To introduce a new behavior

Summary:

How to introduce a new behavior?
One describes here the addition of a new behavior to solve a nonlinear problem posed on a structure, with STAT_NON_LINE or DYNA_NON_LINE, for all the elements 2D/3D (and multifibre hulls, pipes, beams,…) or to add a new metallurgical behavior in CALC_META.

Crucial steps :

- Writing of the reference material R (equations of the law of behavior)
- Modification of the catalogue of DEFI_MATERIAU (parameters material of the law of behavior)
- Addition of the catalogue python of the relation of behavior
- Choice of the method of integration among the following possibilities:
  - writing of an autonomous routine lc0nn integrating the behavior in a point of integration
  - integration clarifies (ALGO_INTE=' RUNGE_KUTTA') and writing of the associated routines
  - implicit integration in the environment PLASTI, establishment “supplements” (ALGO_INTE=' NEWTON')
  - implicit integration in the environment PLASTI, “easy” establishment (ALGO_INTE=' NEWTON_PERT')
- To produce tests!

Notice : it is also possible to program laws of behavior is in a routine of the type *Umat* (cf [U2.10.01]), that is to say using *MFront* (cf [U2.10.02] and the paragraph 6 this document).

Contents

1 Outline general of resolution in STAT_NON_LINE

Iteration of Newton on the complete system

- on part de la valeur obtenue à l'issue de la phase de prédiction :

$\Delta \mathbf{u}_i^0$

- on calcule la suite d'incréments $(\delta \mathbf{u}_i^{n+1})$

$\mathbf{K}_i^n . \delta \mathbf{u}_i^{n+1} = \mathbf{L}_i^{méca} - \mathbf{L}^{int}(\Delta \mathbf{u}_i^n, \boldsymbol{\sigma}_{i-1}, \boldsymbol{\alpha}_{i-1})$

- on actualise le déplacement $\mathbf{u}$:

$\Delta \mathbf{u}_i^{n+1} = \Delta \mathbf{u}_i^n + \delta \mathbf{u}_i^{n+1}$

- on teste la convergence et on réitère si besoin

Newton
i : n° de pas de temps
n : n° d'itération de Newton

Reference: Principle of resolution (algorithm of Newton) [R5.03.01] and modulates Nonlinear course Aster

au niveau global : l'itération *n* de Newton fournit $\Delta \mathbf{u}_i^n = \Delta \mathbf{u}_i^{n-1} + \delta \mathbf{u}_i^n$

au niveau de l'élément : calcul de $\boldsymbol{\varepsilon}(\Delta \mathbf{u}_i^n)$ **en chaque point de Gauss (PG)**

au niveau du PG : intégration de la loi de comportement

calcul des contraintes et variables internes $\begin{cases} \boldsymbol{\sigma}_{i-1}, \boldsymbol{\alpha}_{i-1} \\ \boldsymbol{\varepsilon}(\Delta \mathbf{u}_i^n) \end{cases} \rightarrow \boldsymbol{\sigma}_i^n, \boldsymbol{\alpha}_i^n$

calcul éventuel de la dérivée de $\boldsymbol{\sigma}$ par rapport à $\boldsymbol{\varepsilon}$ $\begin{cases} \boldsymbol{\sigma}_{i-1}, \boldsymbol{\alpha}_{i-1} \\ \boldsymbol{\varepsilon}(\Delta \mathbf{u}_i^n) \end{cases} \rightarrow \left(\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}\right)_i^n$

seul endroit où intervenir

calcul des forces nodales élémentaires ${}^E(\mathbf{L}^{int})_i^n = \int_{\Omega_E} \mathbf{Q}^T \boldsymbol{\sigma}_i^n \, d\Omega$

calcul éventuel de la matrice tangente élémentaire ${}^E\mathbf{K}_i^n = \int_{\Omega_E} \mathbf{Q}^T \left(\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}\right)_i^n \mathbf{Q} \, d\Omega$

assemblage des forces nodales $\mathbf{L}^{int}(\Delta \mathbf{u}_i^n, \boldsymbol{\sigma}_{i-1}, \boldsymbol{\alpha}_{i-1}) = \sum_E {}^E(\mathbf{L}^{int})_i^n$

calcul du résidu $\mathbf{L}_i^{meca} - \mathbf{L}^{int}(\Delta \mathbf{u}_i^n, \boldsymbol{\sigma}_{i-1}, \boldsymbol{\alpha}_{i-1})$

assemblage éventuel de la matrice tangente $\mathbf{K}_i^n = \sum_E {}^E\mathbf{K}_i^n$

1.1 Options of calculation concerned

1.1.1 Option FULL_MECA (Full Newton)

With the step of time i and with the iteration of Newton n, starting from the internal constraints and variables with preceding balance $(\sigma_{i-1}, \alpha_{i-1})$ and of the increment of deformation $\varepsilon(\Delta \mathbf{u}_i^n)$ (and possibly with variables of order : temperature, hydration,…), calculation in each point of Gauss of each finite element:

- internal constraints and variables (SIEF_ELGA, VARI_ELGA):

$$\begin{cases} \sigma_{i-1}, \alpha_{i-1} \\ \varepsilon(\Delta \mathbf{u}_i^n) \end{cases} \rightarrow \sigma_i^n, \alpha_i^n$$

- of the tangent operator:

$$\begin{cases} \sigma_{i-1}, \alpha_{i-1} \\ \varepsilon(\Delta \mathbf{u}_i^n) \end{cases} \rightarrow \left(\frac{\partial \sigma}{\partial \varepsilon}\right)_i^n$$

This option is calculated if REAC_ITER= m in the command file, and that the number of iteration n is multiple of m (reactualization of the coherent tangent matrix).

1.1.2 Option RAPH_MECA (Newton-Raphson)

With the step of time i and with the iteration of Newton n, starting from the internal constraints and variables with preceding balance $(\sigma_{i-1}, \alpha_{i-1})$ and of the increment of deformation $\varepsilon(\Delta \mathbf{u}_i^n)$ (and possibly with variables of order : temperature, hydration,…), calculation in each point of Gauss of each finite element:

- internal constraints and variables (SIEF_ELGA, VARI_ELGA):

$$\begin{cases} \sigma_{i-1}, \alpha_{i-1} \\ \varepsilon(\Delta \mathbf{u}_i^n) \end{cases} \rightarrow \sigma_i^n, \alpha_i^n$$

This option is calculated if REAC_ITER=0 or REAC_ITER=m in the command file, and that the number of iteration n is not multiple of m.

