis	11
4 Intrinsic coordinates	12
5 Properties of those coordinates	12
6 Computation of the S vectors	12
7 Entering the adjacent domain	13
8 2-D and 3-D cases	16
9 Non-regular domains	16
C Yield locus interpolation	17
1 Linear interpolation in cartesian coordinates (hyperplane method)	18
2 Linear interpolation in spherical coordinates (stress-strain interpolation)	19
3 Inverse interpolation	19
4 'Bubble' mode	20
5 Second order interpolation	20
6 Continuity between adjacent domains	22
D Fitting of a yield locus formulation	22
1 Plane formulation	23
2 Spherical part yield locus	23
3 Hill yield locus	23
4 Tsai criterion	25
E Yield locus normal	25
F Application of the interpolation to the stress integration	28
1 Work hardening	29
2 Tangent stiffness matrix	31
3 Stress integration	32
a. Explicit formulation	32
b. Explicit formulation with radial correction	33
c. Implicit formulation	33
G Implementation into the finite element code	34
1 INTERT	35
2 SEARCHU	38
3 MINTY3	40
CHAPTER III THE HYPERPLANE MODEL	43

CHAPTER IV	VALIDATION	45
A	Introduction	45
B	Tensile test along Y-axis	45
C	π-sections of the yield locus	47
D	Simulation on one finite element	53
E	Tensile test of a holed sheet	54
CONCLUSIONS		56
REFERENCES		58

Acknowledgement

The author would like to thank the Région Wallonne and the steel research centre RDCS for their support. Anne Marie Habraken and André Godinas are also acknowledged for their help in the development of the presented model.

Finally, the research team of professor Paul Van Houtte is acknowledged for providing us the texture analysis modules.

Introduction

The goal of this research is to integrate the influence of the material texture into a finite element code. The constitutive law describing the mechanical behaviour of the studied sample is based on a microscopic approach. The law computation takes place on the crystallographic level. A large number of crystals must be used to represent correctly the global behaviour. The micro-macro transition links the global behaviour to the crystallographic results. The Taylor's model described in Chapter I is used for the computation of the microscopic behaviour of each crystal and for the micro-macro transition. Unfortunately, this model does not lead to a general law with a mathematical formulation of the yield locus. Only one point of the yield locus corresponding to a particular strain rate direction can be computed. The "direct Taylor's model" assumes that one macroscopic stress results from the average of the microscopic stresses related to each crystal belonging to a set of representative crystals.

The computation of the mechanical behaviour involves a large number of crystals and must be repeated for each integration point of the finite element model, for each iteration of each time step. So, such a micro-macro approach consumes large computation time and seems practically not useful.

However, using different simplified approaches, various constitutive laws based on texture analysis have been implemented in the non linear finite element code LAGAMINE. Our first step in the integration of the texture effects has been the use of a 6th order series yield locus defined by a least square fitting on a large number of points (typically 70300) in the deviatoric stress space (see Ref. [13]). Those points were calculated by the Taylor's model based on an assumed constant texture of the material. This fitting is performed once, outside the FEM code. It provides 210 coefficients to describe the whole yield locus. This method, i.e. a global description of the yield locus, is actually used in the FEM code and will be referred to as the law ANI3VH.

Unfortunately, taking into account the texture evolution effects with this yield locus would imply the computation of the 210 coefficients of the 6th order series for each integration point, each time a texture updating is necessary. This would require an impressive amount of computation and memory storage (210 coefficients for each integration point) which is only partially useful as generally the stress state remains in a local zone of the yield locus. So, two new approaches, where the whole yield locus is unknown, have been investigated.

In the first case, some points in the interesting part of the yield locus are computed with the Taylor's model. This local zone of the yield locus is then represented by a set of hyperplanes which are planes defined in the five-dimensional deviatoric stress space. These planes being fitted on the Taylor's points.

As it has been shown in [4], the yield locus discontinuities bred by this very simple interpolation method give rise to convergence problems in the finite element code. That is the reason why a second method has been developed.

For that second approach, no yield locus is defined and a direct stress-strain interpolation between Taylor's points is achieved. In this case, the yield stress continuity conditions are fulfilled but, as there is no yield locus formulation, a particular stress integration scheme has to be used.

Both interpolation methods allow us an important computation time reduction with respect to the "direct Taylor's model" application. The Taylor's model is only used to compute some points in order to achieve the interpolation.

These points must be computed in two cases:

- When the current part of the yield locus does not content anymore the new stress state and that a new local zone of the yield locus is required.
- When the plastic strains significantly deform the material and induce changes in the crystallographic orientations, i.e. when the texture evolves. Indeed, the corresponding mechanical behaviour of the material would no more be correctly represented by the old points. A texture updating must take place.

The part yield locus approach presented in this work can be placed between the microscopic approach (accurate but very slow) and the global yield locus approach (fast but inaccurate and especially not adapted for texture updating).

The interpolation methods (hyperplane or stress-strain interpolation) and the precision of their integration scheme in the finite element code are very important points and are presented hereafter.

The work contents is the following one:

The first chapter presents the main particular points of the Taylor's model and the micro-macro transition: the method used to compute the yield limit in the deviatoric stress space for a particular prescribed strain rate. This is in fact one point of the yield locus.

The second chapter largely describes the stress-strain interpolation and the underlying theoretical formulations.

The third chapter is devoted to the hyperplane method while the fourth chapter compares the two interpolation methods together and also to the global yield locus approach.

The computation of the texture evolution is an important part of the presented research. However, it will be further investigated in the Ph. D. thesis.

As already mentioned, the validity of the presented model is the prediction of the anisotropic plastic behaviour of metals on the basis of their crystallographic texture. It is then dedicated to polycrystalline materials. Starting from the initial texture of the material, the model computes the texture updating due to plastic strain and takes it into account to define plastic behaviour during the whole finite element simulation. At this stage, the model is available for face centred (f.c.c.) and body centred cubic (b.c.c.) metals but can be adapted to hexagonal compact lattices.

Beneath this plastic model, a power-type isotropic hardening law and an isotropic elastic behaviour are implemented.

Notation:

- Vector: \underline{V}
- Second order tensor: \underline{T}
- Scalar product between two vectors: $\underline{A}^T \bullet \underline{B} \rightarrow A_i \cdot B_i$
- Scalar product between two second order tensors: $\underline{A} \bullet \underline{B} \rightarrow A_{ij} \cdot B_{jk}$
- Einstein summation is always applied
- Tensor contraction: $\underline{A} \circ \underline{B} \rightarrow A_{ij} \cdot B_{ij}$

Chapter I Stress tensor Derived From The Texture

In order to take the texture of the material into account during a finite element simulation, we use the Full Constraint (FC) Taylor's model which is summarised hereafter.

The velocity gradient \underline{L} of a single crystal (which is assumed to be the same as the macroscopic one in a FC Taylor's model) can be decomposed in a slip induced part and a rate of crystal lattice rotation $\underline{\omega}^L$:

$$(1) \quad \underline{L} = \underline{K}_S \cdot \dot{\gamma}_S + \underline{\omega}^L, \quad S = 1, 24 \text{ in } b.c.c. \text{ crystals} \\ 1, 12 \text{ in } f.c.c. \text{ crystals}$$

where \underline{K}_S are matrices constructed from the normal to the slip plane and the slip direction of each slip system S . They represent the velocity gradient due to the plastic deformation induced by a particular slip system. $\dot{\gamma}_S$ is the associated slip rate. Einstein summation is applied on S representing one slip system. This approach neglects elastic crystal behaviour. In this chapter, strain rate means plastic strain rate.

The velocity gradient \underline{L} may be split into a deviatoric strain rate and a rate of rigid body rotation:

$$(2) \quad \begin{cases} \underline{\dot{\epsilon}} = \text{sym}(\underline{L}) = \underline{A}_S \cdot \dot{\gamma}_S \\ \underline{\omega} = \text{skw}(\underline{L}) = \underbrace{\underline{Z}_S \cdot \dot{\gamma}_S}_{=\underline{\omega}^P} + \underline{\omega}^L \end{cases}$$

with $\underline{A}_S = \text{sym}(\underline{K}_S)$ and $\underline{Z}_S = \text{skw}(\underline{K}_S)$.

The resolved shear stress acting on a slip system can be derived by projecting the microscopic stress on the corresponding \underline{A}_S matrix:

$$(3) \quad \tau_S = \underline{A}_S \circ \underline{\sigma}$$

The yield locus of a single crystal is then defined by the Schmid's law:

$$(4) \quad -\tau_{-S}^c \leq \tau_S \leq \tau_{+S}^c \quad \Rightarrow \quad -\tau_{-S}^c \leq \underline{A}_S \circ \underline{\sigma} \leq \tau_{+S}^c$$

where $\tau_{\pm S}^c$ are the so-called critical resolved shear stresses.

Equal signs hold for plastic deformation, while the inequalities delimit the elastic domain.

Several different combinations of slip rates may achieve the prescribed strain rate according to the first part of equation (2). Taylor's principle states that only the one which minimises the internal power dissipation in the slip systems is chosen:

$$(5) \quad P_T = \tau_S^c \cdot |\dot{\gamma}_S| = \min$$

In order to solve equation (2) constraint to the condition (5), we use a linear programming algorithm which computes the slip rates $\dot{\gamma}_S$ and gives the microscopic stress tensor (see Ref. [16]).

The so-called Taylor's factor M is conventionally derived from the plastic power dissipation per unit volume P for a given strain rate mode $\underline{U}_{\dot{\epsilon}}$ as:

$$(6) \quad M(\underline{\underline{U\dot{\epsilon}}}) = \frac{P}{\tau^c \cdot \dot{\epsilon}_{EQ}} = \frac{\underline{\underline{\sigma}} \circ \underline{\underline{\dot{\epsilon}}}}{\tau^c \cdot \dot{\epsilon}_{EQ}} = \frac{\underline{\underline{\sigma}} \circ \underline{\underline{U\dot{\epsilon}}}}{\tau^c}$$

where the prescribed strain rate $\underline{\underline{\dot{\epsilon}}}$ has been split up into a scalar magnitude $\dot{\epsilon}_{EQ}$ and a mode $\underline{\underline{U\dot{\epsilon}}}$ with the definition:

$$(7) \quad \dot{\epsilon}_{EQ} = \sqrt{\dot{\epsilon}_{ij} \cdot \dot{\epsilon}_{ij}} \quad \text{and} \quad \underline{\underline{U\dot{\epsilon}}} = \frac{\underline{\underline{\dot{\epsilon}}}}{\dot{\epsilon}_{EQ}}$$

and τ^c is the reference critical resolved shear stress. It assumes that all τ_s^c are identical to τ^c .

The plastic power dissipation per unit volume P has been defined according to the maximum work principle as:

$$(8) \quad P = \underline{\underline{\sigma}} \circ \underline{\underline{\dot{\epsilon}}}$$

Finally, it is important to remark that the previously described internal dissipation power and plastic power dissipation result from a primal-dual approach of the crystallographic slip resolution. These two formulations of the power are then equal.

The micro-macro transition can be easily achieved through an averaging of the microscopic strain tensor and the microscopic stress tensor related to each crystallographic orientation. This concept is very simple but the averaging must be done simultaneously on the stress and the strain because of their interaction in the single crystal. A solution to this problem is then very hard to find.

Several assumptions have then been proposed in order to be able to solve the micro-macro transition. The main ones are:

- The Sachs' model. It assumes a homogeneous stress distribution throughout the whole polycrystal. The microscopic stress in each crystal being equal to the macroscopic one. The averaging must only be achieved on the strains.
- The Taylor's model. On the other hand, it assumes a homogeneous distribution of the plastic strain rate. The macroscopic stress of the polycrystal is then computed by averaging the microscopic stress on each crystal. This model will be used in the present work.
- The self-consistent model (see Ref. [1]). It considers each grain in turn as an inclusion embedded in an ideal homogeneous plastic matrix. As this plastic matrix behaviour is obtained by an averaging of the single crystal behaviour, an iterative procedure must be used. This model should conceptually be more accurate than the two previous ones but requires larger computation time.

In order to be able to compute the stress corresponding to a particular strain rate direction with the Taylor's model, some data about the studied steel must be obtained.

The first one is the initial texture of the material, i.e. the orientation of the different grains contained in the polycrystal. This texture can be measured by a X-ray diffraction method. It is described by the so-called Orientation Distribution Function (ODF) which is a mathematical formulation commonly used to represent the texture. A convenient way to formulate the ODF is to use the Euler angles describing all the possible orientations of a grain in the 3-D space. The ODF is then simply the volume fraction of the grains having the orientation described by the three Euler angles.

The ODF is a convenient way to express the texture of a material. However, before applying the Taylor's model, a representative set of crystallographic orientations must be deduced from the ODF. This representative set is computed according to the statistical method proposed in Ref. [15]. The Taylor's model is then applied on each crystal of the representative set and the micro-macro transition is achieved through averaging the microscopic stress from each grain of the representative set.

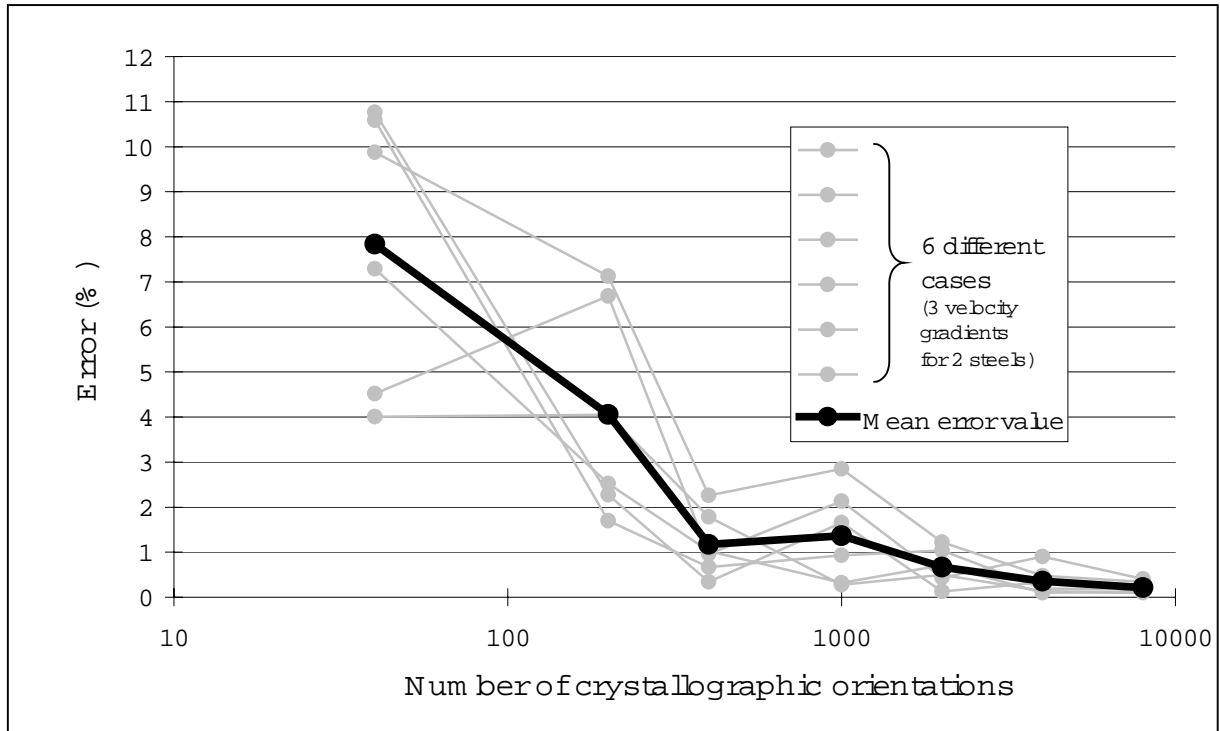


Figure 1: Influence of the number of crystal orientations on the accuracy (the reference result is taken as the one computed with 40000 orientations).

Figure 1 shows that the number of those orientations should not be smaller than 2000 in order to be accurate (mean error smaller than 1%). This number of crystallographic orientations included in the representative set will generally be used with the Taylor's model; especially in the applications presented in Chapter IV of the present work.

Secondly, the permitted slip systems of the crystal lattice must be introduced as the Taylor's model data. The parameters describing these permitted slip systems are needed in order to compute the \underline{K}_S matrices of equation (1) for each slip system. For b.c.c. metals, we use the 24 $\{1\ 1\ 0\} \langle 1\ 1\ 1 \rangle$ and $\{1\ 1\ 2\} \langle 1\ 1\ 1 \rangle$ slip systems. For f.c.c. metals, the 12 $\{1\ 1\ 1\} \langle 1\ 1\ 0 \rangle$ permitted slip systems are imposed.

Beside these material data needed by the Taylor's model, the constitutive law makes use of different aspects of the material properties: the elastic properties and the hardening behaviour. As the elastic and the hardening behaviour are supposed to be isotropic, they can be obtained from classical tensile tests.

As this method uses a refined representation of the plastic behaviour, it requires some particularities in its implementation into a finite element code.

A first remark is that it needs large computation time. Indeed, when a new interpolation domain must be computed or when texture updating takes place, the Taylor's model is called for each crystallographic orientation of the representative set and this can occur at each integration point of the finite element mesh. As the number of the orientations is generally chosen equal to 2000 and the number of integration points can also be very high depending on the structure to model, this constitutive law often leads to large computation time.

It should also be noticed that this model needs a lot of disk space because it is compulsory to store the 3 Euler angles of each crystallographic orientation. Furthermore, as the texture evolution can be different from one point to another, it is required to store those orientations for each integration point. In order to achieve the texture updating, the strain history of each integration point must also be stored.

For the interpolation method, the points computed by the Taylor's model must also be stored in an internal variable meanwhile two domain updating.

Chapter II The Stress-Strain Interpolation

A *Introduction*

The method presented here works in an equivalent way than direct Taylor's model calls. Indeed, from a strain rate direction, the Taylor's model computes the corresponding stress. Our model also computes the stress from the strain rate direction. The goal of the interpolation method is in fact to reduce computation time for strain rate directions close to some already computed directions. We interpolate the results between some points that have been previously computed with the Taylor's model.

B *N-dimensional space geometry*

1 Domain limit vectors

Around a reference direction called \underline{S}_0 (unit vector), the interpolation domain will be defined. To do so, one must find some vectors uniformly placed around \underline{S}_0 . These vectors must fulfil the following prerequisites:

- In a N -dimensional space, N vectors \underline{S}_i are needed
- They are unit vectors:

$$(9) \quad \underline{S}_i^T \bullet \underline{S}_i = 1$$

- There is a common angle between each other:

$$(10) \quad \underline{S}_j^T \bullet \underline{S}_i = 1 - \beta^2 \quad , i \neq j$$

Equations (9) and (10) give:

$$(11) \quad \underline{S}_j^T \bullet \underline{S}_i = 1 + \beta^2 \cdot (\delta_{ij} - 1) \quad , \forall i, j$$

(δ_{ij} is the Kronecker symbol)

- There is a common angle ϑ between \underline{S}_0 and each \underline{S}_i :

$$(12) \quad \underline{S}_0^T \bullet \underline{S}_i = \cos(\vartheta)$$

The angle ϑ is a user parameter and defines directly the size of the interpolation domain.

With these definitions, the reference vector \underline{S}_0 is the mean value of the domain limit vectors \underline{S}_i (the vector's lengths not taken into account):

$$(13) \quad \underline{S}_0 = k \cdot \sum_{i=1}^N \underline{S}_i$$

Combining equation (12) and the fact that \underline{S}_0 is a unit vector, one can deduce the following relations:

$$(14) \quad k = \frac{1}{N \cdot \cos(\vartheta)} \quad \Rightarrow \quad \underline{S}_0 = \frac{1}{N \cdot \cos(\vartheta)} \cdot \sum_{i=1}^N \underline{S}_i$$

$$(15) \quad \text{and} \quad \beta^2 = \frac{N}{N-1} \cdot \sin^2(\vartheta)$$

2 Definition of a vector basis into the domain

As the $N \underline{S}_i$ vectors are linearly independent, they can be used as a vector basis in the N -dimensional stress space. However, they are not orthogonal to each other; so, it is useful to define N new vectors \underline{SS}_i such that they are perpendicular to the \underline{S}_i :

$$(16) \quad \underline{SS}_i^T \bullet \underline{S}_j = \delta_{ij}$$

Due to this definition, these vectors are not unitary; they can be obtained by linear combination of \underline{S}_i and \underline{S}_0 :

$$(17) \quad \underline{SS}_i = a \cdot \underline{S}_i + b \cdot \underline{S}_0$$

Where a and b can be deduced from equation (16); the final form of \underline{SS}_i vectors is:

$$(18) \quad \underline{SS}_i = \frac{1}{\beta^2} \cdot \left[\underline{S}_i - \frac{1-\beta^2}{\cos(\vartheta)} \cdot \underline{S}_0 \right]$$

Two useful relations can be deduced from the definition of \underline{SS}_i vectors:

$$(19) \quad \sum_{i=1}^N \underline{SS}_i = \frac{\underline{S}_0}{\cos(\vartheta)}$$

$$(20) \quad \text{and} \quad \underline{SS}_j^T \bullet \underline{SS}_i = \frac{\delta_{ij}}{\beta^2} - \frac{1-\beta^2}{\beta^2 \cdot N \cdot \cos^2(\vartheta)}$$

3 Use of the new basis

The vectors defined above will be used to decompose any vector \underline{V} (non-unitary) in the following form:

$$(21) \quad \underline{V} = \sum_{i=1}^N \eta_i \cdot \underline{S}_i$$

The η coefficients can be obtained by projecting \underline{V} on the \underline{SS} vectors:

$$(22) \quad \underline{SS}_j^T \bullet \underline{V} = \sum_{i=1}^N \eta_i \cdot \underbrace{\underline{SS}_j^T \bullet \underline{S}_i}_{=\delta_{ij}} = \eta_j$$

These N η -coordinates are independent to each other; their knowledge is equivalent to the knowledge of the vector \underline{V} , with its direction and its length. It can be easily found that:

$$(23) \quad \sum_{i=1}^N \eta_i = \frac{\underline{V}^T \bullet \underline{S}_0}{\cos(\vartheta)}$$

As this equation is not very interesting, new coordinates will also be defined.