1.1.3 Option RIGI_MECA_TANG (calculation of the tangent matrix in prediction)

With the iteration 0 step of time i (initialization of the algorithm of Newton), one chooses like tangent matrix of prediction the tangent matrix to preceding balance ($i-1$), that is to say $\mathbf{K}_i^0 = \mathbf{K}_{i-1}$. With to leave the internal constraints and variables to preceding balance $(\sigma_{i-1}, \alpha_{i-1})$, calculation in each points of Gauss of each finite element:

- of the tangent operator in prediction:

$$\sigma_{i-1}, \alpha_{i-1} \rightarrow \left(\frac{\partial \sigma}{\partial \varepsilon}\right)_i^0$$

This option is calculated if REAC_INCR= m in the command file, and that the number of step of time i is multiple of m (reactualization of the tangent operator in prediction).

1.2 Remarks concerning the calculation of the residue and the tangent matrix

1.2.1 Calculation of the residue

The exact calculation of the residue $\mathbf{L}_i^{\text{meca}} - \mathbf{L}^{\text{int}}(\Delta \mathbf{u}_i^n, \boldsymbol{\sigma}_{i-1}, \boldsymbol{\alpha}_{i-1})$ (and thus of the constraints and the internal variables) is fundamental: it guarantees that one will converge towards the solution of the problem. Usmall error in the evaluation of the residue can have serious consequences .

1.2.2 Calculation of the tangent matrix

Matrix tangent known as coherent or consistent (Option FULL_MECA):

Reactualized with each iteration, it ensures the best speed of convergence (quadratic) the algorithm of Newton (figure 1.2.2-1). Its calculation remains however expensive, and if a direct solvor is used, it is necessary to add to the cost of each reactualization that of a factorization. Lastly, for great increments of loading, the coherent tangent matrix can lead to divergences of the algorithm.

Other "tangent" matrices:

One can make errors or approximations in the calculation of the "tangent" matrix: this resulted in degrading the speed of convergence compared to that which is obtained with the coherent tangent matrix reactualized with each iteration, but the solution obtained remains right as long as the residue is calculated in an exact way. There exist several alternatives (methods of quasi-Newton) possible authorized by STAT_NON_LINE (for more details to see [R5.03.01]):

- Elastic matrix (figure 1.2.2-2)
  - calculated only once (economic) starting from the parameters of elasticity
  - recommended in the event of discharge
  - slow but assured convergence
- Reactualized tangent matrix all them i_0 increments of load (figure 1.2.2-3) or all them n_0 iterations of Newton (figure 1.2.2-4)
  - less cost
  - direction less better evaluated
  - diverge sometimes in the zones from strong not linearity

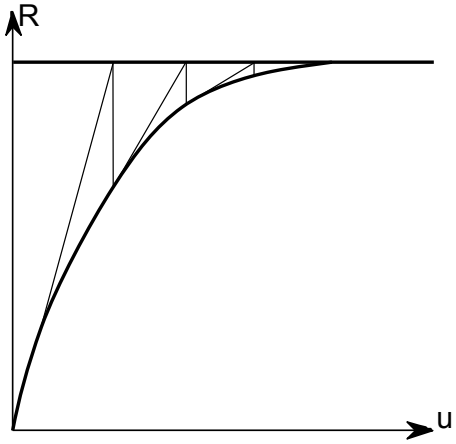


Figure 1.2.2-1: tangent matrix reactualized with each iteration

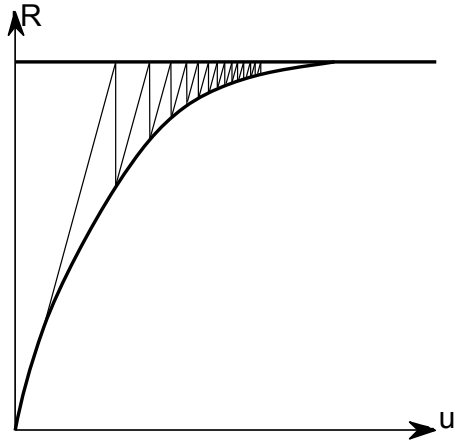


Figure 1.2.2-2: elastic matrix

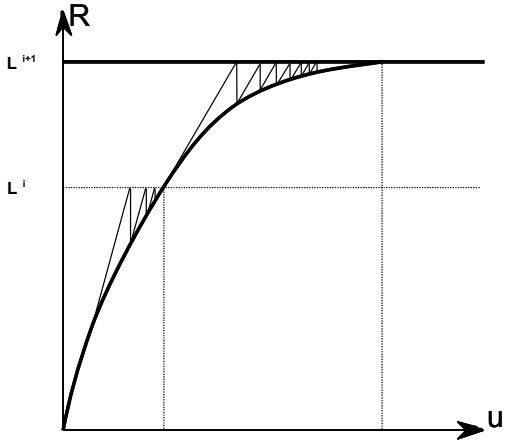


Figure 1.2.2-3: tangent matrix reactualized with each increment of loading

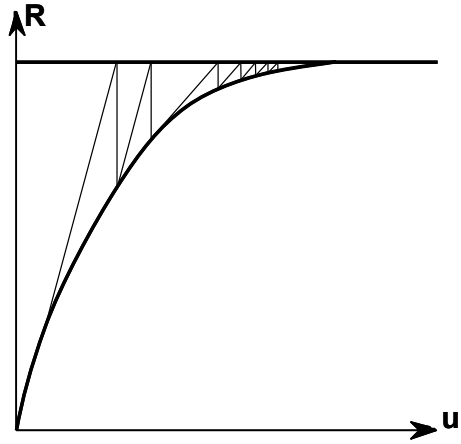


Figure 1.2.2-4: reactualized tangent matrix all two iterations of Newton

2 And choice reference material of the method of integration

The writing of the Reference material is preliminary to the phase of development. The document (*cf.* for example [R5.03.02]) must specify according to the type of modeling concerned (continuous mediums 2D D_PLAN / AXIS and 3D , or models with local plasticity such as the hulls, the plates and the pipes and C_PLAN…):

- the choice of the method of integration (see the various possibilities detailed with § 5);
- equations allowing to calculate the internal constraints and variables;
- equations allowing to calculate the two tangent matrices (options RIGI_MECA_TANG and FULL_MECA).

One can quote other examples of Reference materials:

- [R5.03.04]: *Relations of behavior élasto-visco-plastic of Chaboche.*
- [R5.03.16]: *Elastoplastic relation of behaviour to linear and isotropic work hardening kinematic nonlinear.*
- [R5.03.20]: *Relation of nonlinear elastic behavior in great displacements.*
- [R5.03.21]: *Elastoplastic modeling with isotropic work hardening in great deformations.*

3 Procedure: catalogues

3.1 Modification of the catalogue of DEFI_MATERIAU

The goal of the operator DEFI_MATERIAU [U4.43.01] is to introduce parameters of behavior. These parameters can be common to several relations of behavior (see the example below for the relations of behavior VMIS_ISOT_LINE and VMIS_CINE_LINE).

It is thus necessary hasjouter in the catalogue defi_materiau.py (repertoire code_aster/Catastrophes/Commands) a keyword factor corresponding to the type of behavior which one wishes to introduce, and under this keyword factor, to add the simple keywords representing the parameters of this kind of behavior.

Example:

```
ECRO_LINE = FACT (statut=' f',
    D_SIGM_EPSI = SIMP (statut=' o', typ=' R',),
    SY          = SIMP (statut=' o', typ=' R',),),…
```

mean that the two keywords SY and D_SIGM_EPSI are obligatory for ECRO_LINE (for more precise details, to refer to [U1.03.01]: *Process control supervisor and language*)

3.2 Modification of the catalogue C_RELATION

It is necessary to add to the list returned by C_RELATION () the name chosen for the relation of behavior which one wishes to introduce ('MA_RELATION' in the example below). The catalogue to be modified is c_relation.py (repertoire Code_aster/Catastrophes/Commons).

Example:

```
def C_RELATION (): return (             "ELAS",             #COMMUN#
                                        …
                                        "LAIGLE",
                                        "LEMAITRE",
                                        "LEMAITRE_IRRA",
                                        "LEMA_SEUIL",
                                        "LETK",
                                        "MA_RELATION",
                                        "MAZARS",
                                        "MAZARS_1D",
                                        …
                                    )
```

3.3 To add the catalogue of the law of behavior

This catalogue is to be added in the repertoire bibpyt/Behavior.

3.3.1 Contents

Example: vmis_cine_line.py

```
law = LoiComportement (
    name            = 'VMIS_CINE_LINE',
    lc_type         = ('MECHANICAL',),
    Doc.             = """ Law of Von Mises - Prager with linear kinematic
work hardening [R5.03.02] """,
```

```
        num_lc          = 3,
        nb_vari         = 7,
        nom_vari        = ('XCINXX', 'XCINYY', 'XCINZZ', 'XCINXY', 'XCINXZ',
            'XCINYZ', 'INDIPLAS',),
        mc_mater        = ('ELAS', 'ECRO_LINE',),
        modeling    = ('3D', 'AXIS', 'D_PLAN', '1D',),
        deformation     = ('SMALL', 'PETIT_REAC', 'GROT_GDEP', 'GDEF_LOG',),
        algo_inte       = ('ANALYTICAL',),
        type_matr_tang = ('DISTURBANCE', 'CHECKING',),
        properties      = Nun,
        syme_matr_tang = ('Yes',),
        exte_vari       = Nun,
        deform_ldc      = ('OLD',),
)
```

One thus provides in this catalogue most of the relative information to the behavior:

- `name` : name of the law, identical to that provided for `COMPR_INCR` / `RELATION`
- `num_lc` : number of routine `lc00nn`
- `nb_vari/nom_vari` : many internal variables, and their names (`K8`)
- `mc_mater` : keywords used in `DEFI_MATERIAU`
- `modeling` : possible types of modelings, for the behaviors of continuous mediums: `3D`, `D_PLAN`, `AXIS`, `C_PLAN`, `COMP1D`, `INCO`, `GRADEPSI`, `GRADVARI`,…
- `deformation` : type of possible deformations: 'SMALL', 'PETIT_REAC', 'GROT_GDEP', 'GDEF_LOG', 'GDEF_HYPO_ELAS'.
- `algo_inte` : possible diagrams of integration: implicit (`'ANALYTICAL'`, `'NEWTON_PERT'`…), clarifies (`'RUNGE_KUTTA'`)
- `type_matr_tang` : types of tangent matrices available. In addition to the matrix by disturbance, one can also use the secant matrices, and the combination `TANGENTE_SECANTE`.
- `proprietes` :
- `syme_matr_tang` :
- `exte_vari` : name of the variables of orders taken into account
- `deform_ldc` : type of deformation passed as starter of the law of behavior: `'OLD'`, `'MECHANICAL'` or `'TOTAL'` (*cf.* §4)

Notice S *:*

- *Names of the variabthe interns of the whole of the behaviors are defined in the catalogue python `cata_vari.py` , in order to name in an identical way internal variables of the same significance. This catalogue is available in the repertoire `bibpyt/Behavior`. For a new behavior, it is desirable to re-use already existing names. If new names are added, an error occurs with the execution; it is thus necessary to modify `cata_vari.py` , by justifying its choice during the restitution developments.*
- *The variables of order (or "variable external state") taken into account in the catalogue are specific to certain laws of behavior (see § 5.4.3). For the moment, this list does not describe the generic variables like the temperature or drying.*

Caution *:*

When that one adds a new catalogue, to check well the presence of the map of addition at the top of file. It specifies in which library python to place the file here (`Behavior`):
`#@ ADDITION maloidecomportement Behavior`

3.3.2 Case of a law compatible with kinematics `SIMO_MIEHE`

If the law of behavior is developed to be compatible with a kinematics of the type `SIMO_MIEHE`, besides other kinematics, then two catalogues must be created:

- one first catalogue corresponding to all kinematics except SIMO_MIEHE
- one second corresponding only to kinematics SIMO_MIEHE

The distinction enters the two catalogues is done via the name, on the following model (figure 2 with the 4th position) :
vmis_isot_line.py and vmis2isot_line.py, meta_p_il_pt.py and meta2p_il_pt.py, monocristal.py and mono2ristal.py...

The two catalogues are identical except for attributeS Doc. and deformation :

```
vmis_isot_line.py :
  name            = 'VMIS_ISOT_LINE',
  Doc.            = """ Law of plasticity of Von Mises with linear work
hardening [R5.03.02] """,
  deformation     = ('SMALL', 'PETIT_REAC', 'GROT_GDEP', 'GDEF_LOG',),

vmis2isot_line.py :
  name            = 'VMIS_ISOT_LINE',
  Doc.            = """ Law of plasticity of Von Mises with linear work
hardening for Simo_Miehe [R5.03.02] """,
  deformation     = ('SIMO_MIEHE',),
```

ET possibly of the attribute deform_ldc who phad not to take the value 'MECHANICAL' for the catalogue corresponding to SIMO_MIEHE : it is precisely for to allow other kinematics to function according to the mécaniEMS deform_ldc = 'MECHANICAL' that this system of unfolding of catalogue was set up.

For laws being developed **that** for SIMO_MIEHE , then it is enough to create only one catalogue without modifying its noun, for example: rousselier.py , visc_isot_line.py ...