4 Intrinsic coordinates

These coordinates are defined:

$$(24) \quad \xi_i = \frac{\cos(\vartheta)}{\underline{V}^T \cdot \underline{S}_0} \cdot \underline{V}^T \cdot \underline{SS}_i = \frac{\eta_i}{\sum_{j=1}^N \eta_j}$$

such that $\sum \xi_i = 1$; which is an interesting property.

These ξ_i -coordinates are equivalent to the intrinsic coordinates used in the two or three-dimensional finite element method. Here is presented the N -dimensional general case.

Unlike the η -coordinates, these intrinsic coordinates are dependent to each other. The N^{th} coordinate can be deduced from the $N-1$ others. The \underline{V} vector is then no more completely represented by these ξ -coordinates; only its direction is known, its length must be defined separately.

It can be noticed that the direction and the length are completely uncoupled with this representation. This is an important property but, on the other hand, the \underline{V} vector cannot be completely recovered knowing the intrinsic coordinates.

5 Properties of those coordinates

Some important properties of the η and ξ -coordinates are presented:

- For an unit vector \underline{V} corresponding to a domain limit vector \underline{S}_i , the η and ξ -coordinates are all equal to 0 or 1.
- The limits of an interpolation domain are defined such that all the η and ξ -coordinates are positive. From this definition it can be deduced that the ξ_i are smaller than 1.
- The N limits of the domain correspond to $\eta_i = \xi_i = 0$
- The N vertices are defined by $\eta_i = \xi_i = 1$ and the other coordinates equal to zero. They are the domain limit vectors.
- The knowledge of the N η or ξ -coordinates is equivalent to the knowledge of any vector's direction (corresponding to $N-1$ independent variables); in addition, the η_i include the length of the vector.

6 Computation of the S vectors

The way we compute the \underline{S}_i vectors is based on a particular case. One particular reference direction \underline{S}_0^* is chosen such that all its N components are the same. The domain limit vectors are computed by adding consecutively each vector of the cartesian basis to the reference \underline{S}_0^* :

$$(25) \quad \underline{S}_i^* = \alpha' \cdot \underline{S}_0^* + \beta \cdot \underline{t}_i = \underbrace{\alpha'}_{=\alpha} \cdot \frac{1}{\sqrt{N}} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} + \beta \cdot \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

According to equation (14), we have:

$$(26) \quad \underline{S}_0^* = \frac{1}{N \cdot \cos(\vartheta)} \cdot \sum_{i=1}^N \underline{S}_i^* = \frac{N \cdot \alpha + \beta}{N \cdot \cos(\vartheta)} \cdot \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

where α and β are such that the unitary conditions $\|\underline{S}_0^*\| = 1$ and $\|\underline{S}_i^*\| = 1$ are fulfilled. One can easily check that β is the same than the one defined by equation (15) and:

$$(27) \quad \alpha = \frac{\cos(\vartheta)}{\sqrt{N}} - \frac{\sin(\vartheta)}{\sqrt{N \cdot (N-1)}}$$

Once this particular case is completely defined (reference and domain limit directions), all these vectors will be rotated in order to have \underline{S}_0^* and \underline{S}_0 pointing the same direction. By the way, the \underline{S}_i^* will be equal to the \underline{S}_i . The rotation matrix is computed according to equation (28):

$$(28) \quad \underline{R} = \underline{1} + 2 \cdot \underline{S}_0 \cdot \underline{S}_0^{*T} - \frac{(\underline{S}_0 + \underline{S}_0^*) \cdot (\underline{S}_0 + \underline{S}_0^*)^T}{1 + \underline{S}_0^T \cdot \underline{S}_0^*}$$

Some remarks concerning \underline{R} are listed:

- This definition fulfils the rotation condition: $\underline{R} \cdot \underline{R}^T = \underline{1}$
- The rotation can be applied on the reference vectors:

$$(29) \quad \underline{R} \cdot \underline{S}_0^* = \underline{S}_0$$

- It is also applied on the domain limit vectors in order to compute them:

$$(30) \quad \underline{R} \cdot \underline{S}_i^* = \underline{S}_i$$

- If \underline{S}_0^* and \underline{S}_0 are opposite vectors, equation (28) is not valid; the domain limit vectors \underline{S}_i can be computed as the opposite of the \underline{S}_i^* .

It can be noticed that when the \underline{S}_i vectors are computed, the \underline{SS}_i vectors can be obtained through equation (18).

7 Entering the adjacent domain

During a finite element simulation, the domain where the yield locus is defined must frequently be updated. A new domain is computed when the old one is not anymore valid.

The classical way for such updating is based on the previous paragraph, where the reference direction \underline{S}_0 , which is in the middle of the interpolated domain, will be chosen as close as possible to the direction in which the yield locus must be explored.

However, it can be interesting to take advantage of the neighbouring of the domains. The 3-dimensional representation of Figure 2 compares the two methods.

Next to a particular frontier, defined by $N-1$ domain limit vectors (2 for the 3-D case), there exists only 2 domains. Those 2 domains are defined by their common frontier and only one other domain limit vector. So, to go from one to the other adjacent domain, the required information is only one limit vector; the frontier being known.

As the computation of a vector according to the Taylor's model is time consuming, this method is very interesting. Only one instead of N vectors is computed.

If the old domain is left along the frontier $\xi_i = 0$ (see paragraph 5), the domain limit vector \underline{S}_i must be replaced by the new \underline{S}_i' which is computed according to equation (31).

$$(31) \quad \underline{S}_i' = a \cdot \underline{S}_i + b \cdot \underline{S}_0$$

The constants a and b must be such that \underline{S}_i' fulfils the following conditions:

$$(32) \quad \underline{S}_i'^T \bullet \underline{S}_i' = 1$$

$$(33) \quad \underline{S}_i'^T \bullet \underline{S}_j = 1 - \beta^2 \quad , i \neq j$$

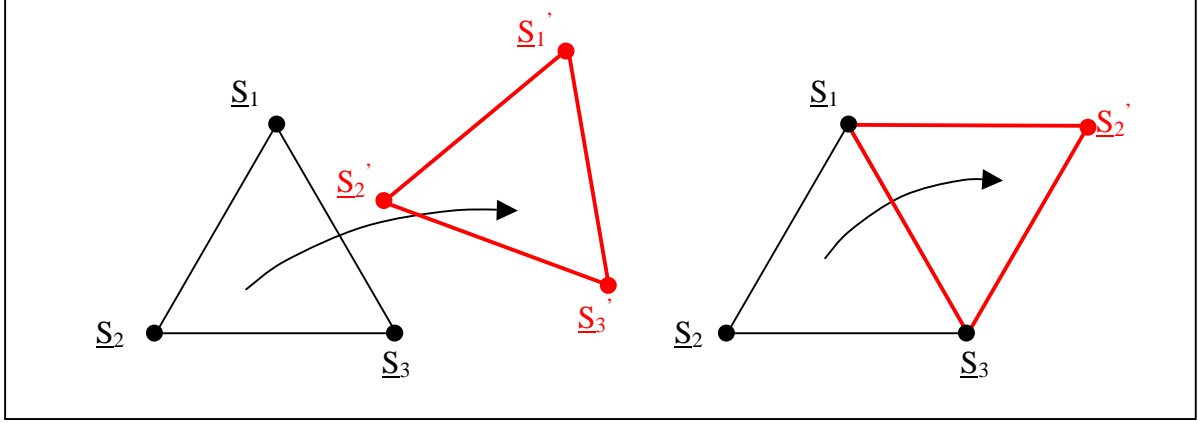


Figure 2: Classical and improved domain updating methods

Finally, as a new domain is explored, a new \underline{S}_0' vector must be computed either according to its definition (equation (14)) or from the old domain values:

$$(34) \quad \left. \begin{aligned} \underline{S}_0' &= \frac{\sum_{j=1}^N \underline{S}_j}{N \cdot \cos(\vartheta)} \quad \text{with } \underline{S}_i \text{ replaced by } \underline{S}_i' \\ \underline{S}_0 &= \frac{\sum_{j=1}^N \underline{S}_j}{N \cdot \cos(\vartheta)} \end{aligned} \right\} \Rightarrow \underline{S}_0' = \underline{S}_0 + \frac{\underline{S}_i' - \underline{S}_i}{N \cdot \cos(\vartheta)}$$

When the limit vectors of the new domain are computed, it is interesting to look at the new coordinates. The frontier opposite to the vertex \underline{S}_i is defined by:

$$(35) \quad \eta_i = \underline{V}^T \bullet \underline{SS}_i = 0$$

Where \underline{V} is a particular vector on the frontier.

Replacing \underline{SS}_i by its definition (equation (18)), we get:

$$(36) \quad \underline{V}^T \bullet \underline{S}_i = \underline{V}^T \bullet \underline{S}_0 \cdot \frac{1 - \beta^2}{\cos(\vartheta)}$$

The other η -coordinates which are not equal to zero are:

$$(37) \quad \eta_k = \underline{V}^T \bullet \underline{SS}_k = \underline{V}^T \bullet \left(\underline{S}_k - \underline{S}_0 \cdot \frac{1 - \beta^2}{\cos(\vartheta)} \right) \cdot \frac{1}{\beta^2}$$

In the adjacent domain, the η -coordinates can also be computed. First, the η_i' coordinate corresponding to the new domain limit vector \underline{S}_i' :

$$\begin{aligned}
 \eta'_i &= \underline{V}^T \bullet \underline{SS}'_i \\
 (38) \quad &= \underline{V}^T \bullet \left(\underline{S}'_i - \underline{S}'_0 \cdot \frac{1-\beta^2}{\cos(\vartheta)} \right) \cdot \frac{1}{\beta^2}
 \end{aligned}$$

Using equation (34) to express \underline{S}'_0 :

$$(39) \quad \eta'_i = \underline{V}^T \bullet \left(\underline{S}'_i - \left(\underline{S}_0 + \frac{\underline{S}'_i - \underline{S}_i}{N \cdot \cos(\vartheta)} \right) \cdot \frac{1-\beta^2}{\cos(\vartheta)} \right) \cdot \frac{1}{\beta^2}$$

\underline{S}_0 can be eliminated with the use of equation (18):

$$\begin{aligned}
 \eta'_i &= \underline{V}^T \bullet \left(\underline{S}'_i + \underline{SS}_i \cdot \beta^2 - \underline{S}_i - \frac{\underline{S}'_i - \underline{S}_i}{N \cdot \cos(\vartheta)} \cdot \frac{1-\beta^2}{\cos(\vartheta)} \right) \cdot \frac{1}{\beta^2} \\
 (40) \quad &= \underbrace{\underline{V}^T \bullet \underline{SS}_i}_{=\eta_i} + \underline{V}^T \bullet (\underline{S}'_i - \underline{S}_i) \cdot \left(1 - \frac{1-\beta^2}{N \cdot \cos^2(\vartheta)} \right) \cdot \frac{1}{\beta^2}
 \end{aligned}$$

On the frontier between the two adjacent domains, as η_i is equal to zero, η'_i is equal to zero too if:

$$(41) \quad \underline{V}^T \bullet \underline{S}'_i = \underline{V}^T \bullet \underline{S}_i$$

Due to the definition of the new domain limit vector \underline{S}'_i , equation (41) is fulfilled. Indeed, \underline{V} is a vector on the frontier and this frontier is a symmetry plane between \underline{S}_i and \underline{S}'_i . The projections of these two vectors on \underline{V} are necessary equal.

The frontier is then defined by $\eta'_i = \eta_i = 0$.

The other coordinates of the new domain can also be computed:

$$\begin{aligned}
 \eta'_k &= \underline{V}^T \bullet \underline{SS}'_k \\
 (42) \quad &= \underline{V}^T \bullet \left(\underline{S}_k - \underline{S}'_0 \cdot \frac{1-\beta^2}{\cos(\vartheta)} \right) \cdot \frac{1}{\beta^2}
 \end{aligned}$$

Note that $\underline{S}'_k = \underline{S}_k$ because it is kept from the old domain.

By the same way than previously, we get:

$$(43) \quad \eta'_k = \underline{V}^T \bullet \left(\underline{S}_k - \left(\underline{S}_0 + \frac{\underline{S}'_i - \underline{S}_i}{N \cdot \cos(\vartheta)} \right) \cdot \frac{1-\beta^2}{\cos(\vartheta)} \right) \cdot \frac{1}{\beta^2}$$

Eliminating \underline{S}_0 with equation (18) applied to \underline{SS}_k :

$$\begin{aligned}
 \eta'_k &= \underline{V}^T \bullet \left(\underline{S}_k + \underline{SS}_k \cdot \beta^2 - \underline{S}_k - \frac{\underline{S}'_i - \underline{S}_i}{N \cdot \cos(\vartheta)} \cdot \frac{1-\beta^2}{\cos(\vartheta)} \right) \cdot \frac{1}{\beta^2} \\
 (44) \quad &= \underbrace{\underline{V}^T \bullet \underline{SS}_k}_{=\eta_k} + \underline{V}^T \bullet (\underline{S}'_i - \underline{S}_i) \cdot \frac{1-\beta^2}{N \cdot \cos^2(\vartheta) \cdot \beta^2}
 \end{aligned}$$

Finally equation (41) leads to:

$$(45) \quad \eta'_k = \eta_k$$

This equation implies that along a frontier between two adjacent domains, the η -coordinates computed from one or the other domain are equal. It is a very important and

convenient property that is also encountered in isoparametric finite element intrinsic coordinates.

The main advantages of this method are that it requires only 1 (instead of 5) Taylor's model call for an updating, it improves the continuity of the resulting yield locus and the continuity of its normal. A drawback is that it can only be used when the adjacent domain contains the current stress and then gives rise to positive new intrinsic co-ordinates. This condition is satisfied when the stress evolves smoothly which is often the case during a finite element simulation.

8 2-D and 3-D cases

The yield locus using this interpolation method is located in the 5-dimensional deviatoric stress space. In order to have a more physical view, a 2-D and 3-D representations of the yield locus are explained.

In 2-dimensional space, the yield locus would be a plane curve, i.e. a 1-dimensional domain. The most simple representation is a segment of a straight line limited by 2 points.

In 3-D space, the yield locus would be a surface (a 2-D domain). The most simple representation would be a triangular facet limited by the 3 lines (1-D domains) joining the 3 vertices of the facet.

In the same way, for N -dimensional space, the yield locus is a $(N-1)$ dimensional domain. The most simple representation is an hyperplane (dimension: $N-1$) limited by N frontiers (dimension: $N-2$) and defined by N points.

A point in the domain can be located by N intrinsic coordinates ζ which are linked to the N vertices. On a frontier, one intrinsic coordinate is equal to zero; so, this frontier which is a $N-2$ dimensional domain is defined by $N-1$ points and can be described by $N-1$ intrinsic coordinates (the non-zero ones).

9 Non-regular domains

According to equation (10), the angle between the domain limit vectors are all the same. This implies that only regular domains are used. In 3-D case, the interpolated yield locus is represented by equilateral triangles.

However, it would be possible to define an interpolated domain based on non-uniformly located limit vectors. For 3-D case, the triangular facets would no more be equilateral.

With a non-regular domain, the intrinsic coordinates are still available but some particularities should be noticed:

- The choice of the N domain limit vectors \underline{S}_i is free; the angle between each other is not fixed. However, as they are a basis in the N -dimensional space, they can not be parallel to each other; they must be linearly independent.
- In order to compute the \underline{SS}_i vectors, equation (18) is no more valid. The definition based on equation (16) ($\underline{S}_i^T \cdot \underline{SS}_j = \delta_{ij}$) must be used. As the \underline{S} vectors form a basis, it is always possible to compute the \underline{SS}_i vectors. Equations (19) and (20) are no more fulfilled.
- The η -coordinates are still computed from equation (22) ($\eta_i = \underline{V}^T \cdot \underline{SS}_i$). On the other hand, the \underline{V} vector can be found knowing its coordinates according to equation (21) ($\underline{V} = \underline{\Sigma} \eta_i \underline{S}_i$).
- The intrinsic coordinates are computed in the same way:

$$(46) \quad \xi_i = \frac{\eta_i}{\sum_{j=1}^N \eta_j} = \frac{V^T \cdot \underline{SS}_i}{V^T \cdot \sum_{j=1}^N \underline{SS}_j}$$

Note that this equation is slightly different from equation (24) (the original definition) in the sense that the sum on the \underline{SS}_i vectors can no more be replaced by its value as a function of \underline{S}_0 .

The sum of the intrinsic coordinates is again equal to one.

- The \underline{S}_0 vector is no more a particular direction; it can however be defined:

$$(47) \quad \underline{S}_0 = c \cdot \sum_{i=1}^N \underline{S}_i$$

The c constant must be chosen such that \underline{S}_0 is a unit vector.

For that direction, all the intrinsic coordinates are the same and their common value is $(1/N)$.

- Just like for regular domains, the N domain's frontiers are defined by $\xi_i = \eta_i = 0$; the N vertices are defined by $\xi_i = 1$ and the other coordinates equal to zero.
- The property defined by equation (45) stating that the η -coordinates computed from two adjacent domains are equal on the common frontier is no more valid.

Aside those differences, the main properties of the ξ and η -coordinates are still fulfilled. So, non-regular domains can be used just like regular ones.

C Yield locus interpolation

Contrarily to numerous phenomenological models, in the developments integrating the texture, there does not exist a mathematical formulation describing the yield locus continuously. It is only possible to numerically compute, for a particular direction called \underline{d} , the length of the vector whose extremity lies on the yield locus. The length of this vector will be called s^* .

As the computation of one point of the yield locus with the texture model is time consuming, it is interesting to interpolate locally the yield locus between some computed points. Different interpolation methods can be used; some are presented hereafter.

The problem to solve is to find a formulation in order to interpolate s^* , the length of a vector as a function of its direction \underline{d} which can be described by the intrinsic coordinates.

In the finite element analysis, both stress space and strain rate space are closely linked together. The choice of the interpolation space will be imposed by the texture model. Indeed, the Taylor's model allows to compute, for a given strain rate direction, the stress direction and its length so, one point of the yield locus. On the other hand, if one wants to impose the stress direction, it is necessary to iterate in order to find the corresponding strain rate direction.

Due to this, both spaces will be used. For the strain rate space, the domain limit vectors will be computed according to the method presented in paragraph B6; a regular domain will be obtained. The Taylor's model will be used to compute the N domain limit vectors in the stress space corresponding to the ones in the strain rate space. The vertices obtained so are not uniformly located. Their irregular spacing is function of the anisotropy of the material behaviour. The interpolation domain in stress space will be non-regular.

Inside a domain, the intrinsic coordinates computed in stress space and in strain rate space are not exactly the same, again due to the anisotropy of the material behaviour. However, the frontiers and the vertices respectively correspond in both spaces. The stress space and its non-regular domains are chosen as reference for the computation of the intrinsic coordinates. In a convenient way, these coordinates will be used in strain rate space too.

The dimension of the interpolating space, i.e. the value of N is the stress space dimension. As the yield locus is assumed to be independent of the mean stress, it is interesting to use the reduced deviatoric stress space whose dimension is 5 instead of 6.