4 Deformation as starter law of behavior

By definition, the law of behavior makes it possible to calculate the constraints and internal variables starting from the deformations known as “mechanics”, i.e. free of the deformations potentially generated by the variables of order, like the thermal deformations due to the temperature or the withdrawal of desiccation due to drying.

Withvant to carry out integration, it is necessary thus first of all to prepare the “mechanical” deformation with which the law of behavior will work.

Several mechanisms are proposed with the developer law of behavior :

- the preparation of the deformation is made upstream law of behavior, which thus receives directly the mechanical deformation as starter. To profit from this mechanism, the attribute deform_ldc must be specified with the value 'MECHANICAL' in the catalogue of the law.
- the preparation is with the load of the developer of the law of behavior, which thus receives the total deflection as starter. The attribute deform_ldc catalogue is then with the value 'TOTAL'.
- if the law of behavior were not amended since the installation of this system 'MECHANICAL'/'TOTAL', then the attribute deform_ldc is with the value 'OLD'. Laws of behavior in 'OLD' are in the long term intended to disappear.

These various mechanisms are proposed only for the laws interfaced with the routine nmcomp. F90 (operation lc).

Historically, majority of the laws Recevait as starter total deflection. The mechanism deform_ldc = 'MECHANICAL' was set up within the framework ofone modification phase of prediction in the

algorithm of Newton aiming at improving convergence of the code in certain cases. The laws have thus very to gain to work with the mechanism deform_ldc = 'MECHANICAL' since it is possible.

According to the mechanism which the law of behavior follows, a particular term (corresponding to term of order 0 of the linearization constraint on which the algorithm of Newton is based) is expected and must be transmitted through the variable sigp at exit of the law of behavior in phase of prediction.

Let us describe the entries precisely/leftS waited according to the various mechanisms.

4.1 Mechanism deform_ldc = 'MECHANICAL'

In this operation, one has inter alia entryS – exitS following for the law of behavior (i.e. for the routine lc ****. F90 corresponding) :

epsm	Mechanical deformation at the previous moment $i-1$	
LIFO	Mechanical increment of deformation on the step of time $\Delta\epsilon_m$	
sigp	**In prediction** , the term of order 0 following is expected: $\tilde{\tilde{\sigma}}(\epsilon_m^{i-1}, c^i, \Delta t) - \frac{\partial \tilde{\tilde{\sigma}}}{\partial \epsilon_m}(\epsilon_m^{i-1}, c^i, \Delta t)\Delta\epsilon_c$ where $\tilde{\tilde{\sigma}}$ is the constraint expressed like function of the mechanical deformation $\epsilon_m = \epsilon - \epsilon_c$ evaluated at the moment $i-1$ c is it (or them) variable of order, evaluated at the moment i Δt is the increment of time (on which the law depends explicitly in the viscous case) and $\Delta\epsilon_c$ is the increment of deformation due to the variables of order, with $\Delta\epsilon_c = \Delta\epsilon_{c1} + \Delta\epsilon_{c2} + \ldots$ if there are several of them.	**In correction**, one waits classically constraints at the current moment i : $\sigma_i = \tilde{\tilde{\sigma}}(\epsilon_m^i, c^i, \Delta t)$.
dsidep	**EN prediction and in correction:** $\frac{\partial \tilde{\tilde{\sigma}}}{\partial \epsilon_m}(\epsilon_m^{i-1}, c^i, \Delta t)$.	

Perimeter of cover:

This mechanism is not appropriate that when the variables of order generate deformations of the type spherical (proportional to the matrix Identity).

It is available only for kinematics of the small type deformations for which one has additivity of the mechanical deformation and the deformations due to the variables of order $\epsilon = \epsilon_m + \epsilon_c$. It acts in fact of all kinematics except SIMO_MIEHE, that is to say: SMALL, PETIT_REAC, GDEF_LOG and in theory GROT_GDEP, although the mechanism is not for the moment not available for this last kinematics.
That means that law working with SIMO_MIEHE (and thus for the moment GROT_GDEP) does not have to work in deform_ldc = 'MECHANICAL' .

It will be also specified that the assumption of responsibility of the deformations due to the variables of order and the preparation of the mechanical deformation are carried out for the following cases :

- calculation and taking into account of the thermal deformation with a thermal dilation coefficient ALPHA isotropic/anisotropic/transverse isotropic, and différenciable according to the metallurgical phase in the isotropic case;

- calculation and taking into account of the withdrawal of desiccation and the endogenous withdrawal if the parameters material controlling them, K_DESSIC and B_ENDOGE , are isotropic;
- psmall channel in account of the deformations hasnelastic 'EPSAXX' , 'EPSAYY' , 'EPSAZZ' , 'EPSAXY' , 'EPSAXZ' , 'EPSAXZ' .

If the variables of order would generate other deformations and that it is necessary to leave this perimeter (for example, if the generated deformations are not of type spherical), then the mechanism deform_ldc = 'MECHANICAL' cannot be adopted and it is necessary to be directed towards the mechanism deform_ldc = 'TOTAL'.

One will note that Lbe laws of the type MFront and UMAT follow LE mechanism deform_ldc = 'MECHANICAL' .

4.2 Mecanism deform_ldc = 'TOTAL'

This mechanism is proposed for the laws for which the calculation of the deformations due to the variables of order is more complicated than what is currently covered by the mechanism deform_ldc = 'MECHANICAL', so that they can profit all the same from a better prediction.

In this mechanism, one has inter alia the inputs/outputs routine lc ****. F90 corresponding following:

epsm	Deformation total at the previous moment $i-1$	
LIFO	Increment of deformation total on the step of time $\Delta\epsilon$	
sigp	**In prediction** , the developer can turn over the term of order 0 which it considers most relevant. A possibility is the quantity: $\tilde{\sigma}(\epsilon^{i-1}, c^i, \Delta t)$ where $\tilde{\sigma}$ is the constraint expressed like function of the total deflection ϵ evaluated at the moment $i-1$ c is it (or them) variable of order, evaluated at the moment i Δt is the increment of time (on which the law depends explicitly in the viscous case).	**In correction**, one waits classically constraints at the current moment i : $\sigma_i = \tilde{\sigma}(\epsilon^i, c^i, \Delta t)$.
dsidep	**In prediction and in correction:** $\frac{\partial \tilde{\sigma}}{\partial \epsilon}(\epsilon^{i-1}, c^i, \Delta t)$.	

In other words, this mechanism offer the possibility with the developer of communicating second member of its choice with the algorithm of Newton for phase of prediction.

One will insist on the fact that LE developer must here to deal with the preparation of the mechanical deformation inside the law of behavior.

Laws working in SIMO_MIEHE must to follow this mechanism.