1 Linear interpolation in cartesian coordinates (hyperplane method)

This interpolation method is the most simple and implies only the computation of the length of the yield stress vector (s_i) along the N domain limit directions \underline{S}_i . The stress vector is defined as the mean value of the yield stresses along the domain limit directions weighted by the intrinsic coordinates of the selected direction:

$$(48) \quad \underline{\sigma} = s^* \cdot \underline{d} = \sum_{i=1}^N \xi_i \cdot s_i \cdot \underline{S}_i \quad \text{with} \quad \sum_{i=1}^N \xi_i = 1$$

Using the properties of the \underline{SS}_i vectors, it is possible to compute the value of the interpolated length s^* as a function of the direction \underline{d} :

$$(49) \quad s^* \cdot \underline{SS}_j^T \cdot \underline{d} = \sum_{i=1}^N \xi_i \cdot s_i \cdot \underline{SS}_j^T \cdot \underline{S}_i = \xi_j \cdot s_j$$

$$(50) \quad \text{now} \quad \sum_{i=1}^N \xi_i = \sum_{i=1}^N \frac{s^* \cdot \underline{SS}_i^T \cdot \underline{d}}{s_i} = 1$$

$$(51) \quad \Rightarrow \quad \frac{1}{s^*} = \sum_{i=1}^N \frac{\underline{SS}_i^T \cdot \underline{d}}{s_i}$$

Furthermore, for the case of a regular domain, equation (24) leads to:

$$(52) \quad \frac{1}{s^*} = \frac{S_0^T \cdot \underline{d}}{\cos(\vartheta)} \cdot \sum_{i=1}^N \frac{\xi_i}{s_i}$$

An important remark should be mentioned here; when all the s_i for the N domain limit vectors are equal, the interpolated length is not equal to the common value:

$$(53) \quad \frac{1}{s^*} = \sum_{i=1}^N \frac{\underline{SS}_i^T \cdot \underline{d}}{s} \quad \Rightarrow \quad s^* = \frac{s}{\sum_{i=1}^N \underline{SS}_i^T \cdot \underline{d}} \neq s$$

For a regular domain, according to equation (19)

$$(54) \quad s^* = s \cdot \frac{\cos(\vartheta)}{S_0^T \cdot \underline{d}} \neq s$$

And e.g. for $\underline{d} = \underline{S}_0$:

$$(55) \quad s^* = s \cdot \cos(\vartheta) < s$$

This interpolation appears as a plane facet as shown in the 2-dimensional representation of Figure 3.

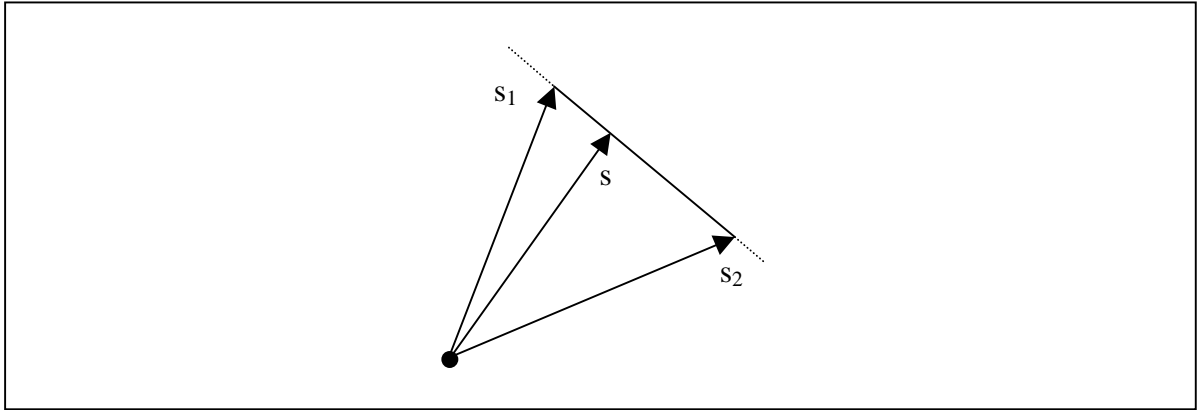


Figure 3: 2-D representation of the hyperplane interpolation

2 Linear interpolation in spherical coordinates (stress-strain interpolation)

Instead of interpolating in cartesian coordinates, i.e. each component of the stress vector independently, it is possible to interpolate the stress direction and the stress vector length separately.

In a 3-dimensional space, the direction is represented by 2 scalars (which are often chosen to be 2 angles) and the length of any vector is 1 scalar. In the N -dimensional space, the direction is defined by $N-1$ parameters and the length is 1 parameter. The N intrinsic coordinates ξ will be used to represent the direction; indeed, they represent $N-1$ independent parameters (see paragraph B4).

The interpolation will be carried out this way: the N intrinsic coordinates will be deduced from the direction of the stress vector (the direction to which the yield locus is explored). The length s^* of the vector leading to the yield stress will be linearly interpolated according to equation (56).

$$(56) \quad s^* = \sum_{i=1}^N \xi_i \cdot s_i$$

With this interpolation method, if the domain limit vector lengths s_i are all the same, the interpolated value s^* will also be the same. So, a hypersphere (a sphere in the N -dimensional space) will be exactly represented through this interpolation.

Both interpolation methods presented above are worth to be integrated in a finite element code. The first one is more adapted to represent a yield locus with facets (like the Tresca's one). The second one should be used for quadratic yield loci (like Von Mises' one). So, the superiority of one method will depend on the material behaviour.

3 Inverse interpolation

Instead of using equation (56), it is possible to interpolate the inverse of the vector length:

$$(57) \quad \frac{1}{s^*} = \sum_{i=1}^N \frac{\xi_i}{s_i}$$

This method is not very different from the direct interpolation; particularly, in the case of domain limit vectors having the same length, both interpolations lead to the same result.

However, if the N vector lengths s_i are significantly different from each other, this interpolation formulation is similar to the hyperplane method (compare equations (52) and (57)).

Another variant of equation (56) would be to use the η -coordinates:

$$(58) \quad s^* = \sum_{i=1}^N \eta_i \cdot s_i = \frac{d^T \cdot S_0}{\cos(\vartheta)} \cdot \sum_{i=1}^N \xi_i \cdot s_i \quad (\text{for regular domains})$$

4 'Bubble' mode

The cartesian and spherical linear interpolations presented above can be improved by the adjunction of one term:

$$(59) \quad a \cdot \prod_{i=1}^N N \cdot \xi_i$$

where a is a constant.

Due to the fact that on a frontier, at least one intrinsic coordinate is equal to zero, the added term has only an influence inside the interpolated domain. This is the reason why it is called 'bubble' mode.

The constant a must be such that a particular yield stress inside the domain fulfils the interpolation. So, this 'bubble' mode implies the computation of a new yield stress vector according to the Taylor's model.

For convenience, if we choose the S_0 direction (with all the ξ_i equal to $1/N$), the constant a can be easily found for the spherical interpolation:

$$(60) \quad s_0 = \sum_{i=1}^N \xi_i \cdot s_i + a \cdot \prod_{i=1}^N N \cdot \xi_i = \sum_{i=1}^N \frac{s_i}{N} + a \quad \Rightarrow a = s_0 - \sum_{i=1}^N \frac{s_i}{N}$$

The formulation of the improved spherical interpolation is then:

$$(61) \quad s^* = \sum_{i=1}^N \xi_i \cdot s_i + \left(s_0 - \sum_{i=1}^N \frac{s_i}{N} \right) \cdot \prod_{i=1}^N N \cdot \xi_i$$

For the cartesian linear interpolation, the 'bubble' mode is a vector:

$$(62) \quad \underline{\sigma} = s^* \cdot \underline{d} = \sum_{i=1}^N \xi_i \cdot s_i \cdot \underline{S}_i + a \cdot \prod_{i=1}^N N \cdot \xi_i$$

The S_0 direction is also chosen for the determination of a :

$$(63) \quad s_0 \cdot \underline{S}_0 = \sum_{i=1}^N \frac{1}{N} \cdot s_i \cdot \underline{S}_i + a \cdot \prod_{i=1}^N N \cdot \frac{1}{N} \quad \Rightarrow a = s_0 \cdot \underline{S}_0 - \sum_{i=1}^N \frac{s_i \cdot \underline{S}_i}{N}$$

The 'bubble' mode improves the accuracy inside the domain for the cartesian and spherical interpolation but requires the computation of one additional particular yield stress.

5 Second order interpolation

Just like in finite element analysis, second order interpolation can be investigated. As for the 'bubble' mode, new yield stress vectors will have to be computed. We will choose points on the frontier between two vertices. In particular, the more interesting points have two intrinsic coordinates equal to $\frac{1}{2}$ and the $(N-2)$ other equal to zero. These new directions are called \underline{S}_{ij} and the lengths corresponding to the yield stresses are s_{ij} .

As these new \underline{S}_{ij} vectors are on a frontier just half-way between two vertices \underline{S}_i , they can be defined as:

$$(64) \quad \underline{S}_{ij} = \alpha \cdot \left(\frac{S_i}{2} + \frac{S_j}{2} \right)$$

The α -constant must be so that \underline{S}_{ij} is a unit vector. One can easily find:

$$(65) \quad \underline{S}_{ij}^T \cdot \underline{S}_{ij} = 1 \quad \Rightarrow \quad S_{ij} = \frac{(S_i + S_j)}{\sqrt{4 - 2 \cdot \beta^2}}$$

The interpolation must combine a constant term, $(N-1)$ linear terms (as the linear interpolations) and $N(N-1)/2$ quadratic terms. With the use of the intrinsic coordinates, all these terms are contained in the $N(N+1)/2$ quadratic terms of the N dependent ξ_i . Due to the fact that their sum is equal to one, the linear and constant terms are taken into account.

Indeed, for the spherical interpolation, the formulation is:

$$(66) \quad s^* = \sum_{i=1}^N \sum_{j=1}^N c_{ij} \cdot \xi_i \cdot \xi_j \quad \text{with} \quad c_{ij} = c_{ji}$$

If one wants to add linear terms having the form:

$$(67) \quad s_{lin}^* = \sum_{i=1}^N d_i \cdot \xi_i = \sum_{i=1}^N \left(d_i \cdot \xi_i \cdot \left(\sum_{j=1}^N \xi_j \right) \right) = \sum_{i=1}^N \sum_{j=1}^N d_i \cdot \xi_i \cdot \xi_j$$

they are included in the formulation of equation (66). For a constant term, the same conclusion appears:

$$(68) \quad s_{cons}^* = e = e \cdot \left(\sum_{i=1}^N \xi_i \right) \cdot \left(\sum_{j=1}^N \xi_j \right) = \sum_{i=1}^N \sum_{j=1}^N e \cdot \xi_i \cdot \xi_j$$

The $N(N+1)/2$ constants c_{ij} are determined by identification with the N values along the vertices: s_i and the $N(N-1)/2$ values on the frontiers between two vertices:

$$(69) \quad c_{ii} = s_i$$

$$(70) \quad c_{ij} = c_{ji} = 2 \cdot s_{ij} - \frac{s_i}{2} - \frac{s_j}{2} \quad \text{with} \quad i \neq j$$

Note that these two equations can be summarised in:

$$(71) \quad c_{ij} = c_{ji} = 2 \cdot s_{ij} - \frac{s_i}{2} - \frac{s_j}{2} \quad \text{for any } i \text{ and } j$$

if we define s_{ii} to be equal to s_i .

The second order interpolation can also be applied to the hyperplane method:

$$(72) \quad \underline{\sigma} = s^* \cdot \underline{d} = \sum_{i=1}^N \sum_{j=1}^N f_{ij} \cdot \xi_i \cdot \xi_j \cdot \underline{S}_{ij}$$

In this formulation, it is assumed that \underline{S}_{ii} is defined to be \underline{S}_i and $f_{ij} = f_{ji}$. The identification of these constants is done in a similar way than for the spherical interpolation.

The inverse interpolation can also be used:

$$(73) \quad \frac{1}{s^*} = \sum_{i=1}^N \sum_{j=1}^N g_{ij} \cdot \xi_i \cdot \xi_j$$

Higher order interpolation can also be investigated. Again, as the sum of the intrinsic coordinates is equal to one, only the higher order terms should be considered (see Ref. [13] and [17]). Such methods improve the accuracy of the interpolations. On the other hand, more yield stresses have to be computed with the Taylor's model. A compromise between accuracy and computation time must be done.

The size of the interpolation domain can also be linked to these two factors. Indeed, with higher order interpolations, it is possible to use larger domains with the same accuracy. Then, the compromise must be done between small domains with linear interpolation or large domains with a higher order interpolation. Both choices are valid. However, as the goal of our method is to integrate the texture updating, the interpolated domain will have to be updated regularly. Then, small domains with a linear interpolation should be more interesting from the point of view of the computation time.

6 Continuity between adjacent domains

The N frontiers of the domains are defined by one intrinsic coordinate equal to zero. If the interpolations are formulated as polynomials of these coordinates, the values computed on a frontier will be only a function of the $(N-1)$ non-zero coordinates. For the adjacent domain limited by the same frontier, the same conclusion will hold. Then, the continuity between the two adjacent domains is fulfilled.

It should be noticed that the continuity between the domains can only be fulfilled when the new domain is the neighbour of the old one (see paragraph B7 and Figure 2).

The continuity presented here is of the C_0 type; it assures the continuity of the interpolated function, i.e. the length of the interpolated stress s^* . Unfortunately, a C_1 continuity is not fulfilled; the continuity of the yield locus normals can not be obtained between two adjacent domains. This problem is also encountered in 3-D shell finite elements where perfect slope compatibility can not be fulfilled.

D Fitting of a yield locus formulation

The fitting of the part yield locus parameters consists in the identification on some points of the real yield locus, which have been computed with the Taylor's model. Instead of interpolating the yield locus into a chosen domain, it can be interesting to fit a classical phenomenological yield locus formulation that locally describes this chosen domain.

The common form of phenomenological yield loci is:

$$(74) \quad f = F(\underline{\sigma}) - \sigma_s^2 = 0$$

Based on classical formulation, the reliability of these yield models has already been proved. This is a first advantage. A second one concerns all the developments already investigated. For instance, the computation of yield locus normals and stiffness matrices are well known.

Note that equation (74) refers to a scaled yield locus. Only the shape of the yield locus representing the anisotropy of the material is taken into account. The scaling factor is based on the work hardening and will be introduced in paragraph F1.

1 Plane formulation

The most simple formulation for the part yield locus is:

$$(75) \quad f = \underline{C}^T \bullet \underline{\sigma} - \sigma_s^2 = 0$$

and represents the equation of a plane in the stress space. The N components of the \underline{C} vector which are the part yield locus parameters can be determined by identification along the domain limit vectors \underline{S}_j :

$$(76) \quad f_i = s_i \cdot \underline{C}^T \bullet \underline{S}_i - \sigma_s^2 = 0$$

Due to the property of the \underline{S}_j -vectors (see equation (16)), it can easily be checked that the following form of \underline{C} -vector fulfils equation (76):

$$(77) \quad \underline{C} = \sigma_s^2 \cdot \sum_{j=1}^N \frac{\underline{S}_j}{s_j}$$

Replacing this value into the part yield locus formulation (equation (75)), one obtains:

$$(78) \quad f = \left(s^* \cdot \underline{d} \right)^T \bullet \left(\sigma_s^2 \cdot \sum_{j=1}^N \frac{\underline{S}_j}{s_j} \right) - \sigma_s^2 = 0$$

$$\Rightarrow \frac{1}{s^*} = \sum_{j=1}^N \frac{\underline{d}^T \bullet \underline{S}_j}{s_j}$$

which is exactly the form of equation (51). Indeed, these two formulations represent a part of a plane in the N -dimensional stress space.

2 Spherical part yield locus

The formulation is:

$$(79) \quad f = \underline{\sigma}^T \bullet \underline{\sigma} - \sigma_s^2 = 0$$

Which can be written:

$$(80) \quad f = \left(s^* \right)^2 \cdot \underline{d}^T \bullet \underline{d} - \sigma_s^2 = 0 \quad \Rightarrow s^* = \sigma_s$$

This is the Von Mises criterion which requires only one yield stress computation. As we expect a good representation of the anisotropy of the material behaviour, this too simple model is not kept.

3 Hill yield locus

An anisotropic quadratic criterion is chosen:

$$(81) \quad f = \frac{1}{2} \cdot \underline{\sigma}^T \bullet \underline{A} \bullet \underline{\sigma} - \sigma_s^2 = 0$$

In this formulation, \underline{A} is a symmetric matrix containing $N(N+1)/2$ components describing the material behaviour. Generally, the \underline{A} matrix has a dominant diagonal. A simplified orthotropic version is often used; in the 5-dimensional deviatoric stress space, only 6 (instead of 15) components are non-zero (see Ref. [12]).

In order to stay general and to have the higher accuracy, the complete version will be used. For the identification of the parameters, the \underline{A} matrix will be written:

$$(82) \quad \underline{\underline{A}} = \sum_{i=1}^N \sum_{j=1}^N C_{kl} \cdot \underline{\underline{SS}}_k \cdot \underline{\underline{SS}}_l^T$$

The identification will require $N(N+1)/2$ known points to compute the $N(N+1)/2$ components of the parameter matrix. The N domain limit vectors $\underline{\underline{S}}_i$ and the $N(N-1)/2$ $\underline{\underline{S}}_{ij}$ directions (defined in paragraph C5) will be used. For the first ones, we obtain:

$$(83) \quad f_i = \frac{1}{2} \cdot s_i^2 \cdot \underline{\underline{S}}_i^T \cdot \underbrace{\left(\sum_{i=1}^N \sum_{j=1}^N C_{kl} \cdot \underline{\underline{SS}}_k \cdot \underline{\underline{SS}}_l^T \right)}_{=C_{ii}} \cdot \underline{\underline{S}}_i - \sigma_s^2 = 0$$

$$\Rightarrow C_{ii} = \frac{2 \cdot \sigma_s^2}{s_i^2}$$

For the $\underline{\underline{S}}_{ij}$ vectors, the following formulation appears:

$$(84) \quad f_{ij} = \frac{1}{2} \cdot s_{ij}^2 \cdot \underline{\underline{S}}_{ij}^T \cdot \left(\sum_{i=1}^N \sum_{j=1}^N C_{kl} \cdot \underline{\underline{SS}}_k \cdot \underline{\underline{SS}}_l^T \right) \cdot \underline{\underline{S}}_{ij} - \sigma_s^2 = 0$$

Due to the definition of the $\underline{\underline{S}}_{ij}$ vectors of equation (65),

$$(85) \quad f_{ij} = \frac{1}{2} \cdot \frac{s_{ij}^2}{(4 - 2 \cdot \beta^2)} \cdot \left(\sum_{i=1}^N \sum_{j=1}^N C_{kl} \cdot \underbrace{\left(\underline{\underline{S}}_i + \underline{\underline{S}}_j \right)^T \cdot \underline{\underline{SS}}_k \cdot \underline{\underline{SS}}_l^T \cdot \left(\underline{\underline{S}}_i + \underline{\underline{S}}_j \right)}_{=\delta_{ik} \cdot \delta_{li} + \delta_{ik} \cdot \delta_{lj} + \delta_{jk} \cdot \delta_{li} + \delta_{jk} \cdot \delta_{lj}} \right) - \sigma_s^2 = 0$$

$$(86) \quad \Rightarrow f_{ij} = \frac{s_{ij}^2}{2 \cdot (4 - 2 \cdot \beta^2)} \cdot (C_{ii} + C_{ij} + C_{ji} + C_{jj}) - \sigma_s^2 = 0$$

$$(87) \quad \Rightarrow C_{ij} = C_{ji} = \frac{\sigma_s^2 \cdot (4 - 2 \cdot \beta^2)}{s_{ij}^2} - \frac{(C_{ii} + C_{jj})}{2}$$

The formulation of the part yield locus (equation (81)) can be transformed in the form of an interpolation:

$$(88) \quad f = \frac{1}{2} \cdot (s^*)^2 \cdot \underline{\underline{d}}^T \cdot \left(\sum_{i=1}^N \sum_{j=1}^N C_{kl} \cdot \underline{\underline{SS}}_k \cdot \underline{\underline{SS}}_l^T \right) \cdot \underline{\underline{d}} - \sigma_s^2 = 0$$

$$\Rightarrow \frac{1}{(s^*)^2} = \frac{1}{2 \cdot \sigma_s^2} \cdot \sum_{i=1}^N \sum_{j=1}^N C_{kl} \cdot \underline{\underline{d}}^T \cdot \underline{\underline{SS}}_k \cdot \underline{\underline{SS}}_l^T \cdot \underline{\underline{d}}$$

Using the definition of the η -coordinates, this can be written:

$$(89) \quad \frac{1}{(s^*)^2} = \frac{1}{2 \cdot \sigma_s^2} \cdot \sum_{i=1}^N \sum_{j=1}^N C_{kl} \cdot \eta_k \cdot \eta_l$$

For regular domains, according to equation (24), the interpolation can be expressed as a function of the intrinsic coordinates:

$$(90) \quad \frac{1}{(s^*)^2} = \frac{(d^T \cdot S_0)^2}{2 \cdot \sigma_s^2 \cdot \cos^2(\vartheta)} \cdot \sum_{i=1}^N \sum_{j=1}^N C_{kl} \cdot \xi_k \cdot \xi_l$$

This formulation can be compared to the one presented in paragraph C; two particularities should be noticed:

- The interpolation is quadratic and inverse,
- The η -coordinates are used rather than the intrinsic coordinates. This is similar to the hyperplane interpolation. Note that the constant factor $(1/2\sigma_s^2)$ could be integrated in the constants C_{kl} .

4 Tsai criterion

The Hill model can still be improved by the adjunction of a linear function:

$$(91) \quad f = \underline{C}^T \cdot \underline{\sigma} + \frac{1}{2} \cdot \underline{\sigma}^T \cdot \underline{A} \cdot \underline{\sigma} - \sigma_s^2 = 0$$

In comparison with the Hill model containing $N(N+1)/2$ parameters, this model requires N additional yield stresses computed with the Taylor's model. So, in an interpolation domain, beside the $N \underline{S}_i$ and the $N(N-1)/2 \underline{S}_{ij}$ directions, N new other directions must be used.

E Yield locus normal

In order to be able to integrate the interpolation method into a finite element code, the tangent stiffness matrix must be computed. This implies the computation of the yield locus normal. With a classical analytical yield locus formulation, the computation of these normals is well known. For a part yield locus approach, the computation is presented here.

With the assumption of an associated plasticity, the yield locus normal, i.e. the derivative of the yield locus function with respect to the stress vector, is the direction of the plastic strain rate.

Different formulations of the yield locus can appear, three cases are treated:

- The yield locus is a function of s , the length of the stress vector and \underline{d} , its direction:

$$(92) \quad \frac{\partial F}{\partial \underline{\sigma}} = \frac{\partial s}{\partial \underline{\sigma}} \cdot \frac{\partial F}{\partial s} + \frac{\partial \underline{d}}{\partial \underline{\sigma}} \cdot \frac{\partial F}{\partial \underline{d}}$$

With the use of the definition $\underline{\sigma} = s \cdot \underline{d}$:

$$(93) \quad \begin{aligned} \underline{\sigma}^T \cdot \underline{\sigma} = s^2 &\quad \Rightarrow \quad s = \sqrt{\underline{\sigma}^T \cdot \underline{\sigma}} \\ \Rightarrow \frac{\partial s}{\partial \underline{\sigma}} = \frac{1}{2} \cdot (\underline{\sigma}^T \cdot \underline{\sigma})^{-1/2} \cdot 2 \cdot \underline{\sigma} = \frac{\underline{\sigma}}{s} = \underline{d} \end{aligned}$$

and:

$$(94) \quad \frac{\partial \underline{d}}{\partial \underline{\sigma}} = \frac{\partial \left(\frac{\underline{\sigma}}{s} \right)}{\partial \underline{\sigma}} = \frac{1}{s} \cdot \underline{I} - \left(\frac{\underline{\sigma}}{s^2} \right) \cdot \frac{\partial s}{\partial \underline{\sigma}} = \frac{1}{s} \cdot (\underline{I} - \underline{d} \cdot \underline{d}^T)$$

The normal is then:

$$\begin{aligned}
 \underline{n} &= \frac{\partial F}{\partial \underline{\sigma}} = \underline{d} \cdot \frac{\partial F}{\partial s} + \frac{1}{s} \cdot (\underline{I} - \underline{d} \cdot \underline{d}^T) \bullet \frac{\partial F}{\partial \underline{d}} \\
 (95) \quad &= \underline{d} \cdot \left(\frac{\partial F}{\partial s} - \frac{1}{s} \cdot \underline{d}^T \bullet \frac{\partial F}{\partial \underline{d}} \right) + \frac{1}{s} \cdot \frac{\partial F}{\partial \underline{d}}
 \end{aligned}$$

- If the yield locus is a function of the η -coordinates:

$$(96) \quad \frac{\partial F}{\partial \underline{\sigma}} = \frac{\partial s}{\partial \underline{\sigma}} \cdot \frac{\partial F}{\partial s} + \sum_{i=1}^N \left(\frac{\partial \eta_i}{\partial \underline{\sigma}} \cdot \frac{\partial F}{\partial \eta_i} \right)$$

With the definition of the η -coordinates given by equation (22):

$$\begin{aligned}
 \eta_i &= \underline{d}^T \bullet \underline{SS}_i \\
 (97) \quad \Rightarrow \frac{\partial \eta_i}{\partial \underline{\sigma}} &= \frac{\partial \underline{d}^T}{\partial \underline{\sigma}} \bullet \underline{SS}_i = \frac{1}{s} \cdot (\underline{I} - \underline{d} \cdot \underline{d}^T) \bullet \underline{SS}_i = \frac{1}{s} \cdot (\underline{SS}_i - \eta_i \cdot \underline{d})
 \end{aligned}$$

So,

$$\begin{aligned}
 \underline{n} &= \frac{\partial F}{\partial \underline{\sigma}} = \underline{d} \cdot \frac{\partial F}{\partial s} + \frac{1}{s} \cdot (\underline{I} - \underline{d} \cdot \underline{d}^T) \bullet \sum_{i=1}^N \left(\underline{SS}_i \cdot \frac{\partial F}{\partial \eta_i} \right) \\
 (98) \quad &= \underline{d} \cdot \left(\frac{\partial F}{\partial s} - \frac{1}{s} \cdot \sum_{i=1}^N \left(\eta_i \cdot \frac{\partial F}{\partial \eta_i} \right) \right) + \frac{1}{s} \cdot \sum_{i=1}^N \left(\underline{SS}_i \cdot \frac{\partial F}{\partial \eta_i} \right)
 \end{aligned}$$

- In the case of the intrinsic coordinates:

$$(99) \quad \frac{\partial F}{\partial \underline{\sigma}} = \frac{\partial s}{\partial \underline{\sigma}} \cdot \frac{\partial F}{\partial s} + \sum_{i=1}^N \left(\frac{\partial \xi_i}{\partial \underline{\sigma}} \cdot \frac{\partial F}{\partial \xi_i} \right)$$

With the use of equation (24) defining the ξ -coordinates:

$$\begin{aligned}
 \xi_i &= \frac{\eta_i}{\sum_{j=1}^N \eta_j} \\
 (100) \quad \Rightarrow \frac{\partial \xi_i}{\partial \underline{\sigma}} &= \frac{\frac{\partial \eta_i}{\partial \underline{\sigma}} - \xi_i \cdot \sum_{j=1}^N \frac{\partial \eta_j}{\partial \underline{\sigma}}}{\sum_{j=1}^N \eta_j} = \frac{\underline{SS}_i - \eta_i \cdot \underline{d} - \xi_i \cdot \sum_{j=1}^N (\underline{SS}_j - \eta_j \cdot \underline{d})}{s \cdot \sum_{j=1}^N \eta_j} \\
 &= \frac{\underline{SS}_i - \xi_i \cdot \sum_{j=1}^N (\underline{SS}_j)}{s \cdot \sum_{j=1}^N \eta_j} - \underline{d} \cdot \left(\frac{\xi_i}{s} - \frac{\xi_i \cdot \sum_{j=1}^N \eta_j}{s \cdot \sum_{j=1}^N \eta_j} \right) = \frac{\underline{SS}_i - \xi_i \cdot \sum_{j=1}^N (\underline{SS}_j)}{s \cdot \sum_{j=1}^N (\underline{SS}_j^T \bullet \underline{d})}
 \end{aligned}$$

The yield locus normal can be computed:

$$(101) \quad \underline{n} = \frac{\partial F}{\partial \underline{\sigma}} = \underline{d} \cdot \frac{\partial F}{\partial s} + \frac{1}{s \cdot \sum_{j=1}^N (\underline{SS}_j^T \cdot \underline{d})} \cdot \sum_{i=1}^N \left(\left(\underline{SS}_i - \xi_i \cdot \sum_{j=1}^N (\underline{SS}_j) \right) \cdot \frac{\partial F}{\partial \xi_i} \right)$$

For a regular domain, equation (19) can be used:

$$(102) \quad \sum_{j=1}^N (\underline{SS}_j) = \frac{S_0}{\cos(\vartheta)}$$

$$\Rightarrow \underline{n} = \underline{d} \cdot \frac{\partial F}{\partial s} + \frac{1}{s \cdot \underline{S}_0^T \cdot \underline{d}} \cdot \sum_{i=1}^N \left((\cos(\vartheta) \cdot \underline{SS}_i - \xi_i \cdot \underline{S}_0) \cdot \frac{\partial F}{\partial \xi_i} \right)$$

The computation of the yield locus normals leads to an important problem. From a strain rate direction, the Taylor's model computes the corresponding stress. Then, the interpolation leads to a yield locus formulation. From the yield locus, using equation (95), (98), (101) or (102), the yield locus normal is computed. The problem is that this direction does not always coincide with the initial strain rate direction.

It could be possible to reduce this problem by increasing the number of parameters contained in the interpolation (higher order interpolation,...). However, from a numerical point of view, this solution can be worse than the initial problem. Discontinuities between interpolated domains can appear. Indeed, this problem has already been encountered in plate finite elements.

Practically, when one strain rate direction is known, two slightly different ways can be followed. Both present an incompatibility. The first one is computed with the Taylor's model and does not fulfil the normality rule; the second one is computed with the yield locus normal equations and does not fulfil the Taylor's model.

With a global yield locus approach, the continuity of the surface and the continuity of the normals are always fulfilled. With the part yield locus approach, a second problem appears. As already mentioned in paragraph C6, the continuity of the yield locus between adjacent domains can be easily fulfilled due to a common interpolation on the frontier. However, the perfect continuity of the yield locus normals can not be obtained.

In consequence, on the frontier between two domains, two yield locus normals can be computed. The first one comes from equation (95), (98), (101) or (102) applied on the first domain; the second one is computed by the same equation but applied on the second domain. Note that these two contradictory normals are only present on the frontier between adjacent domains contrarily to the first problem where two normals appear inside a domain.

If the two normals on a frontier are divergent, the stress will tend to be localised on the frontier because all the strain rate directions included between the two normals will correspond to a stress on the frontier. On the other hand, a locally non-convex yield locus corresponds to two convergent normals. This case is more inconvenient and leads to numerical instabilities and convergence problems (see the two-dimensional representation of Figure 4).

Our choice is to use the normals computed with Taylor's model for frontier points. So, there is only one normal on the frontier because Taylor's model does not depend on the mathematical choice used for the interpolation. Inside the domain, the normals are interpolated on the base of the normals computed on the domain limit points.

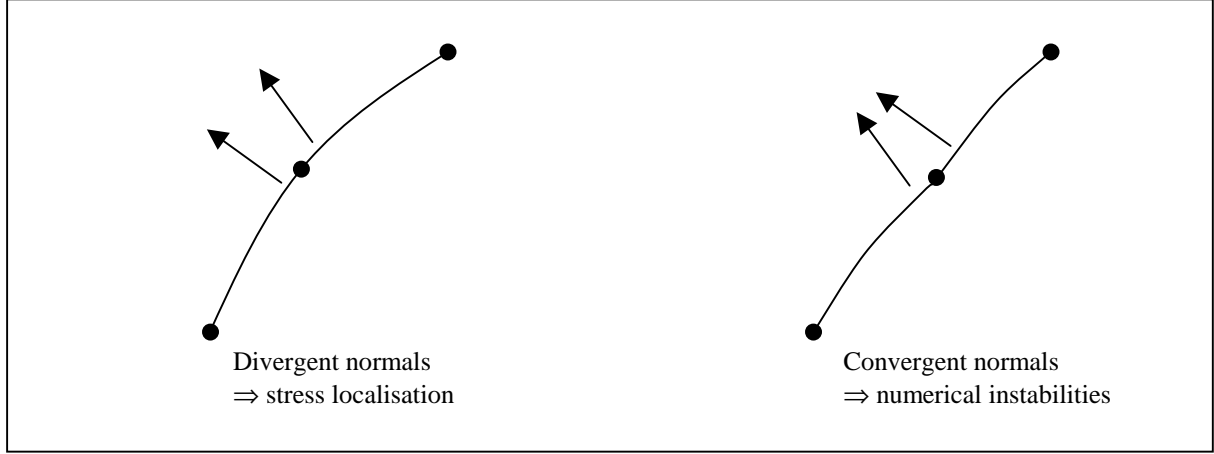


Figure 4: Yield locus normals at the frontier between adjacent domains

F Application of the interpolation to the stress integration

The interpolation method presented in the previous paragraphs can be applied on any variable:

$$(103) \quad \varphi = \sum_{i=1}^N \varphi_i \cdot \eta_i$$

where the φ_i are punctual values computed on the interpolation domain limits and η_i are the interpolation functions. The φ value can either be scalar, vector, tensor... When the domain of validity is updated, the continuity of the interpolated variable is fulfilled; even the direction is continuous in the case of φ vector.

The interpolation formulation will be particularised to the finite element application. The part yield locus represented here does not take work hardening into account. The stress vector $\underline{\sigma}$ and the strain rate direction \underline{u} (unit vector) are the two interesting variables. The formulation for $\underline{\sigma}$ and \underline{u} are then:

$$(104) \quad \underline{\sigma} = \sum_{i=1}^N \underline{\sigma}_i \cdot \eta_i$$

and

$$(105) \quad \underline{u} = \sum_{i=1}^N \underline{u}_i \cdot \eta_i$$

where $\underline{\sigma}_i$ and \underline{u}_i are the values computed on the domain limit points. The Taylor's model computes the stress $\underline{\sigma}_i$ corresponding to the strain rate direction \underline{u}_i for the N vertices of the interpolation domain. The coordinates for the stress $\underline{\sigma}^*$ and the strain rate direction \underline{u}^* are computed with:

$$(106) \quad \eta_i = \underline{\sigma} \underline{\sigma}_i^T \cdot \underline{\sigma}^*$$

$$(107) \quad \eta_i = \underline{u} \underline{u}_i^T \cdot \underline{u}^*$$

for any stress $\underline{\sigma}^*$ or any strain rate direction \underline{u}^* . The $\underline{\sigma}\underline{\sigma}_i$ and the $\underline{u}\underline{u}_i$ vectors are equivalent to the $\underline{S}\underline{S}_i$ vectors defined previously and must fulfil following equations (see also equation(16)):

$$(108) \quad \underline{\sigma}\underline{\sigma}_i^T \bullet \underline{\sigma}_j = \delta_{ij}$$

$$(109) \quad \underline{u}\underline{u}_i^T \bullet \underline{u}_j = \delta_{ij}$$

In the finite element analysis, the 5-dimensional deviatoric stress space and the 5-D strain rate direction space correspond to each other. To one point on the yield locus correspond one stress vector and one strain rate direction. The link between the two spaces is for us the Taylor's model.

Due to this close link between the two spaces, it is assumed that the coordinates computed with equation (106) or (107) are equal if the stress $\underline{\sigma}^*$ physically correspond to the strain rate direction \underline{u}^* . This property is exactly fulfilled on the domain limit directions: the stress $\underline{\sigma}_i$ corresponds to the strain rate direction \underline{u}_i and the coordinates are $\eta_i=1$ and $\eta_j=0$ ($i \neq j$) in both spaces. Inside the domain, the property is assumed to be extended because it is the best approximation and the most convenient.

If we put together equations (104) and (107), we get:

$$(110) \quad \underline{\sigma} = \sum_{i=1}^N \underline{\sigma}_i \cdot \underline{u}\underline{u}_i^T \bullet \underline{u}$$

or

$$(111) \quad \underline{\sigma} = \underline{C} \bullet \underline{u}$$

This equation introducing \underline{C} matrix is a very convenient way to present the interpolation formulation. Just like the Taylor's model, equation (111) allows us to compute the yield stress corresponding to any strain rate direction. Furthermore, this formulation is rather faster than the Taylor's model as only a matrix multiplication is involved.

The definition of the \underline{C} matrix is based on the $\underline{\sigma}_i$ and the $\underline{u}\underline{u}_i$ vectors which are computed on the interpolation domain vertices with the Taylor's model. It should be borne in mind that this \underline{C} matrix is only valid inside the interpolation domain for which it has been computed. If a new domain is explored, a new matrix must be computed.

1 Work hardening

An isotropic work hardening is chosen (see Ref. [11]). The yield locus part described by equation (111) grows uniformly with its shape remaining constant. The work hardening law can be expressed by a scalar formulation:

$$(112) \quad \sigma = K \cdot \left(\varepsilon^0 + \varepsilon^{plastic} \right)^n$$

where σ and $\varepsilon^{plastic}$ are scalars describing respectively the evolution of the yield stress length and the plastic strain length. The strength coefficient K , the offset ε^0 and the hardening exponent n are material parameters fitted on experimental data. For convenience, tensile tests are generally used. In that case, σ and $\varepsilon^{plastic}$ are the stress and the plastic strain along the tensile direction.

According to Taylor's model presented previously, the critical resolved shear stress $\tau_{\pm S}$ can be used to take an isotropic work hardening into account.

In reality, $\tau_{\pm S}$ evolves in a different way for each particular slip system and for each grain as a function of the crystallographic slip γ_s accumulated on that particular slip and of the

dislocation density. The strain rate also has an influence. However, to be able to analytically describe this behaviour, some assumptions have been done:

- The critical resolved shear stress remains equal to a common value for all the slip systems for a particular grain.
- The effect of the work hardening is assumed to be the same for all the slip systems.
- Furthermore, τ_{cS} are the same for all the crystallographic grains at a particular material point. In the finite element analysis, a particular point will be one finite element or more precisely, one integration point of a finite element.

In other words, a common work hardening described by the evolution of τ occurs for all the slip systems of all the grains included in the representative set of crystal orientations associated to an integration point. However, a different value of τ is computed for each integration point. A spatially non-uniform work hardening occurs.

Taylor's model proposes the evolution of the critical resolved shear stress to be expressed as a function of the accumulated polycrystal induced slip Γ :

$$(113) \quad \Gamma(g) = \int_0^t \dot{\Gamma}(g) dt$$

where g represents a particular grain orientation. The total induced slip starts from zero at the initial annealed state and grows as plastic deformations occur.

The first assumption presented above allows us to express the total slip rate as a function of the Taylor's factor M and the strain rate magnitude ($\partial \epsilon_{EQ} / \partial t$):

$$(114) \quad \dot{\Gamma}(g) = \dot{\epsilon}_{EQ} \cdot M(g)$$

The third assumption states that the work hardening is the same for all the grains. The crystallographic orientation g can then be eliminated by averaging:

$$(115) \quad \bar{\Gamma} = \int_0^t \dot{\bar{\Gamma}} dt$$

The average form of equation (114) must then be used:

$$(116) \quad \dot{\bar{\Gamma}} = \dot{\bar{\epsilon}}_{EQ} \cdot \bar{M}$$

where the average of Taylor's factor is only a function of the strain rate direction (the length being taken into account in the strain rate magnitude).

The integration on a time step can be done, assuming that the Taylor's factor remains constant during this time interval:

$$(117) \quad \Delta \bar{\Gamma} = \Delta \bar{\epsilon}_{EQ} \cdot \bar{M}$$

Due to this equation, the plastic equivalent strain describing the evolution of the material behaviour in equation (112) can be replaced by the total induced slip. On the other hand, according to equation (6), the definition of the Taylor's model can be expressed as (see also Ref.[10]):

$$(118) \quad \bar{M} = \frac{\sigma^T \cdot u}{\tau}$$

So, the equivalent stress σ of equation (112) can be replaced by the critical resolved shear stress. The stress-strain curve ($\sigma - \epsilon^{plastic}$) can be converted into a critical resolved shear

stress-induced slip curve (τ - I). The formulation is then (the indices have been removed for convenience):

$$(119) \quad \tau = K \cdot (\Gamma^0 + \Gamma)^n$$

K , I^0 and n are the new material parameters to be fitted to tensile test data.

The interpolation formulation of equation (111) can integrate the work hardening. The new formulation is:

$$(120) \quad \underline{\underline{\sigma}} = \underline{\underline{\tau}} \cdot \underline{\underline{C}} \cdot \underline{\underline{u}}$$

The Taylor's factor can be expressed as a function of the interpolation variables. Starting from its definition (equation (118)):

$$(121) \quad \bar{M} = \frac{(\underline{\underline{\tau}} \cdot \underline{\underline{C}} \cdot \underline{\underline{u}})^T \cdot \underline{\underline{u}}}{\tau} = \underline{\underline{u}}^T \cdot \underline{\underline{C}}^T \cdot \underline{\underline{u}} = \underline{\underline{u}}^T \cdot \underline{\underline{C}} \cdot \underline{\underline{u}}$$

2 Tangent stiffness matrix

As the interpolated model which is used for the computation of the stress has a formulation rather different from the classical approach, the computation of the tangent stiffness matrix is also different. The developments presented hereafter are for the deviatoric part of the stresses and the strains. The hydrostatic part is computed separately and is assumed to have a perfect elastic behaviour.