4.3 Mechanism deform_ldc = 'OLD'

The value of the attribute deform_ldc = 'OLD' indicate here only that the law was not amended to work with the entries/was leftS described above since the preceding mechanisms were set up. In this mechanism, the term of order 0 is evaluated with the constraints converged at the previous moment (option FORC_NODA), without communication with the law of behavior.
This system is intended to disappear, and a lately developed law should not thus take this value of the attribute.

5 Procedure: routines to be written

It is on this level which must be made the choice of the type of integration. There exist four possibilities:

1. To use the architecture of the environment of integration clarifies by a diagram of Runge-Kutta of order 2 [R5.03.14] (ALGO_INTE=' RUNGE_KUTA'):

   - it is the simplest method. In addition to the recovery of the data materials, it is enough to write a routine calculating the derivative of the internal variables
   - the calculation of the tangent operator is not available under this environment, it is the elastic operator of rigidity which is used

2. Establishment "supplements" new behavior in the implicit environment of integration PLASTI [R5.03.14] (ALGO_INTE=' NEWTON'):

   - resolution of the local nonlinear system by the method of Newton. In addition to the recovery of the data materials, it is necessary to write several routines called in the algorithm of local Newton (evaluation of the threshold, calculation of the residue, analytical calculation of the matrix jacobienne…)
   - The coherent operator is obtained directly starting from the jacobienne of the local system, and the tangent operator in prediction is by default the elastic operator of rigidity
   - PLASTI does not allow to obtain models optimized in time calculation

3. "Easy" establishment of the new behavior in the implicit environment of integration PLASTI with [R5.03.14] (ALGO_INTE=' NEWTON_PERT'):

   - the local non-linear system can be rewritten so that the evaluation of the residue require on behalf of the developer to specify the expression of the derivative of the internal variables written within the framework of method 2 (explicit integration by RK2). One can thus also carry out an implicit integration in PLASTI with the two only routines necessary to explicit integration.
   - The jacobienne of the local system is calculated by disturbance, calculation is thus even more expensive than with method 2. Same manner, the coherent tangent operator is obtained directly starting from the jacobienne of the local system

4. To create an autonomous routine of complete integration of the behavior:

   - often allows to obtain the most powerful models (for example, by reducing the system to be solved with only one scalar equation, nonlinear, to see for example [R5.03.04], [R5.03.16], [R5.03.21],…)
   - require more work "on paper" to optimize the equations

Caution :

In case 4, one will choose a number of routine nn and the routine will be written lc00nn. In the other cases one will choose like entrance point number 32: LC0032 call PLASTI or NMVPRK (Runge-Kutta) according to the value of ALGO_INTE chosen by the user.

5.1 First possibility: to introduce a new explicit behavior – diagram of RUNGE - KUTTA

This kind of integration corresponds to ALGO_INTE=' RUNGE_KUTTA', C'fastest D is the way' of introducing a new behavior.

A routine should be written initially XXXMAT called by the routine of shunting LCMATE in order to recover the parameters materials and the size of the non-linear differential connection to integrate.

```
SUBROUTINE XXXMAT (FAMI, KPG, KSP, MOD, IMAT, NMAT, MATERD, MATERF,
                   MATCST, NDT, NDI, NR, NVI, VIND)
```

Arguments as starter:

```
FAMI, KPG, KSP: family and number of point of gauss/under-point
IMAT          : address of material
MOD           : type of modeling
NMAT          : dimension of MATERD/MATERF
```

Arguments at exit:

```
MATERD        :  coefficients material has T
MATERF        :  coefficients material has t+dt
                 MATERx (*, 1) = characteristic rubber bands
                 MATERx (*, 2) = characteristic plastics
MATCST        :  'yes' if  material has T = material has t+dt
                 'not' if not
NDT           :  total Nb of components of the tensors
NDI           :  Nb of direct components of the tensors
NR            :  Nb of components of the non-linear system
NVI           :  Nb of internal variables
```

One gives below an example for each of the two principal functions which this routine must fill

- **ASSIGNMENT OF DIMENSIONS OF THE LOCAL PROBLEM (NDT, NDI, NR, NVI)**

```
NVI=7
IF (MOD .EQ. '3D') THEN
   NDT = 6
   NDI = 3
   NR  = NDT+2
ELSE IF (MOD .EQ. 'D_PLAN' .OR. MOD .EQ. 'AXIS') THEN
   NDT = 4
   NDI = 3
   NR  = NDT+2
ELSE
   CAL U2MESS ('F',…)
ENDIF
```

- **RECOVERY OF MATERIAL**

```
 NOMC (1) = 'E'
 NOMC (2) = 'NAKED'
 NOMC (3) = 'ALPHA'
 NOMC (4) = 'SY'
 NOMC (5) = 'D_SIGM_EPSI'
 CAL RCVALB (FAMI, KPG, KSP, '-', IMAT, '', 'ELAS', 0,'',
&            0.D0,3, NOMC (1), MATERD (1.1), ICODRE, 1)
 CAL RCVALB (FAMI, KPG, KSP, '-', IMAT, '', 'ECRO_LINE', 0,'',
&            0.D0,2, NOMC (4), MATERD (1.2), ICODRE, 1)
```

A routine should then be written RKDXXX called by the routine of shunting LCDVIN and giving the derivative temporal of the internal variables.

Examples of routine RKDXXX : RKDCHA, RKDVEC, RKDHAY.

5.2 Second possibility: introduction "supplements" of a new behavior in PLASTI (implicit)

This type of integration corresponds to ALGO_INTE=' NEWTON'. Environment PLASTI allows to integrate in a systematic way of the non-linear relations of behavior by a local method of Newton (on the level of the point of Gauss). Knowing the internal constraints and variables at the moment $i-1$ as well as the increment of total deflection $\Delta\boldsymbol{\varepsilon}_i^n$ given by the algorithm of total Newton, the local system of equations to be solved in purely implicit form is written in the following way:

$$R(\Delta\boldsymbol{y})=\begin{pmatrix} g(\Delta\boldsymbol{y}) \\ l(\Delta\boldsymbol{y}) \\ f(\Delta\boldsymbol{y}) \end{pmatrix}=0 \text{ with } \Delta\boldsymbol{y}=\begin{pmatrix} \Delta\boldsymbol{\sigma} \\ \Delta \text{ vari} \\ \Delta p \end{pmatrix}$$