During a finite element simulation, for a plastic loading, equation (120) representing the yield locus and equation (122) are fulfilled:

$$(122) \quad \frac{\partial \underline{\underline{\sigma}}}{\partial t} = \underline{\underline{C}}^e \cdot \left(\frac{\partial \underline{\underline{\varepsilon}}^{tot}}{\partial t} - \frac{\partial \underline{\underline{\varepsilon}}^p}{\partial t} \right)$$

Equation (122) represents the stress growth and is based on the Hooke's law which links the stress and the elastic strain with the use of the elastic matrix $\underline{\underline{C}}^e$. The time derivatives must be objective; Jaumann formulation is used.

The derivative of equation (120) yields:

$$(123) \quad \frac{\partial \underline{\underline{\sigma}}}{\partial t} = \underline{\underline{C}} \cdot \left(\frac{\partial \underline{\underline{\tau}}}{\partial t} \cdot \underline{\underline{u}} + \underline{\underline{\tau}} \cdot \frac{\partial \underline{\underline{u}}}{\partial t} \right)$$

Using equations (122) and (123):

$$(124) \quad \underline{\underline{C}}^e \cdot \left(\frac{\partial \underline{\underline{\varepsilon}}^{tot}}{\partial t} - \underline{\underline{u}} \cdot \frac{\partial \underline{\underline{\varepsilon}}_{EQ}}{\partial t} \right) = \underline{\underline{C}} \cdot \left(\frac{\partial \underline{\underline{\tau}}}{\partial t} \cdot \underline{\underline{u}} + \underline{\underline{\tau}} \cdot \frac{\partial \underline{\underline{u}}}{\partial t} \right)$$

Where the plastic strain rate has been decomposed into its direction and its length, assuming a constant plastic strain rate direction during a time step. Furthermore, the fact that $\underline{\underline{u}}$ is a unit vector implies that the following property is fulfilled:

$$(125) \quad \underline{\underline{u}}^T \cdot \frac{\partial \underline{\underline{u}}}{\partial t} = 0$$

Equation (124) can then be written:

$$(126) \quad \underline{\underline{u}}^T \cdot \underline{\underline{C}}^{-1} \cdot \underline{\underline{C}}^e \cdot \frac{\partial \underline{\underline{\varepsilon}}^{tot}}{\partial t} - \underline{\underline{u}}^T \cdot \underline{\underline{C}}^{-1} \cdot \underline{\underline{C}}^e \cdot \underline{\underline{u}} \cdot \frac{\partial \underline{\underline{\varepsilon}}_{EQ}}{\partial t} = \frac{\partial \underline{\underline{\tau}}}{\partial t}$$

now, derivating equation (119):

$$\begin{aligned}
 (127) \quad \frac{\partial \tau}{\partial t} &= n \cdot K \cdot (\Gamma^0 + \Gamma)^{n-1} \cdot \frac{\partial \Gamma}{\partial t} = \left(\frac{n \cdot \tau}{\Gamma^0 + \Gamma} \right) \cdot \frac{\partial \Gamma}{\partial t} \\
 &= \left(\frac{n \cdot \tau}{\Gamma^0 + \Gamma} \right) \cdot \bar{M} \cdot \frac{\partial \varepsilon_{EQ}}{\partial t}
 \end{aligned}$$

Then, the equivalent plastic strain rate can be obtained:

$$(128) \quad \frac{\partial \varepsilon_{EQ}}{\partial t} = \frac{\underline{u}^T \cdot \underline{C}^{-1} \cdot \underline{C}^e \cdot \frac{\partial \varepsilon^{tot}}{\partial t}}{\underline{u}^T \cdot \underline{C}^{-1} \cdot \underline{C}^e \cdot \underline{u} + \left(\frac{n \cdot \tau}{\Gamma^0 + \Gamma} \right) \cdot \bar{M}}$$

Replacing this value in equation (122), the stress evolution can be computed:

$$(129) \quad \frac{\partial \underline{\sigma}}{\partial t} = \underline{C}^e \cdot \left(\underline{I} - \frac{\underline{u} \cdot \underline{u}^T \cdot \underline{C}^{-1} \cdot \underline{C}^e}{\underline{u}^T \cdot \underline{C}^{-1} \cdot \underline{C}^e \cdot \underline{u} + \left(\frac{n \cdot \tau}{\Gamma^0 + \Gamma} \right) \cdot \bar{M}} \right) \cdot \frac{\partial \varepsilon^{tot}}{\partial t}$$

The tangent stiffness matrix is then simply:

$$(130) \quad \underline{C}^{\tan} = \underline{C}^e - \frac{\underline{C}^e \cdot \underline{u} \cdot \underline{u}^T \cdot \underline{C}^{-1} \cdot \underline{C}^e}{\underline{u}^T \cdot \underline{C}^{-1} \cdot \underline{C}^e \cdot \underline{u} + \left(\frac{n \cdot \tau}{\Gamma^0 + \Gamma} \right) \cdot \bar{M}}$$

The formulation can be simplified if two new vectors are defined:

$$(131) \quad \underline{a} = \underline{C}^e \cdot \underline{u}$$

$$(132) \quad \underline{b} = \underline{C}^{-T} \cdot \underline{u}$$

Then

$$(133) \quad \underline{C}^{\tan} = \underline{C}^e - \frac{\underline{a} \cdot \underline{b}^T \cdot \underline{C}^e}{\underline{b}^T \cdot \underline{a} + \left(\frac{n \cdot \tau}{\Gamma^0 + \Gamma} \right) \cdot \bar{M}}$$

3 Stress integration

The stress at the end of a finite element time step can not be obtained directly. Several methods with different accuracy levels are presented.

a. **Explicit formulation**

The stress at the end of the time step $\underline{\sigma}_B$ is directly computed from the stress at the beginning of the time step $\underline{\sigma}_A$ with the use of the tangent stiffness matrix:

$$(134) \quad \underline{\sigma}_B = (\underline{I} + \alpha dt) \cdot \underline{\sigma}_A + \underline{C}^{\tan} \cdot \Delta \varepsilon^{tot}$$

Where the first term represents the contribution of the Jaumann derivative.

This is a rather simple formulation but the accuracy is not so good unless very small time steps are used. In order to reduce the error, the integration can also be computed on sub-intervals.

b. Explicit formulation with radial correction

An improvement of the previous method can easily be achieved. The problem with the explicit formulation is that the stress deviates gradually from the exact solution and then from the yield locus. The explicit method can then be used to compute the stress direction:

$$(135) \quad \underline{\sigma}_{approx} = (\underline{I} + \underline{\omega}dt) \bullet \underline{\sigma}_A + \underline{C}^{tan} \bullet \underline{\Delta\varepsilon}^{tot}$$

and a correction will be applied on the stress length in order to stay on the yield locus:

$$(136) \quad \underline{\sigma}_B = k \cdot \underline{\sigma}_{approx}$$

The factor k must be so that the stress at the end of the time step is on the yield locus. The interpolated yield locus formulation of equation (120) will be applied on the state corresponding to the end of the step:

$$(137) \quad \underline{\sigma}_B = \tau_B \cdot \underline{C} \bullet \underline{u}_B \quad \Rightarrow \underline{u}_B = \underline{C}^{-1} \bullet \underline{\sigma}_{approx} \cdot \frac{k}{\tau_B}$$

The final stress lies on the yield locus if the final strain rate direction \underline{u}_B is effectively a unit vector:

$$(138) \quad \underline{u}_B^T \bullet \underline{u}_B = 1 \quad \Rightarrow k = \frac{\tau_B}{\sqrt{\underline{\sigma}_{approx}^T \bullet \underline{C}^{-T} \bullet \underline{C}^{-1} \bullet \underline{\sigma}_{approx}}}$$

The influence of the work hardening is assumed to be small; the critical resolved shear stress at the end of the time step τ_B will then be computed explicitly:

$$(139) \quad \tau_B = \tau_A + \left[\frac{\partial \tau}{\partial \varepsilon_{EQ}} \right]_A \cdot [\Delta\varepsilon_{EQ}]_A$$

where $[\Delta\varepsilon_{EQ}]_A$ can be computed with a formulation similar to equation (128) applied on the beginning of the time step:

$$(140) \quad [\Delta\varepsilon_{EQ}]_A = \frac{\underline{u}_A^T \bullet \underline{C}^{-1} \bullet \underline{C}^e \bullet \underline{\Delta\varepsilon}^{tot}}{\underline{u}_A^T \bullet \underline{C}^{-1} \bullet \underline{C}^e \bullet \underline{u}_A + \left[\frac{n \cdot \tau}{(\Gamma^0 + \Gamma)} \cdot \bar{M} \right]_A}$$

c. Implicit formulation

This method does not give a direct formulation for the stress at the end of the time step; an iterative process has to be used. However, more accurate results are expected.

The integration of the stress has a formulation close to the one of equation (134):

$$(141) \quad \underline{\sigma}_B = (\underline{I} + \underline{\omega}dt) \bullet \underline{\sigma}_A + \underline{C}^e \bullet (\underline{\Delta\varepsilon}^{tot} - \underline{\Delta\varepsilon}^{plastic})$$

However, the tangent stiffness matrix is no more used because it can only be computed on the basis of the beginning of the time step. As this is not interesting, the elastic matrix is chosen.

The plastic strain increment $\underline{\Delta\varepsilon}^{plastic}$ can be decomposed into its length and its direction. Furthermore, its direction is assumed to be between the initial and the final strain rate direction:

$$(142) \quad \underline{\Delta\varepsilon}^{plastic} = \Delta\varepsilon_{EQ} \cdot k \cdot ((1 - \vartheta) \cdot \underline{u}_A + \vartheta \cdot \underline{u}_B)$$

The ϑ -parameter must be chosen in the interval $[0;1]$ and fixes the implicit/explicit ratio. If $\vartheta=0$, the method is explicit; if $\vartheta=1$, the method is fully implicit. An intermediate value is generally chosen.

The k -coefficient must be so that the direction computed between \underline{u}_A and \underline{u}_B is a unit vector:

$$(143) \quad k^2 \cdot \left((1-\vartheta) \cdot \underline{u}_A + \vartheta \cdot \underline{u}_B \right)^T \cdot \left((1-\vartheta) \cdot \underline{u}_A + \vartheta \cdot \underline{u}_B \right) = 1$$

$$\Rightarrow k = \frac{1}{\sqrt{1 + 2 \cdot \vartheta \cdot (1-\vartheta) \cdot \left(\underline{u}_A^T \cdot \underline{u}_B - 1 \right)}}$$

The stress can be written:

$$(144) \quad \underline{\sigma}_B = \left(\underline{I} + \underline{\omega} dt \right) \cdot \underline{\sigma}_A + \underline{C}^e \cdot \Delta \underline{\varepsilon}^{tot} - \Delta \varepsilon_{EQ} \cdot k \cdot \underline{C}^e \cdot \left((1-\vartheta) \cdot \underline{u}_A + \vartheta \cdot \underline{u}_B \right)$$

In this formulation, two unknowns are present: \underline{u}_B and $\Delta \varepsilon_{EQ}$. Two additional equations are then required in order to be able to compute the stress at the end of the time step. The first one is the yield locus formulation applied on $\underline{\sigma}_B - \underline{u}_B$ (equation (120)). The second one is the fact that \underline{u}_B is a unit vector. The vector type equation system obtained is not explicit and must be solved with an iterative method. Note that the hardening variable τ_B will be computed explicitly according to equations (139) and (140).

G Implementation into the finite element code

This paragraph briefly points out some particularities that have been encountered during the implementation of the constitutive law based on the interpolation method.

The constitutive law has been divided into five sub-routines:

- The first one is called PERT. This sub-routine computes the 5 domain limit directions in the strain rate space according to the principle of paragraph B6. The size of the domain which is linked to the angle between the domain limit vectors (parameter β of equation (10)) is the most important parameter required by the sub-routine and must be chosen by the user.
- The second sub-routine is ACPAY3 and is in charge of the Taylor's model calls along the directions computed by PERT. The scaled yield stresses corresponding to particular strain rate directions are determined.
- The third one is called INTERT and achieves the stress interpolation on the basis of the 5 points computed by ACPAY3. If the actual domain is no more valid due to a stress direction evolution, the interpolation tends to become an extrapolation and a new domain must be determined. First, an adjacent domain is investigated according to the method presented in the paragraph B7. If there is no valid adjacent domain, a completely new domain will be computed with 5 new domain limit vectors computed with ACPAY3 along the directions proposed by PERT.
- The fourth sub-routine is SEARCHU and computes the stress at the end of the time step in the elastic-plastic case.
- The last one is called MINTY3 (for Microscopic INTerpolated Yield locus 3 dimensions). This sub-routine is the main part of the constitutive law and uses the 4 other ones to compute the stress at the end of the time step in the general case, the tangent stiffness matrix and the updated state variables. The "3-dimensions" characterise the geometric space associated to solid elements. This law is not linked with axisymmetric state or plane strain state.

The three sub-routines INTERT, SEARCHU and MINTY3 are briefly detailed as they are directly linked to the stress-strain interpolation method.

1 INTERT

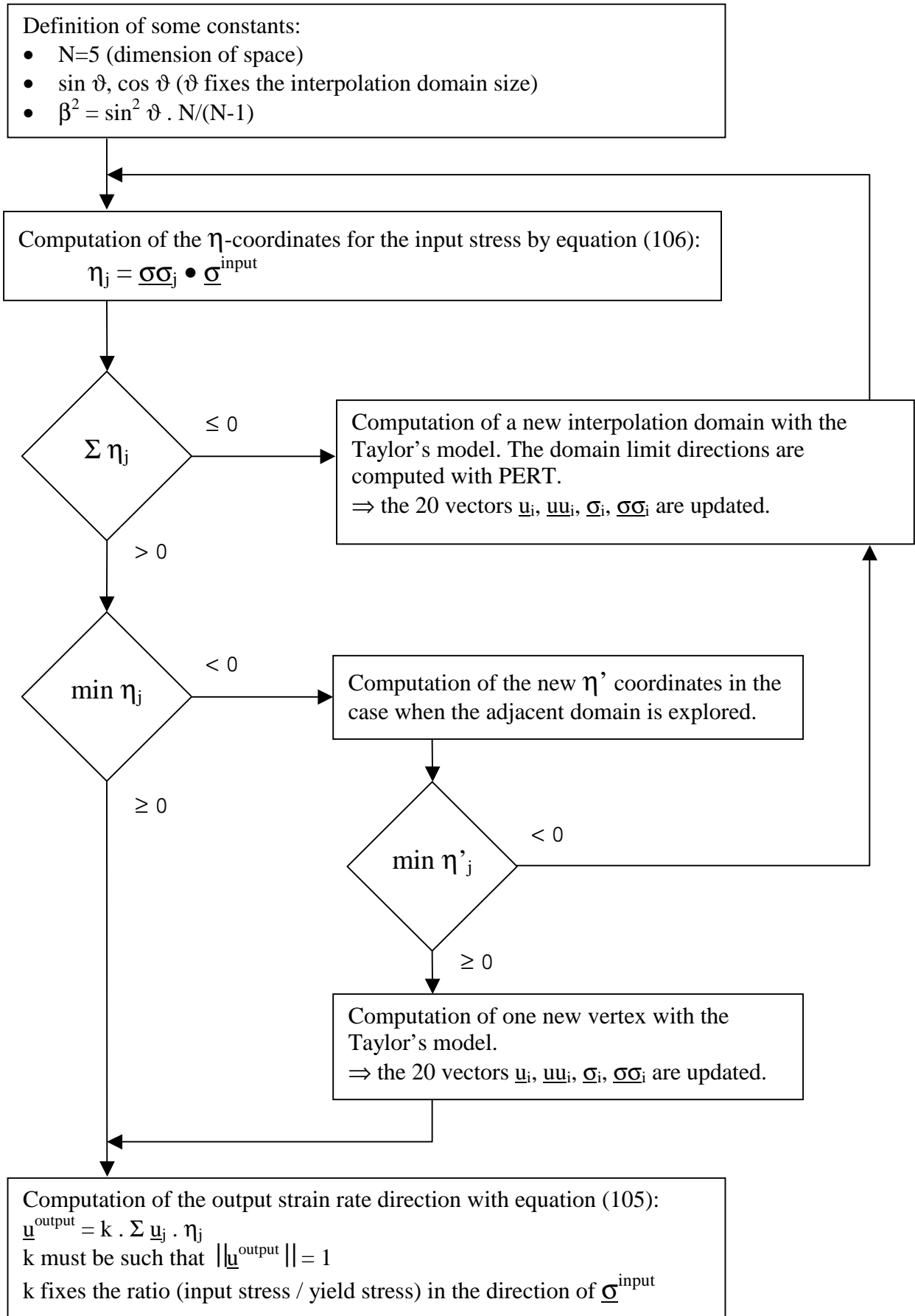
The input variables are the stress vectors that must be compared to the interpolated yield locus. The output will be the ratio of the input stress to the yield stress along the same direction. It is then possible to know whether the input stress is inside, outside or on the yield locus. The corresponding strain rate direction is also computed.

20 working vectors are necessary to achieve the interpolation; they are:

- 5 $\underline{\sigma}_i$ vectors which are the stresses computed by the Taylor's model along the 5 domain limit directions,
- 5 $\underline{\sigma\sigma}_i$ vectors used to compute the η -coordinates and determined with equation (16),
- 5 \underline{u}_i vectors which are the strain rate directions corresponding to the 5 $\underline{\sigma}_i$,
- 5 \underline{uu}_i vectors used to compute the η -coordinates for the strain rates.

Generally, and in particular when it is called by MINTY3, this sub-routine will compute the strain rate direction from the input stress vector. However, the interpolation formulation is not limited to this operation. The inverse operation can be performed: a stress can be obtained from a particular strain rate direction. The sub-routine has been implemented in both cases; the input and the output of the sub-routine can be exchanged.

Flow chart of INTERT



On the flow chart of INTERT, two criteria are used for the updating of the interpolation domain:

- $\sum \eta_j \leq 0$
- $\min \eta_j < 0$

Now, if the output strain rate direction is projected on the \underline{u}_0 direction which is the mean value of the domain limit directions, equations (12) and (21) applied to the strain rates give:

$$(145) \quad \begin{aligned} \underline{u}^{output} \bullet \underline{u}_0 &= \left(\sum_j \eta_j \cdot \underline{u}_j \right) \bullet \underline{u}_0 = \sum_j \eta_j \cdot \underbrace{\underline{u}_j \bullet \underline{u}_0}_{=\cos \vartheta} \\ \Rightarrow \sum_j \eta_j &= \frac{\underline{u}^{output} \bullet \underline{u}_0}{\cos \vartheta} \end{aligned}$$

The first criterion can be written $\underline{u}^{output} \bullet \underline{u}_0 \leq 0$ (because ϑ will always be chosen quite smaller than 90°). This means that the output strain rate direction \underline{u}^{output} forms an angle larger than 90° with the direction \underline{u}_0 , i.e. the direction pointing the middle of the interpolation domain. Obviously, the actual part of the yield locus is not valid for the interpolation and must be completely updated.

It should be noticed that at the beginning of the simulation, the working vectors \underline{u}_i , \underline{uu}_i , $\underline{\sigma}_i$, $\underline{\sigma\sigma}_i$ are not yet computed. During the first call to the sub-routine INTERT, for any input stress vector, the computed η -coordinates will then all be zero. The first updating criterion will then be fulfilled and a first interpolation domain will be computed.

The second criterion is fulfilled when at least one η -coordinate is negative. This means that the input stress direction is outside the actual interpolation domain. This criterion being less strict than the first one, the updating of only one point instead of five will be investigated. It will be checked whether an adjacent domain includes the input stress direction.

In order to check the validity of the adjacent domain, the new coordinates η' corresponding to the new domain must be computed. These η' -coordinates are computed without any Taylor's model call. They are based on the old coordinates and on the properties of the interpolation model. The formulation presented in paragraph B7 is used knowing that regular interpolation domains are used in the strain rate space. According to paragraph B7, the η and η' -coordinates are equal on the frontier between two adjacent domains. However, the explored point is not necessarily on that frontier; it is indeed expected to be inside the new domain. The property of equal coordinates is then not anymore fulfilled and the new coordinates must be computed.

Once the new coordinates are known, the validity of the adjacent domain can be checked. If the new domain is valid, one new Taylor point is computed; otherwise, the classical updating technique is used and five new points are computed.

Finally, the 20 working vectors must be updated; the \underline{u}_i and the $\underline{\sigma}_i$ are computed with the sub-routine PERT and the Taylor's model. The \underline{uu}_i vectors are computed with equation (18) because regular domains are used in the strain rate space. For the $\underline{\sigma\sigma}_i$ vectors, the definition of equation (16) applied for the stresses must be used:

$$(146) \quad \underline{\sigma\sigma}_i^T \bullet \underline{\sigma}_j = \delta_{ij}$$

This equation means that the 5x5 matrix formed by the 5 $\underline{\sigma\sigma}_i$ vectors is the transpose of the inverse of the 5x5 matrix formed by the 5 $\underline{\sigma}_i$ vectors.

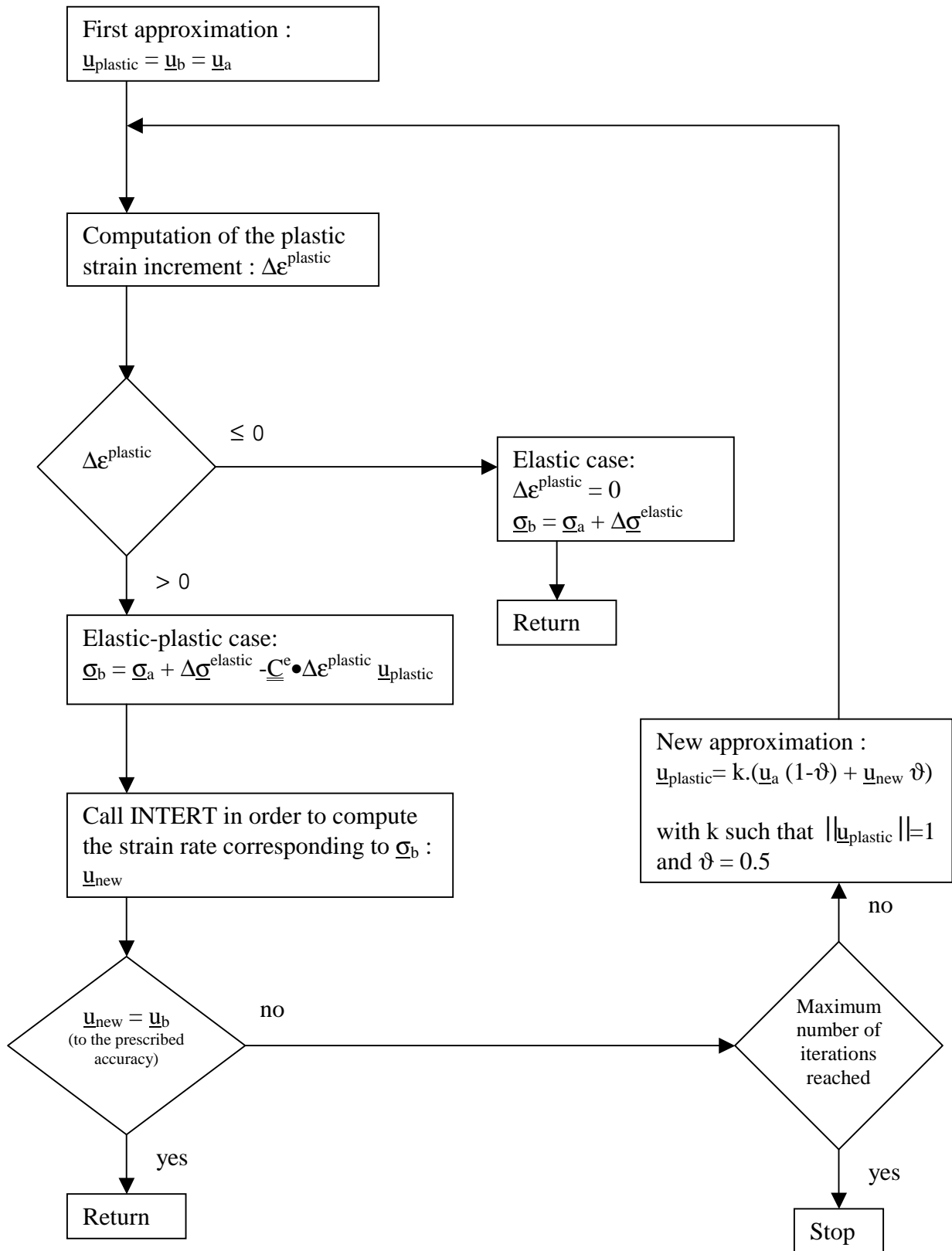
The last step of INTERT's flow chart is the computation of the output values. It should be noticed that the k factor used to norm the strain rate direction fixes the ratio of the input stress to the yield stress. Indeed, to an unit strain rate direction corresponds a yield stress lying on the yield locus. If the input stress is outside the yield locus, the length of the computed strain rate direction will be larger than one. And inversely if the input stress is inside the yield locus.

2 SEARCHU

This sub-routine will compute the stress and the strain rate direction at the end of the time step.

The input variables are the stress vector and the strain rate at the beginning of the time step (state A). The stress increment in a pure elastic hypothesis $\Delta\underline{\sigma}^{elastic}$ including the Jaumann derivatives is also given to the sub-routine.

Flow chart of SEARCHU



The stress integration method presented in this flow chart is based on the implicit method presented in paragraph F3c. An iterative process must be used because the values at the end of the time step are required while their computation is in progress.

The plastic strain increment is computed with equation (128) where the actual strain rate direction, without any other better value, is chosen to be the approximation $\underline{u}_{plastic}$.

The stress at the end of the step is computed according to equation (141) where the strain rate has been decomposed into its length and its direction; this direction being computed with the method proposed by equations (142) and (143).

The first convergence criterion compares the old value of \underline{u} to the new one computed by INTERT according to the actual approximation for stress at the end of the time step. The required accuracy is a parameter which is generally fixed to 10^{-5} . This value means that the scalar product of \underline{u}_B and \underline{u}_{new} must be close to 1, with an accuracy of 10^{-5} . This will correspond to an angle between these two directions of:

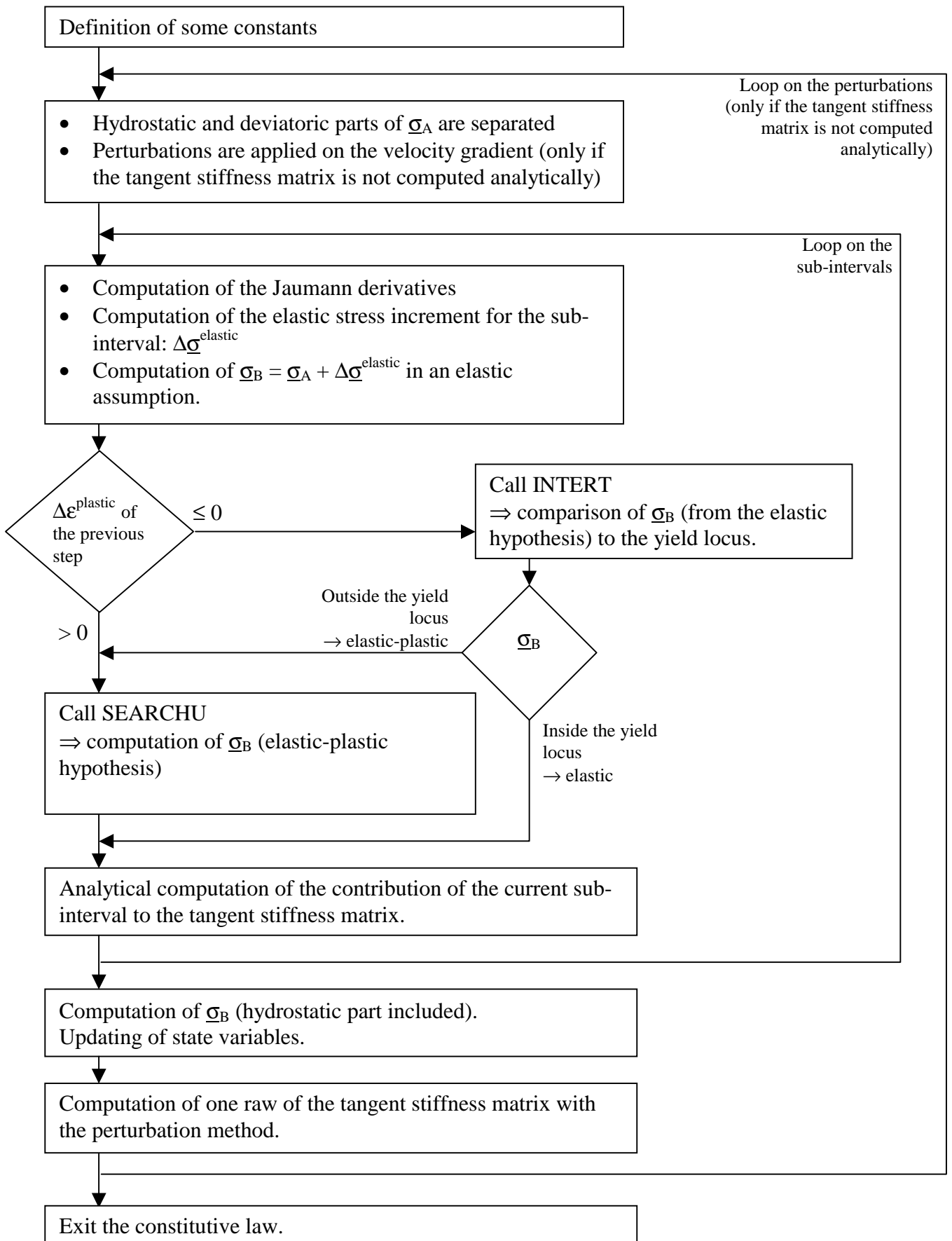
$$(147) \quad \begin{aligned} \cos(\alpha) &= \underline{u}_{new}^T \bullet \underline{u}_B > 1 - 10^{-5} \\ \Rightarrow \alpha &< \arccos(0.99999) = 0.26^\circ \end{aligned}$$

The second convergence criterion fixes the maximum number of iterations for the computation of the final plastic strain rate. This limit number is fixed by the user. If the limit value is reached, the sub-routine is left and the actual time step is reduced.

3 MINTY3

The sub-routine MINTY3 is the constitutive law routine. The input and output parameters are fixed by the finite element code. The input parameters are as classically: the stress and the state variables at the beginning of the time step, the prescribed velocity gradient, the time step and the material parameters. The output of the law are the stress and the state variables at the end of the time step and the tangent stiffness matrix.

Flow chart of MINTY3



A first remark concerning the flow chart of MINTY3 is that the separation of the hydrostatic and the deviatoric part is a well known method. The hydrostatic part is computed elastically due to the assumption of a constant material volume for the plastic deformations. The deviatoric part is extracted from the 3x3 general form of the stress tensor and transformed into a 5-dimensional vector. The constitutive law and the two sub-routines INTERT and SEARCHU only use the vector form for stresses and strains. The conversion is achieved through equation (148):

$$(148) \quad \left\{ \begin{array}{l} \sigma^{5D}(1) = \frac{\sqrt{2}}{2} \cdot (\hat{\sigma}_{11}^{3D} - \hat{\sigma}_{22}^{3D}) \\ \sigma^{5D}(2) = \frac{\sqrt{6}}{2} \cdot (\hat{\sigma}_{11}^{3D} + \hat{\sigma}_{22}^{3D}) \\ \sigma^{5D}(3) = \sqrt{2} \cdot \hat{\sigma}_{23}^{3D} \\ \sigma^{5D}(4) = \sqrt{2} \cdot \hat{\sigma}_{13}^{3D} \\ \sigma^{5D}(5) = \sqrt{2} \cdot \hat{\sigma}_{12}^{3D} \end{array} \right.$$

The work hardening effect described by equation (119) is taken into account during the comparison of the final stress vector $\underline{\sigma}_B$ to the yield locus interpolated with INTERT. It is simply a scalar factor applied on the stress lengths.

Finally, the last remark is about the tangent stiffness matrix. For the perturbation case, a classical method is used: a small perturbation is applied successively on each component of the velocity gradient. Each row of the tangent stiffness matrix will then correspond to one perturbation. The constitutive law is then computed ten times (one without perturbation and nine for the nine perturbations corresponding to the nine components of the velocity gradient) for each integration point and at every time step.

In order to reduce computation time with the elimination of the loop on the perturbations, the analytical tangent matrix can be used. For the elastic-plastic case, the formulation is presented in equation (133) where the \underline{u} -vector will be the one computed by SEARCHU: $\underline{u}_{plastic}$. For the elastic case, SEARCHU is not called; so, $\underline{u}_{plastic}$ is not available. But, in that case, the tangent stiffness matrix is simply the elastic matrix \underline{C}^e .

It should be noticed that this analytical tangent matrix is computed sub-interval per sub-interval. The contribution of each sub-interval is computed separately. The final tangent stiffness matrix is the mean value of all these contributions. This method allows the tangent stiffness matrix to be very close to the stress integration scheme. The resulting convergence of the finite element code is then improved.

Chapter III The Hyperplane Model

The first investigated method to interpolate the yield locus between the points computed by the Taylor's model was the fitting of an hyperplane on the domain limit yield stresses.

The formulation of the hyperplane interpolation was already theoretically developed in Chapter IIC1 and Chapter IID1. In practice, the used hyperplane model is a little bit different in the sense that five hyperplanes are always computed simultaneously. In fact, in addition to the five domain limit vectors, the central point is also computed with the Taylor's model. Six yield stresses are then available; the five hyperplanes are fitted on the central point and on four of the five domain limit vectors.

This method requires one more point computed with the Taylor's model but it is expected to be more accurate. Indeed, five small hyperplanes will better fit a convex surface than one large hyperplane.

Since the normal to a hyperplane is a constant direction, the normality rule would imply that the plastic strain rate direction is fixed for a particular hyperplane. This assumption is not satisfactory and neglects the available information of the normal at the domain limit directions and the central direction. It also induces sudden plastic strain rate discontinuities when the stress direction perforates a new hyperplane or when an updating occurs. So, we chose to improve the model by using a non constant normal which is computed by interpolation between the normal of the five domain limit directions defining the current hyperplane. So, the normality rule is not anymore fulfilled in this model. As already mentioned in Chapter IIE, the interpolation of the normals between the domain limit values instead of computing the normal to the interpolated surface is already used for the stress-strain interpolation in order to improve the continuity between adjacent domains. Here, the advantages of this method are more important because a constant normal on a hyperplane would imply several convergence problem in the finite element code. Only one strain rate direction would then correspond to one hyperplane. As the strain rate evolves continuously during a finite element simulation, the current point on the interpolated yield locus would, except particular case, always be on the frontier between two or more hyperplanes. A localisation of the stress at the frontiers would be noticed. Beside the convergence problems, this would lead to a large number of hyperplane updating and then Taylor's model calls.

In spite of the normal interpolation, some convergence problems have been noticed with the use of the hyperplane method. As it will be presented in the next chapter, these convergence problems are mainly due to a lack of continuity of the interpolated yield locus.

Beside these problems, an excessively high number of Taylor's model calls due to hyperplane updating has been noticed. In order to reduce the number of hyperplane updating, the stress integration scheme was improved. Initially, the stress integration scheme was based on the elastic predictor and radial return method proposed by Kaiping (see Ref. [12]). This method contained two embedded loops. So, in order to reduce computation time, a simplified method with only one loop was developed.

Finally, the elastic predictor was supposed to be at the origin of the numerous Taylor's model calls. This predictor is indeed a very bad assumption of the stress at the end of the step. So, during the stress integration scheme, a hyperplane containing the stress with the elastic predictor must be computed; after the radial return, this hyperplane is no more valid and another one must be computed. That is the reason why a new stress integration scheme was

developed. In this method, a plastic predictor with a proportional return is investigated. The plastic predictor is indeed expected to be very close to the stress at the end of the time step. The same hyperplane will then contain both vectors and no updating will be necessary. These three radial return methods have already been compared in [5].

The sub-routine devoted to the hyperplane methods are:

- In the manner of MINTY3 for the stress-strain interpolation, MIPAY3 is the constitutive law based on the hyperplane interpolation.
- RR1PAY is the sub-routine devoted to the first radial return method (with two embedded loops).
- The second radial return method with only one loop is implemented in the sub-routine RR2PAY.
- The third method with a plastic predictor is in RR3PAY.

One of these three sub-routine is called by MIPAY3 according to the user radial return choice.

- The interpolation is implemented in the sub-routine INTPAY. Just like INTERT for MINTY3, this routine is called by MIPAY3 or one of the radial return sub-routine when one point of the yield locus must be found.
- The sub-routines ACPAY3 and PERT already used by MINTY3, are also called by MIPAY3. The interpolation domain will then have the same geometry for both interpolation method; the geometry being imposed by PERT. It must be noted that the use of the adjacent domain has not been implemented for the hyperplane method.
- All the sub-routine dedicated to the Taylor's model computation are also identical for both methods.

Chapter IV Validation

A *Introduction*

This chapter describes a large number of tests used to check the constitutive law MINTY3. These tests validate this law with respect to theoretical solutions for the simple cases and to other laws for the complex cases where no theoretical solution exists.

The validation will compare not only the computed stresses, displacements and forces with MINTY3 but also the computation time and the observed convergence.

The first test is a simple tensile test in one direction; a tensile test of a holed steel sheet described by over than 500 elements is also investigated. The interpolated yield loci are also compared through π -sections. Finally, tests with any imposed displacements are presented. These tests allow us to point out convergence problems for the constitutive law MINTY3. The results for MIPAY3 are also presented.

B *Tensile test along Y-axis*

This is the most simple finite element simulation: tensile test along the Y -axis on one cubic finite element. The simulation is achieved until the element length along Y is doubled. The strain corresponding to the final state is then: $\varepsilon_Y = \ln(L/L_0) = \ln(2) = 0.69$.

These simulations have been applied on MINTY3 and MIPAY3 in the same conditions. These common conditions are:

- Same initial geometry
- Same imposed displacements
- Same hardening law for the steel
- Same mechanical parameters for the steel
- The texture of the steel is represented by the same set of representative crystals and the same Taylor's model is used in both cases
- Five sub-intervals are used
- The initial time step is 10^{-4}
- Two maximum time step are used for both laws: $2 \cdot 10^{-3}$ and 10^{-2}
- Five different sizes of the interpolated yield locus are checked. They are represented by the angle between the domain limit vectors and the vector pointing the centre of the domain: ϑ . The values are: $1^\circ, 5^\circ, 10^\circ, 20^\circ$ and 30° .
- For MIPAY3, three stress integration schemes are used: the first one is a classical elastic predictor with radial return computed with two embedded loops; the second one is a simplified one with only one loop for the radial return; the third one uses a plastic predictor.

These tests applied to the MIPAY3 law are presented in [5]. This paper compares the three radial return methods and emphasis the influence of the size of the interpolation domain. The conclusions of this paper are that the size of the hyperplane has an important influence on the computed stress as shown in Table 1. Secondly, the plastic predictor gave very good results in terms of computation time, convergence and stability for the tensile tests.

Size of the hyperplanes	1°	5°	10°	20°	30°
Stress at the end of the test (N/mm ²)	559.05	558.3	557.98	534.54	522.54

Table 1: Tensile tests with MIPAY3

For MINTY3, the results are putted together in Table 2.

Tangent stiffness matrix computation	Size of the interpolated yield locus	maximum time step	CPU time (min:sec.1/100 sec)	Maximum number of iterations	Number of time step reductions	Total number of time steps	Total number of iterations	Mean number of iterations per step	Number of total updating of the interpolation domain	Number of updating using the adjacent domain	Final stress along Y-axis (N/mm ²)
An.	1°	2.10 ⁻³	1:50.03	5	0	520	1544	2.97	7	52	553
An.	5°	2.10 ⁻³	42.32	5	0	520	1542	2.97	2	1	553
An.	10°	2.10 ⁻³	40.04	5	0	520	1541	2.96	1	1	554
An.	20°	2.10 ⁻³	38.33	5	0	520	1542	2.97	1	0	557
An.	30°	2.10 ⁻³	37.42	4	0	516	1539	2.98	1	0	558
An.	1°	10 ⁻²	2:47.35	5	2	134	483	3.6	29	41	553
An.	5°	10 ⁻²	22.13	5	0	128	462	3.6	2	1	554
An.	10°	10 ⁻²	17.99	5	0	128	457	3.57	1	1	555
An.	20°	10 ⁻²	17.30	5	0	128	450	3.52	1	0	558
An.	30°	10 ⁻²	16.37	5	0	128	426	3.33	1	0	558
Pert.	1°	2.10 ⁻³	16:05.83	5	0	520	1545	2.97	52	714	553
Pert.	5°	2.10 ⁻³	1:11.55	4	0	516	1534	2.97	2	1	553
Pert.	10°	2.10 ⁻³	1:06.06	4	0	516	1532	2.97	1	1	554
Pert.	20°	2.10 ⁻³	1:29.54	7	0	520	1544	2.97	1	24	557
Pert.	30°	2.10 ⁻³	1:06.95	4	0	516	1539	2.98	1	0	558
Pert.	1°	10 ⁻²	12:07.02	5	2	136	390	2.87	95	367	553
Pert.	5°	10 ⁻²	25.25	4	0	128	370	2.89	2	1	554
Pert.	10°	10 ⁻²	21.35	4	0	128	368	2.88	1	1	555
Pert.	20°	10 ⁻²	43.43	7	0	128	368	2.88	1	24	558
Pert.	30°	10 ⁻²	19.82	4	0	128	375	2.93	1	0	558

Table 2: Tensile tests with MINTY3

A first remark is that this table shows the good quality of the analytical tangent matrix. Indeed, for both analytical and perturbation based matrix, the convergence is very good: acceptable maximum number of iterations (7 in the worst case); low number of time step reduction due to a poor convergence. However, the computation time is largely reduced with the analytical matrix because the loop on the perturbations is eliminated. For the case with the larger computation time, i.e. with the small size of the domain, the reduction is from 16 minutes to 1 minute and 50 seconds and from 12 minutes to 2 minutes and 47 seconds. For the faster cases, the diminution is smaller but the analytical method is always faster.