The first equation represents for example the elastic relation stress-strain (6 equations with 6 unknown factors), with $\mathbf{A}$ the operator of elasticity (possibly modified for the laws with damage), $\Delta\boldsymbol{\varepsilon}^p$ variation of plastic deformation and $\Delta\boldsymbol{\varepsilon}^{th}$ thermal variation of deformation:

$$g(\Delta\boldsymbol{y})=\Delta\boldsymbol{\sigma}-\mathbf{A}(\Delta\boldsymbol{\varepsilon}_i^n-\Delta\boldsymbol{\varepsilon}^{th}-\Delta\boldsymbol{\varepsilon}^p)=0$$

the second represents the whole of the laws of evolution of the various scalar and/or vectorial internal variables (n_v scalar equations with n_v unknown factors)

$$l(\Delta\boldsymbol{y})=0$$

the last represents the possible criterion of plasticity (1 equation)

$$f(\Delta\boldsymbol{y})=0$$

This system of $6+n_v(+1)$ equations with $6+n_v(+1)$ unknown factors is solved by a method of Newton:

$$\begin{cases} \dfrac{\partial R}{\partial \Delta y}(\Delta y_k).d[\Delta y_k]=-R(\Delta y_k) \\ \Delta y_{k+1}=\Delta y_k+d(\Delta y_k) \end{cases}$$

With convergence, one thus obtains the increments of constraints and internal variables. The coherent tangent operator as for him is calculated in a systematic way starting from the jacobienne of the local system by the routine LCOPTG (see [R5.03.14] for the detail of the equations). It is thus necessary to program has minima a routine defining the residue (formed by the equations above) as well as a routine building the matrix jacobienne. One describes briefly below the general architecture of PLASTI, by indicating the list of the routines progressively to be written.

General architecture of PLASTI :

```
CAL LCMATE (...)
→ writing necessary of one specific routine XXXMAT of recovery of
  material identical to RUNGE_KUTTA
```

```
IF (OPT .EQ. 'RAPH_MECA' .OR. OPT .EQ. 'FULL_MECA') THEN
  ELASTIC INTEGRATION ON DT
  CAL LCELAS (...)
  → writing possible of one specific routine XXXELA (by default
    LCELIN: linear elasticity)

  ELASTIC PREDICTION STATE WITH T+DT
  CAL LCCNVX (…, THRESHOLD)
  → writing necessary of one specific routine XXXCVX of evaluation of
    threshold

  IF (THRESHOLD .GE. 0.D0) THEN
    CAL LCPLAS (...)
  ENDIF
ENDIF
```

The routine LCPLAS call LCPLNL, which realizes **the loop of Newton** the structure is the following one:

Notations:

YD= (SIGD, VIND) : vector of the unknown factors (of dimension $6+n_v$) at the moment T
YF= (SIGF, VINF) : vector of the unknown factors at the moment T+DT
DY : increment of the vector of the unknown factors between the moments T and T+DT
DDY : increment of vector of the unknown factors between two successive iterations of Newton
R : residue
DRDY : jacobienne

One thus solves: $R(DY) = 0$
By a method of Newton $DRDY(DYK)\ DDYK = -R(DYK)$
$DYK+1 = DYK + DDYK \quad (DY0\ BEGINNING)$
and one reactualizes $YF = YD + DY$

```
CALCULATION OF THE INITIAL OUTPUT SOLUTION D TEST OF SYSTEM NL OUT OF DY
CAL LCINIT (… DY,…)
→ writing possible of one specific routine XXXINI (by default DY is
  initialized with 0)

ITERATIONS OF NEWTON
ITER = 0
1 CONTINUOUS
  ITER = ITER + 1

INCREMENTING OF  YF = YD + DY
  CAL LCSOVN (NR, YD, DY, YF)

CALCULATION OF THE TERMS OF THE SYSTEM WITH T+DT = - R (DY)
  CAL LCRESI (…, DY, R, IRET)
→ writing necessary of one specific routine XXXRES of calculation of
  residue

CALCULATION OF THE JACOBIEN OF THE SYSTEM WITH T+DT = DRDY (DY)
  if ALGO_INTE=' NEWTON' exact calculation of the jacobienne
    CAL LCJACB (… DY,… DRDY, IRET)
    → writing necessary of one specific routine XXXJAC
  if not if ALGO_INTE=' NEWTON_PERT', calculation by disturbance (cf § 5.3)
    CAL LCJACP (… DRDY,…)

RESOLUTION OF LINEAR SYSTEM DRDY (DY) .DDY = - R (DY)
  CAL LCEQMN (NR, DRDY, DRDY1)
```

```
   CAL LCEQVN (NR,  R,  DDY)
   CAL MGAUSS ('NCWP', DRDY1, DDY, NR, NR, 1, RBID, IRET)

DY REACTUALIZATION = DY + DDY
   CAL LCSOVN (NR, DDY, DY, DY)

LINEAR RESEARCH in case ALGO_INTE=' NEWTON_RELI'
   CAL LCRELI (…)

ESTIMATE OF CONVERGENCE
   CAL LCCONV (DY, DDY, NR, ITMAX, TOLER,…, R,…, IRTET)
   → writing possible of one specific routine XXXCVG of the criterion of
     convergence (relative criterion by default in LCCONG)
   IF (IRTET.GT.0) GOTO 1

CONVERGENCE - > INCREMENTING OF  YF = YD + DY
CAL LCSOVN (NDT+NVI, YD, DY, YF)

UPDATE OF SIGF, VINF
CAL LCEQVN (NDT,   YF (1)      , SIGF)
CAL LCEQVN (NVI-1, YF (NDT+1), VINF)
```

In short, for an introduction "supplements" of a new behavior in PLASTI, necessarily should be written the following specific routines:

- XXXMAT called by LCMATE : recovery of material and the size of the local problem
- XXXCVX called by LCCNVX : evaluation of the threshold
- XXXRES called by LCRESI : calculation of the residue
- XXXJAC called by LCJACB : calculation of the jacobienne

It can also be useful, according to the need, to write the following specific routines:

- XXXELA called by LCELAS : elastic integration (if non-linear elasticity)
- XXXINI called by LCINIT : initialization (for an initialization other than DY0=0)
- XXXCVG called by LCCONV : to modify the convergence criteria

5.3 Third possibility: use of the routines of integration clarifies in an implicit integration with PLASTI

5.3.1 Principle

This last case corresponds to ALGO_INTE_=' NEWTON_PERT', it is the method D'I"easy" mplantation of the new behavior in the implicit environment of integration PLASTI. It is possible to use the two routines directly XXXMAT and RKDXXX (recovery of the data material, and derived from the internal variables) used with ALGO_INTE_=' RUNGE_KUTTA' to carry out an implicit integration. Indeed, the system of differential equations solved by RUNGE_KUTTA can be written:

$$\begin{cases} \Delta\sigma = \mathbf{A}\left(\Delta\varepsilon_i^n - \Delta\varepsilon^{th} - \Delta\varepsilon^p(\boldsymbol{Y})\right) \\ \dfrac{d\boldsymbol{Y}}{dt} = F(\boldsymbol{Y}, t; \sigma) \end{cases}$$

where $\boldsymbol{Y}$ represent the whole of the internal variables of the model. The relation between the tensor of the constraints and the elastic part of the tensor of the deformations is generally linear, but can be evaluated in a nonlinear way by a specific expression.