Due to this, henceforth, the tangent stiffness matrix will always be computed with the analytical method and the resulting convergence will never cause any problem.

The number of updating of the interpolation domain is also presented in the Table 2. This number becomes very high when the size of the domain is small (1°). For the other

values of ϑ , there is only a few updating; this is rather normal in a simple tensile test because the stress direction is expected to be rather constant.

The number of domain updating has a large influence on the computation time. For example, for the case of the tangent stiffness matrix computed with the perturbation method, with a maximum time step of 10^{-2} , when the size of the domain goes from 30° to 1° , the computation time rises from 19.82 to 12:07.02, i.e. it is 36 times higher. This is mainly due to a larger number of Taylor's model calls.

The size of the interpolation domain has also an influence on the results of the simulation. The last column of Table 2 shows the stress along the Y -axis at the end of the simulation. It can be noticed that the stress grows slightly when the size of the interpolation domain grows. As it was expected, the accuracy is lower for the large interpolation domains.

For MIPAY3, large domains induce an underestimation of the computed stress (see Table 1). This behaviour is opposite to the one of MINTY3. This can be related to the different interpolation method for the two laws. The next validation explains this phenomenon.

The stress at the end of the tensile test can also be compared to the theoretical value. Using equation (112) describing the work hardening of the material, the stress at the end of the simulation is a little bit higher than 550 N/mm^2 for a tensile test along the Y -axis.

C π -sections of the yield locus

As the yield locus is defined in a 5-D space, it can not be represented unless a section of it is taken. The classical section in the principal stress plane is not adapted to the deviatoric stress space. As the hydrostatic stresses are not taken into account in our case, we chose to use the π -plane section of the yield locus. This plane is indeed perpendicular to the hydrostatic direction. These considerations lead to a stress tensor having the form of equation (149). The direction of this stress can be defined by the angle α , its length (S_1 or S_2) must be such that it lies on the yield locus; while S_3 must fulfil the deviatoric condition.

$$(149) \quad \underline{\underline{\sigma}} = \begin{pmatrix} S_1 & 0 & 0 \\ 0 & S_2 & 0 \\ 0 & 0 & S_3 \end{pmatrix} \quad \text{with} \quad \frac{S_2}{S_1} = \tan(\alpha)$$

In order to explore the yield locus in all the possible directions of the π -plane, we select the angle α ranging from -90° to $+90^\circ$.

With the hyperplane method, the curves obtained for different sizes of the interpolation domains are presented in Figure 5. The case for domain of 20° is presented separately in Figure 6 with the yield locus normals.

It is clear in that representation of the yield locus that the hyperplanes appear as straight lines due to the projection. Furthermore, as the hyperplane size increases the stress length decreases. This is due to the fact that accuracy is better for small hyperplanes which are closer to the real surface and because the convex curvature of the physical yield locus introduces an underestimation of the stress length. The smallest planes are the most accurate, but they imply frequent Taylor's model calls which requires a lot of computations. The relative mean value of the stress length and the number of calls to the Taylor's model necessary to compute the π -sections are summarised in Table 3.

Size of the hyperplanes	1°	5°	10°	20°	30°
Relative mean stress length	1	0.994	0.977	0.902	0.788
Number of Taylor's model calls	507	109	54	24	12

Table 3: Effects of the hyperplanes size on the computed stress

For the stress-strain interpolation of MINTY3, the π -sections are not a set of straight lines but quite more continuous curves (see Figure 7 and Figure 8), even for large interpolation domain sizes. The continuity of the yield locus leads to a better finite element convergence.

A large interpolation domain is expected to produce a lower number of updating during finite element simulations and to reduce computation time. On the figures, it seems that the accuracy becomes poor for domains of 20° and 30° with the interpolation of MIPAY3 and 40° for MINTY3. With the stress-strain interpolation, for the same accuracy, larger domains can be used; computation time reductions can then be expected with MINTY3.

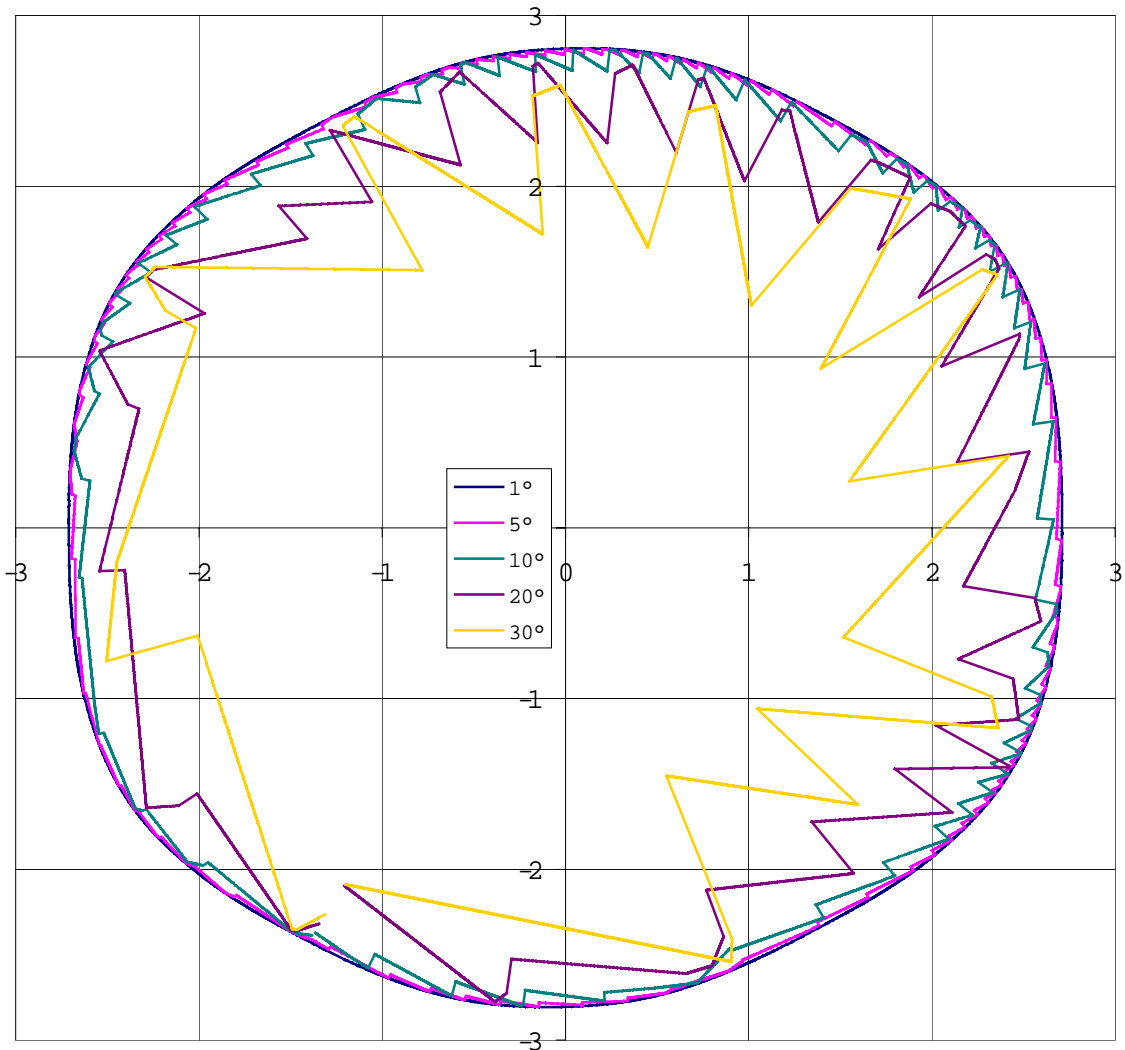


Figure 5: π -sections of the yield locus for the hyperplane method (different sizes of the domain are presented)

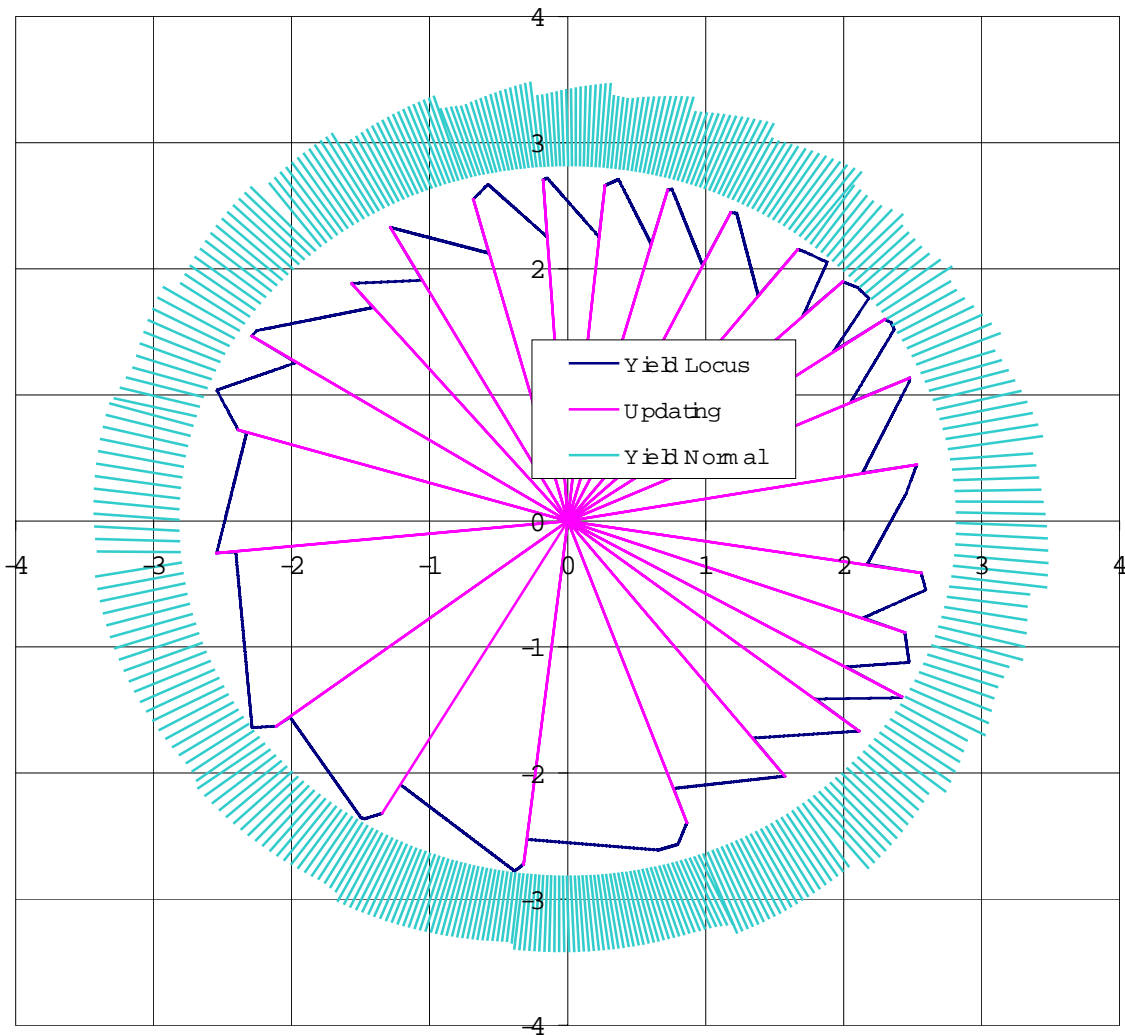
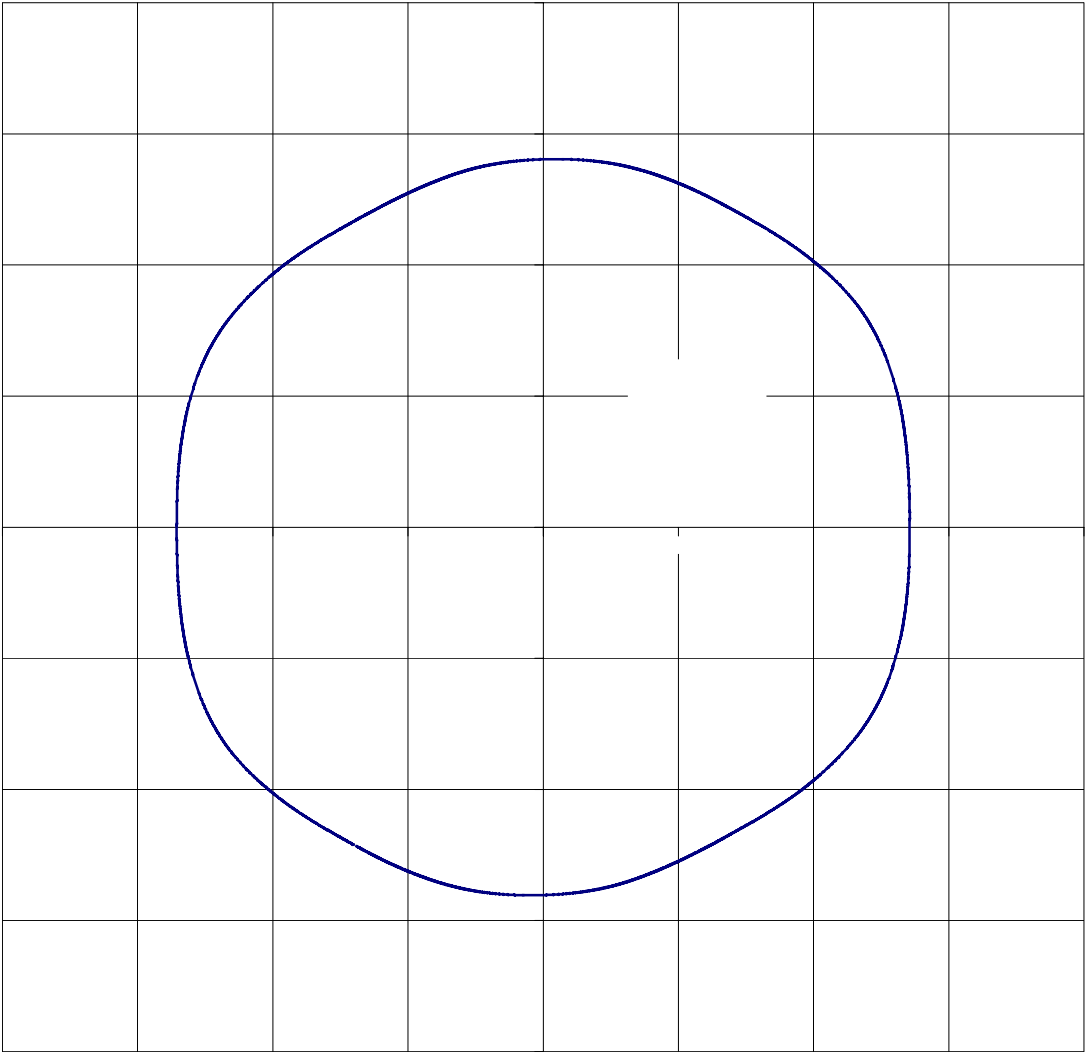


Figure 6: π -sections of the yield locus for the hyperplane method (the size of the domain is 20°)

For the case 40° with MINTY3, the interpolated yield locus is not perfectly superimposed to the reference case (assumed to be the 1° case) but it is continuous. The normals seem also to be continuous. This was not fulfilled with the hyperplane method. Note that the discontinuity observed on Figure 8 is in fact the beginning and the end of the π -section computation.

On Figure 9, the π -sections of the interpolated yield locus with both laws are compared for the smallest size (1°). The two yield locus are superimposed, the normals have also the same directions. This is due to the fact that for small domains, the interpolation method is not important. Only the way to compute the domain limit points, i.e. the Taylor's model has an influence on the resulting yield locus. That is the reason why the 1° case can be used as a reference for the interpolation method.



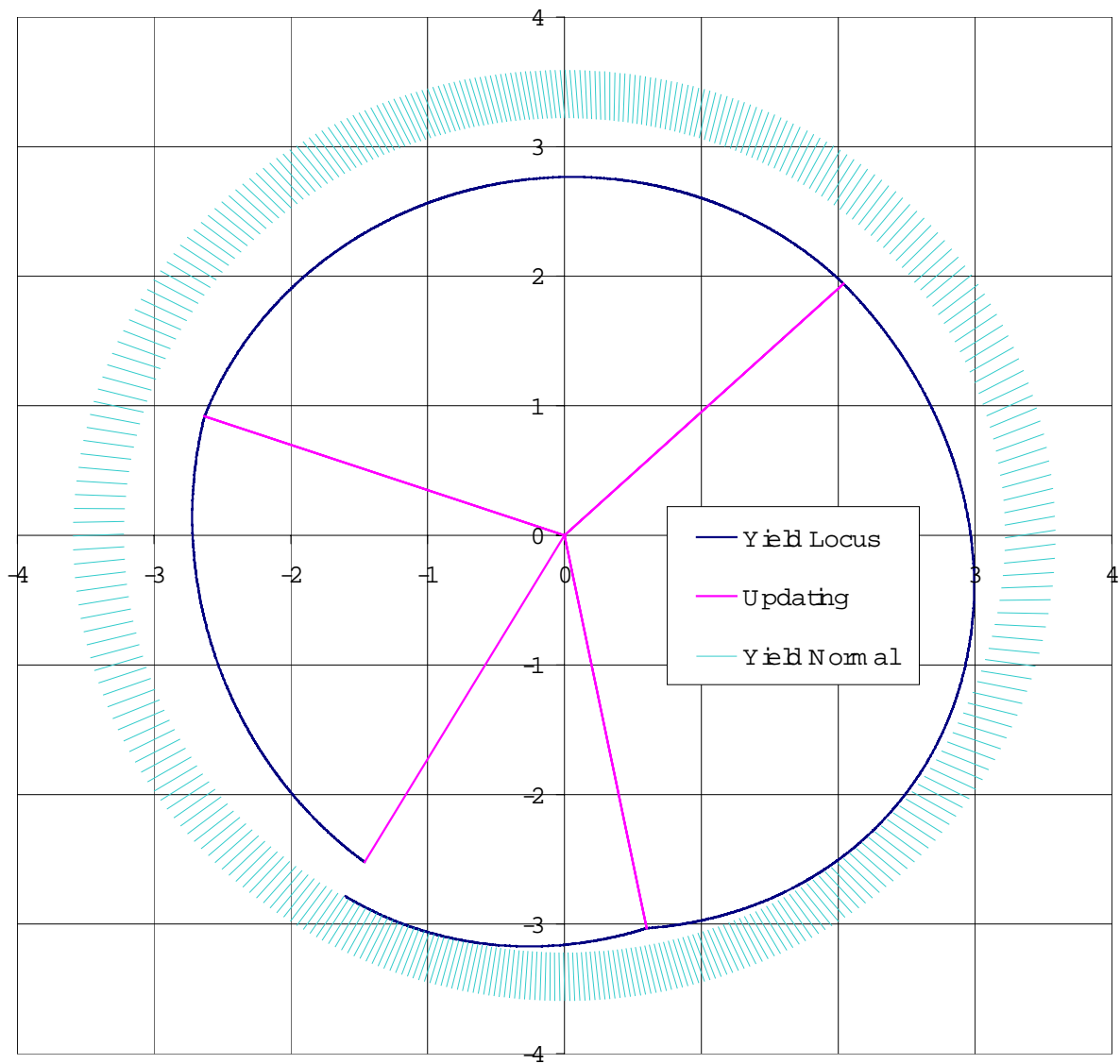


Figure 8: π -sections of the yield locus computed with MINTY3 for domains of 40°

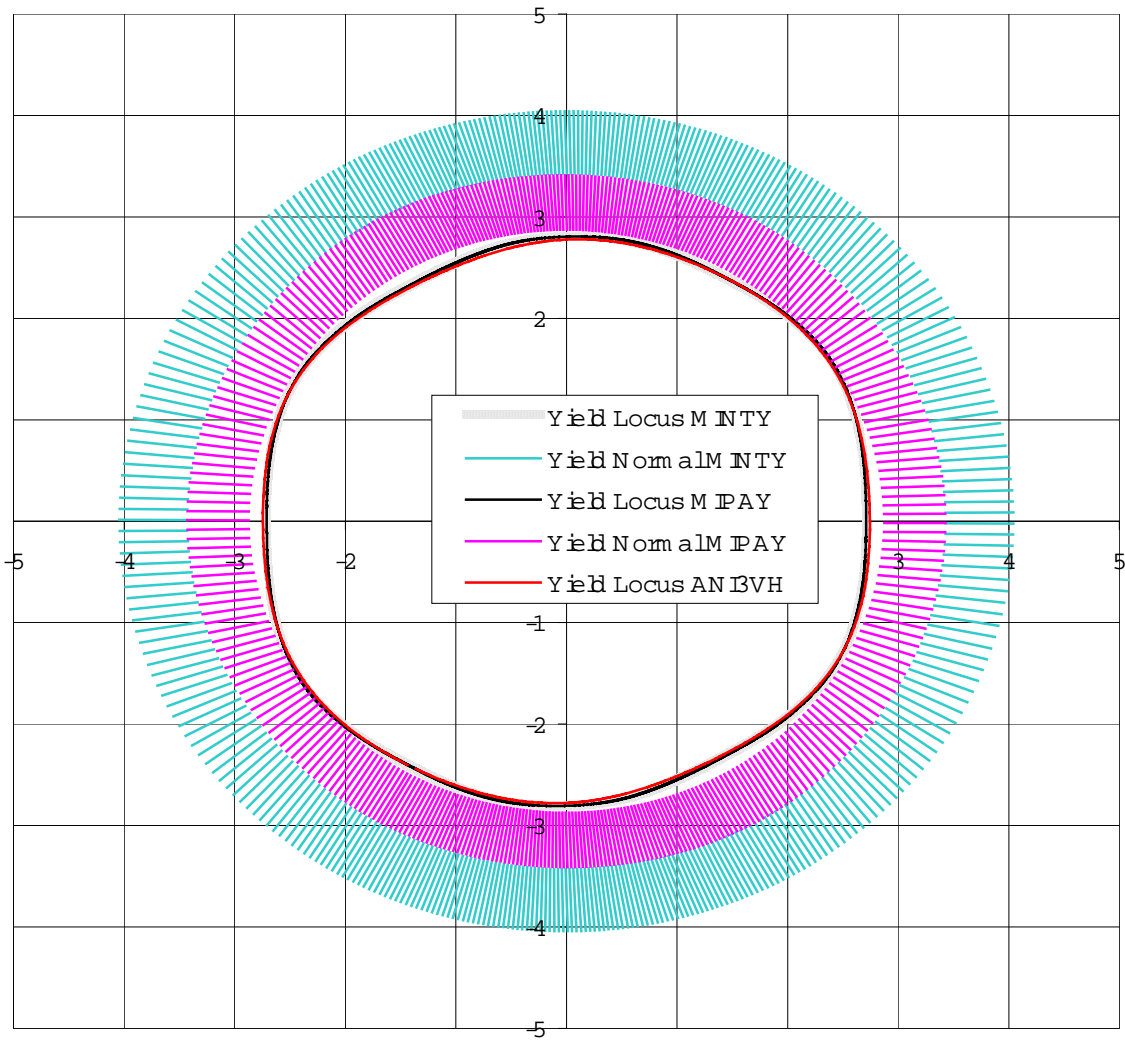
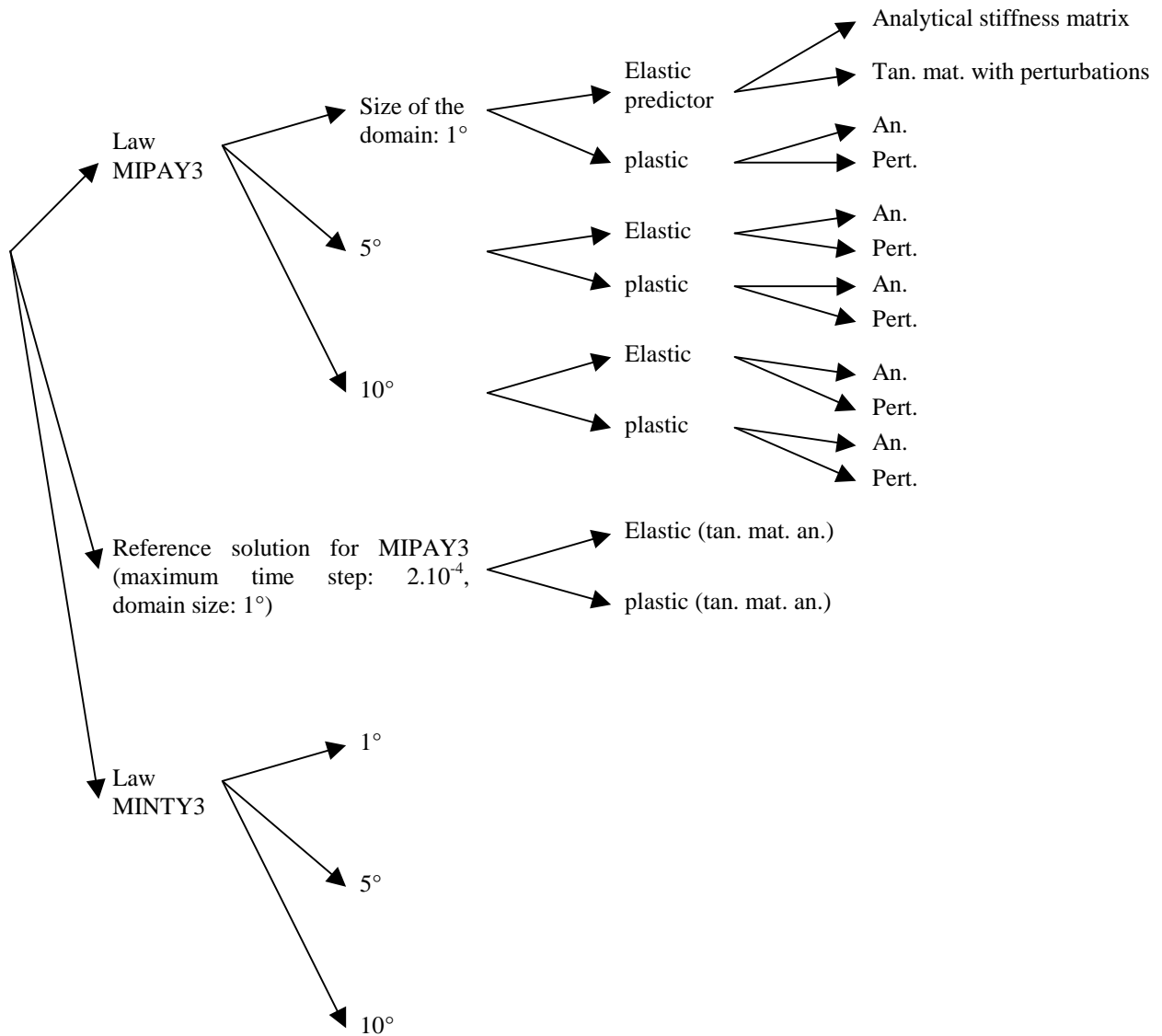


Figure 9: π -sections of the yield locus: comparison of MIPAY3 and MINTY3 for 1° with a global yield locus based on the texture (law ANI3VH)

D Simulation on one finite element

These tests will use the same geometry than for the first validation, i.e. one cubic finite element. However, the imposed strains will be quite different. The imposed displacements imply tensile, compression and shear stresses in the finite element. Eight different cases are investigated. For each of these eight imposed velocity gradients, 17 simulations are achieved:



For these $8 \times 17 = 136$ simulations, the observed convergence was rather good except for some cases with the analytical tangent matrix of the law MIPAY3.

As it has already been noticed earlier, when the size of the interpolation domain is small, the number of updating rises; this leads to a larger number of Taylor's model calls and increase the computation time.

For the law MINTY3, lots of updating make use of the adjacent domain. This can also be seen on Table 2 for the tensile test. The developments presented in Chapter IIB7 are then useful as only one instead of five Taylor's model calls are needed for an updating using the adjacent domain. A significant reduction of the computation time can then be obtained.

Finally, the computed stresses for the 17 different cases for each velocity gradient are compared together. The comparisons can be summarised in the following points:

- The most important and the most frequent errors can be noticed with the use of the plastic predictor of MIPAY3. For tensile tests, as shown in [5], this method has given good results in term of accuracy and computation time. On the contrary, for more complex velocity gradients, the plastic predictor is not so good.
- The influence of the size of the interpolation domain is not as high as expected. This is due to the maximum size of the domain: 10° which is a quite reasonable value. Larger values would imply a larger influence.

E Tensile test of a holed sheet

The validation presented here is a tensile test on a rectangular steel sheet with a hole in its centre. The holed sheet is represented by 766 finite elements; the chosen elements are 8-nodes bricks with a mixed formulation.

Five simulations are presented: the law MINTY3 with domains of 5° , 20° and 40° is used; The hyperplane model with 5° domains is also investigated. As a comparison mean, the law ANI3VH with a global yield locus approach based on the same texture is used.

The computation time are summarised in Table 4.

Constitutive law	MINTY3 (5°)	MINTY3 (20°)	MINTY3 (40°)	ANI3VH	MIPAY3 (5°)
Computation time (hour:min:sec. 1/100 sec)	8:54:32.55	3:11:31.29	2:01:55.29	1:08:30.91	Over than 40 hours!

Table 4: Computation time for the holed sheet tensile test

Remarks:

- The same computer has been used for the simulations with MINTY3 and ANI3VH; the simulation with MIPAY3 has been achieved on several computers (the exact computation time is then not significant).
- For the law ANI3VH, the simulation has been stopped for a lower imposed displacement (6.18 instead of 8 mm of tensile stretching) due to convergence problems.

Here again, the size of the interpolation domain has an important influence on the computation time. Indeed, a ratio higher than 4 is observed between the cases 5° and 40° for MINTY3. The size of 1° has not been investigated due to prohibitive expected CPU time.

In spite of the fact that the simulation was not completely finished, the law ANI3VH seems slightly faster than MINTY3. This is related to the fact that the Taylor's model is not called during the finite element simulation for a global yield locus approach.

Beside computation time, the most significant result is the tensile force applied on the steel sheet as a function of the imposed displacement of the edge of the sheet. This graph for the five simulations is presented in Figure 10.

From this graph, it appears that the results obtained with the law MINTY3 with the size of the interpolation domain equal to 5° are superimposed to the results from ANI3VH. The interpolation method with a reasonable domain size is then validated with respect to a global yield locus approach.

For larger domains (20° and 40°), some small differences can be noticed on the value of the maximum of the tensile force. Before and after that maximum, the results are more or less the same whatever the size of the domain is. The value of the maximum is indeed a very critical variable.

For the law MIPAY3, the results are a little bit different at the maximum and also after the maximum.

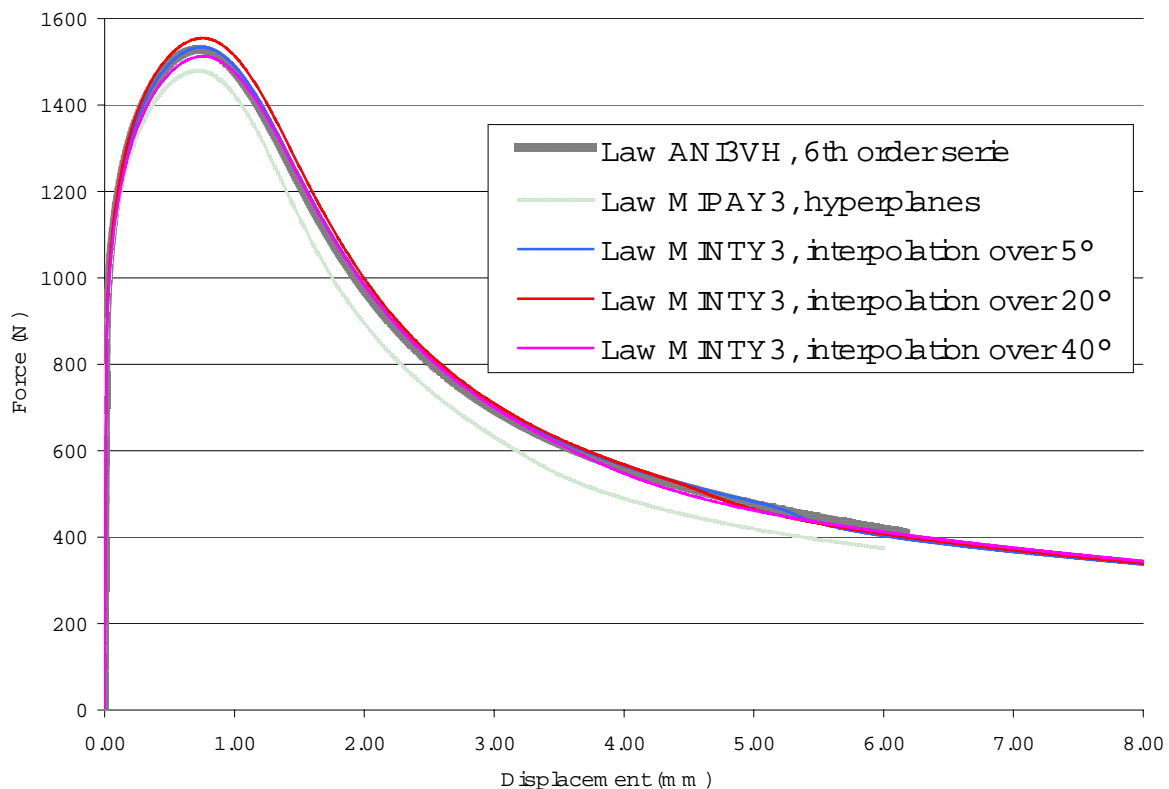


Figure 10: Tensile force applied on the holed steel sheet.

Conclusions

To represent locally the yield locus, regular domains in the strain rate space and non-regular ones in the deviatoric stress space have been defined; their size is linked to the user parameter ϑ .

Two interesting coordinate systems have been defined. These coordinates fulfil the following properties:

- They can easily identify the frontiers of the current domain,
- They are equivalent to the knowledge of the direction of the stress vector from which they are computed,
- Their use permits to separate the direction and the amplitude of the studied vector; amplitude and direction can then be separately interpolated.

The yield locus is defined inside a domain by an interpolation method or a yield locus function. As presented in Chapter IIC and Chapter IID, these two approaches are equivalent. However, interpolation methods are based on a local definition while a yield locus function is more adapted for large yield locus domains, and also for the whole yield locus. Higher order interpolations with a larger number of parameters have also been looked at. However, small linear interpolation domains were expected to be more satisfactory.

Indeed, the four validation tests described in Chapter IV have compared the law MINTY3 to the other laws: MIPAY3 using hyperplanes for the interpolation and ANI3VH using a global yield locus approach.

The results are quite good and the computation time are reasonable for a constitutive law based on the Taylor's model. A good convergence has always been noticed, even with the analytical tangent matrix. This is certainly a consequence of the sub-interval per sub-interval computation of the matrix and also to the continuity of the resulting yield locus thanks to the stress-strain interpolation method.

The first interpolation method that was implemented uses a planar formulation. Unfortunately, it has appeared that hyperplanes can not accurately represent a convex yield locus unless they are numerous and therefore of small size. This has been pointed out on a simple tensile simulation (see Ref. [5]). In order to improve the hyperplane model, three radial return methods have successively been investigated. For the tensile test of [5], the third method with a plastic predictor was well adapted; however, for a more complex simulation, as it was shown in Chapter IVD, the plastic predictor yields inaccurate results.

The considerations brought out in this work show that a particular attention must be paid to the implementation of a local yield locus description in a finite element code. Our first model using hyperplanes was indeed not efficient. The continuity of the yield locus and of the normals are two critical points for the convergence of a finite element simulation. Although the stress-strain interpolation method requires more computation time than a global yield locus approach, it produces interesting results. This is a consequence of an improved theoretical formulation.

It should be noticed that two user parameters have a very important influence on the accuracy and the computation time of the interpolation method. The first one is the number of crystallographic orientations included in the representative set. According to [6], a representative set containing 2000 crystals yields to good results. The second important

parameter is the size of the interpolation domain; this parameter was investigated in Chapter IV. For both parameters, a compromise between accuracy and computation time must be done.

Finally, the goal of the local yield locus approach is the implementation of the texture updating. Larger computation time compared to a global yield locus approach is the price to pay to be able to integrate simultaneous and interacting texture evolution and finite element simulation.

Beside the validation presented in this work, some more complex finite element simulations will be investigated for the Ph. D. thesis. These new validations are:

- Deep-drawing of a cylindrical cup. Finite element results using the MINTY3 law will be compared to experimental results.
- Deep-drawing of a hemispherical cup. This validation is a NUMISHEET'99 and ESAFORM'01 benchmark. Several experimental and numerical results will then be available for the validation.
- Computation of Forming Limit Diagrams (FLD) will also be investigated.

The Ph. D. thesis will also include some improvements of the model: the development of an optimisation method for the texture updating is still in progress.

References

- [1] Berveiller, M., Zaoui, A. (1984), *Modeling of the plastic behaviour of inhomogeneous media*, J. Engng. Mater. Technol., Vol. 106, pp. 295-298.
- [2] Duchêne, L., *Description de la programmation de la méthode d'interpolation de la relation de Taylor*, convention RW n°2748, intermediate report n°30.
- [3] Duchêne, L., *Etude de la précision, de la convergence et de la stabilité de la méthode des hyperplans et de la méthode d'interpolation*, convention RW n°2748, intermediate report n°31.
- [4] Duchêne, L., Godinas, A., Habraken, A. M. (1999), *Metal plastic behaviour linked to texture analysis and FEM method*, Proc. 4th Int. Conf.: NUMISHEET'99.
- [5] Duchêne, L., Habraken, A. M., Cescotto, S. (1998), *Elastoplastic anisotropic model based on texture analysis to simulate steel sheet behavior*, Proc. 7th Int. Symp. (Cancun, Mexico).
- [6] Duchêne, L., Habraken, A. M., Godinas, A. (2000), *Validation of a FEM Model Coupled with Texture Applied to Deep Drawing Process*, Proc. 3rd Int. Conf.: ESAFORM'00.
- [7] Duchêne, L., *Modifications et améliorations apportées à la loi MIPAY3*, convention RW n°2748, intermediate report n°26.
- [8] Godinas, A., *Définition locale de la surface de plasticité – partie1: théorie*, convention RW n°2748, intermediate report n°22.
- [9] Godinas, A., *Définition locale du comportement plastique d'un matériau*, convention RW n°2748, intermediate report n°27.
- [10] Habraken, A. M., *Modelling the plastic anisotropy of polycrystalline materials by FEM*, Grasmec course, February 2000.
- [11] Habraken, A. M., Munhoven, S., *Schema d'implantation d'une version simplifiée du couplage micro-macro: texture – éléments finis*, convention RW n°2748, intermediate report n°10.
- [12] Kaiping, Li (1995-1996), *Contribution to the Finite Element Simulation of Three-Dimensional Sheet Metal Forming*, Ph. D. Thesis, ULG, Liege.
- [13] Munhoven, S., Habraken, A. M., Van Bael, A., Winters, J. (1996), *Anisotropic finite element analysis based on texture*, Proc. 3rd Int. Conf.: NUMISHEET'96, pp.112-119.
- [14] Radu, J.-P., Munhoven, S., Habraken, A. M., *Amélioration de la version simplifiée du couplage micro-macro: texture – éléments finis. Tests de validation*, convention RW n°2748, intermediate report n°11.

- [15] Toth, L.S., Van Houtte, P. (1992), *Discretization techniques for orientation distribution function*, Textures and Microstructures, Vol. 19, pp.229-244.
- [16] Van Houtte, P. (1988), *A comprehensive mathematical formulation of an extended Taylor-Bishop-Hill model featuring relaxed constraints, the Renouard-Wintenberger theory and a strain rate sensitivity model*, Textures and Microstructures, Vols. 8 & 9, pp.313-350.
- [17] Winters, J. (1996), *Implementation of Texture-Based Yield Locus into an Elastoplastic Finite Element Code*, Ph. D. Thesis, KUL, Leuven.