Once programmed the routine RKDXXX allowing to calculate $\frac{d\boldsymbol{Y}}{dt}=F(\boldsymbol{Y},t;\boldsymbol{\sigma})$, it is possible to use it for an implicit integration, which consists in solving (*cf.* [R5.03.14]):

$$R(\Delta\boldsymbol{Z})=0=\begin{bmatrix}R_1(\Delta\boldsymbol{Z})\\R_2(\Delta\boldsymbol{Z})\end{bmatrix}\text{, with }\Delta\boldsymbol{Z}=\begin{pmatrix}\Delta\boldsymbol{\sigma}\\\Delta\boldsymbol{Y}\end{pmatrix}=\boldsymbol{Z}(t+\Delta t)-\boldsymbol{Z}(t)$$

- The first system of equations represents the elastic relation stress-strain
  $$R_1(\Delta Z)=\boldsymbol{A}^{-1}\boldsymbol{\sigma}-(\Delta\boldsymbol{\varepsilon}_i^n-\Delta\boldsymbol{\varepsilon}^{th}-\Delta\boldsymbol{\varepsilon}^p(\boldsymbol{Y}))=\boldsymbol{A}^{-1}\boldsymbol{\sigma}-G(\boldsymbol{Y})=0$$
  By convention, first values of $\boldsymbol{Y}$ represent the variation of plastic deformation, to facilitate the calculation of $G(\boldsymbol{Y})$ (see the routine LCRESA for more details)
- The second expresses the laws of evolution of the various internal variables, that is to say after temporal discretization by a diagram of implicit Euler:
  $$R_2(\Delta Z)=\Delta Y-\Delta t\,.\,F(Y,\boldsymbol{\sigma})=0$$

This system is solved by the method of Newton suggested in the environment PLASTI and described in the preceding paragraph:

$$\begin{cases}\dfrac{\partial R}{\partial\Delta Z}d(\Delta Z_k)=-R(\Delta Z_k)\\ \Delta Z_{k+1}=\Delta Z_k+d(\Delta Z_k)\end{cases}$$

Quantities G and F intervening in the residue are calculated by the routine "clarifies" RKDXXX to write, and the residue is built automatically by the routine LCRESA. The matrix jacobienne is calculated automatically by disturbance (routine LCJACP). The coherent tangent operator as for him is calculated in a systematic way starting from the jacobienne (routine LCOPTG, to see [R5.03.14] for the detail of the equations).

In short, this process allows, with the two only routines necessary to explicit integration, (coefficients material and calculation of the derivative of the internal variables) to use an implicit integration, and to profit from a tangent matrix. This process is economic in terms of time of development, but *a priori* less effective in time CPU than a matrix jacobienne programmed explicitly.

One details below the calculation of the tangent operator by disturbance, as well as the convergence criteria of the algorithm of local Newton.

Calculation by disturbance (LCJACP): finished differences in order 2

- Initialization of the disturbance: $\eta=10^{-7}\|\Delta Z\|$
- buckle on the columns j matrix to be filled:
  - calculation of $R(\Delta Z+\eta I_j)$ with $I_j=[0\ \ 0\ \ \ldots\ \ 1\ \ \ldots\ \ 0\ \ 0]^T$ null vector except with the line j
  - calculation of $R(\Delta Z-\eta I_j)$
  - calculation of the column j : $\left[\dfrac{\partial R}{\partial\Delta Z}\right]_{\ldots,j}\simeq\dfrac{R(\Delta Z+\eta I_j)-R(\Delta Z-\eta I_j)}{2\eta}$

Convergence criteria (LCCONG):

The two blocks are separated R_1 and R_2 residue to avoid the problems due to different orders of magnitude. With each iteration k algorithm of local Newton, one calculates:

$err_1 = \frac{\left\| R_1(\Delta Z_k) \right\|_\infty}{\left\| R_1(\Delta Z_0) \right\|_\infty}$, with $R_1(\Delta Z_0) = \Delta \varepsilon_i^n$ because one has $\Delta Z_0 = 0$ with initialization. (cf § 5.2)

$$err_2 = \frac{\left\| R_2(\Delta Z_k) \right\|_\infty}{\left\| Y(t) + \Delta Y_k \right\|_\infty}$$

The criterion of stop is then the following:

$max(err_1, err_2) < \xi$, where ξ is given by RESI_INTE_RELA.

5.3.2 Example

An example: the law visco-élasto-plastic of Hayhurst.

Routine of reading of the coefficients material (called by the routine of shunting LCMATT):

- HAYMAT

Routine of calculation of the derivative of the internal variables (called by the routine of shunting LCDVIN):

- RKDHAY

The developments clarifies/implicit this relation of behavior are tested and compared in the CAS-test ssnv225 [V6.04.225]. It is about a test at the material point of creep in great deformations making it possible to validate the capacities of the model of HAYHURST to represent primary education, secondary and tertiary creep. Here characteristics of execution for this test in version 11.2:

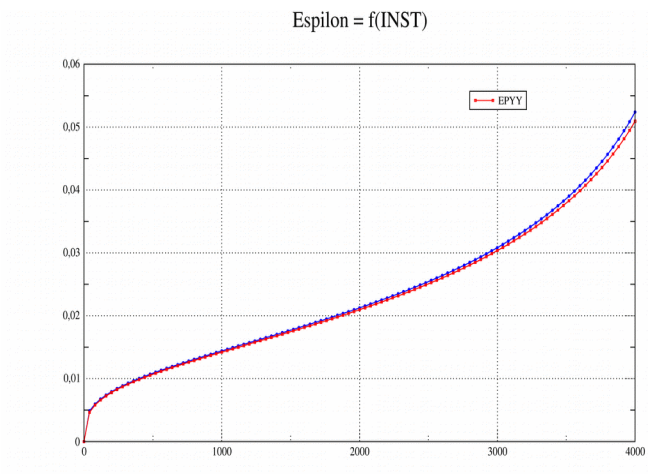
Modeling a: ALGO_INTE=' RUNGE_KUTTA'

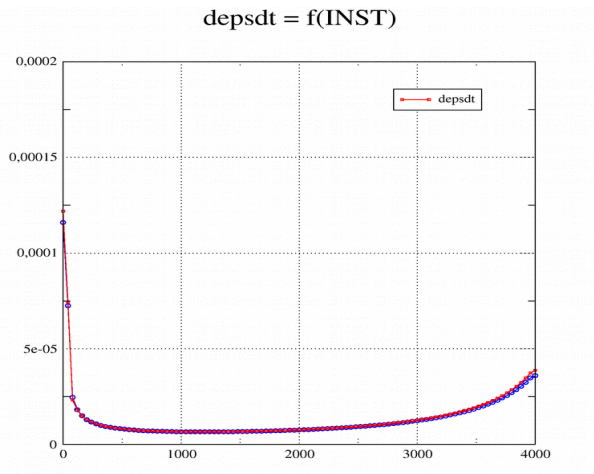
- time CPU: 90.59 S
- Many steps of time: 1660
- Iteration count of Newton: 7615

Modeling b: ALGO_INTE=' NEWTON_PERT' :

- time CPU: 27.38 S
- Many steps of time: 520
- Iteration count of Newton: 1404

The results are nearly identical:





5.4 Fourth possibility: to write a routine lc00nn autonomous

5.4.1 lc00nn : routine relative to a point of integration of an element, specific to a law of behavior

To seek in the repertoire bibfor/algorith a number of routine lc00nn not used $(50< nn< 100)$) and to start from this empty routine.

Notice :

The call to lc00nn by lc0000 (which is the routine calling all the routines of integration of the behavior available in Code_hasster) is already written. However, it should be made sure that the arguments (as well as the order of these arguments) selected with the declaration of lc00nn by the developer are the same ones as with the call in lc0000.

Arguments of entry of a routine lc00nn are *has minimum* :

FAMI family of points of gauss (RIGI, FARMHOUSE,…)
KPG, KSP number of the point of gauss and the under-point
NDIM dimension of L spaces ($3d=3$, $2d=2$, $1d=1$)
IMATE address of material
COMPOR information on the behavior
compor (1) = relation of behavior (vmis_cine_…)
compor (2) = number internal variables
compor (3) = type of deformation (small, green…)
CRIT criteria buildings
crit (1) = number D maximum iterations has convergence (ITER_INTE_MAXI)
crit (3) = value of the tolerance of convergence (RESI_INTE_RELA)
INSTAM moment T
INSTAP moment T = T + dt
EPSM total deflection has t- (or possibly the gradient of transformation according to the type of deformation: the arguments of the appealing routine, lc0000, contain the dimension of EPSM, and LIFO),
LIFO increment of total deflection (even notices)
SIGM constraint has T
VIM internal variables has T
OPTION option of calculation
RIGI_MECA_TANG - > dsidep (T)
FULL_MECA - > dsidep (t+dt), sig (t+dt)
RAPH_MECA - > sig (t+dt)
ANGMAS three angles (nautical or of Euler) of the massive mot_clef
TYPMOD type of modeling: 3D, AXIS, D_PLAN, C_PLAN,…
ICOMP meter for the local recutting of the step of time
NVI many internal variables of the behavior

the arguments of exit are, according to the option of calculation :

VIP internal variables has the current moment (options RAPH_MECA and FULL_MECA)
SIGP term of order 0 in the linearization of the constraint (option RIGI_MECA_TANG) – cf. §4
constraints has the current moment (options RAPH_MECA and FULL_MECA)
DSIDEP coherent tangent operator or of speed (option FULL_MECA or RIGI_MECA_TANG).
CODRET code return allowing to indicate (if it is nonnull) a problem of integration
local, therefore to carry out one recutting of the step of time

Remarks :

- *LE case falling due, it is possible to also use a derived type as starter (variable `BEHinteg` in `LC0000/LCnnnn`). This type contains additional arguments, for example a characteristic length in the case of the nonlocal models…*
- *In the same way, it is possible to transfer from additional arguments at exit of the routine `LCnnnn/LC0000` via the variable `BEHinteg`.*
- *Withttention, in the case `RIGI_MECA_TANG`, the tables relating to the constraints and internal variables at the end of the step of time are not allocated. They thus should not be used to calculate the tangent matrix of prediction.*

5.4.2 Organization of the routine to be written

One takes as with § 3.3 the example of the law of Von Mises with linear kinematic work hardening `VMIS_CINE_LINE` (`num_lc=3` in `vmis_cine_line.py`):

```
 SUBROUTINE LC0003 (FAMI, KPG, KSP, NDIM, IMATE, COMPOR, CRIT, INSTAM,
&                   INSTAP, EPSM, LIFO, SIGM, VIM, OPTION, ANGMAS, SIGP, VIP,
&                   PLUG, TYPMOD, ICOMP, NVI, DSIDEP, CODRET)
```

This routine is in fact a routine of shunting for the behaviors `VMIS_CINE_LINE` and `VMIS_ECMI_*`. The integration of `VMIS_CINE_LINE` is realized in the routine `NMCINE`, called by `lc0003` and whose contents are described here briefly.
It will be noted that in this example, the law follows the mechanism `deform_ldc = 'OLD'`, and that for this reason, the thermal deformations are calculated and cut off inside the law (cf. §4).

- **READING OF THE ELASTIC CHARACTERISTICS OF THE MATERIAL (TIME T = +)**

```
 NOMRES (1) = ' E'
 NOMRES (2) = ' NU'
 CAL RCVALB (FAMI, KPG, KSP, '+', IMATE, '', 'ELAS', 0,'',
&            0.D0,2, NOMRES, VALRES, ICODRE, 2)
 E  = VALRES (1)
 NAKED = VALRES (2)
```

Note:

`RCVALB` is a general routine allowing to interpolate the values of the coefficients materials compared to the variables of orders of which they depend (see the utilities).
`RCVARC` is a routine which makes it possible to recover the value of a variable of order (temperature, drying, irradiation,…) at the moment considered, and the point of Gauss considered (see the utilities). Example:

```
CAL RCVARC ('', 'TEMP',' -', FAMI, KPG, KSP, MT, IRET)
CAL RCVARC ('', 'TEMP', '+', FAMI, KPG, KSP, TP, IRET)
```

- **READING OF THE CHARACTERISTICS OF WORK HARDENING**

```
 NOMRES (1) = ' D_SIGM_EPSI'
 NOMRES (2) = ' SY'
 CAL RCVALB (FAMI, KPG, KSP, '+', IMATE, '', 'ECRO_LINE', 0,'',
&            0.D0,2, NOMRES, VALRES, ICODRE, 2)
 DSDE=VALRES (1)
 SIGY=VALRES (2)
 C  = 2.D0/3.D0*DSDE/(1.D0-DSDE/E)
```

- **CALCU L OF THE ELASTIC CONSTRAINTS AND THE CRITERION OF VON MISES**

```
      C 110 K=1,3
        DEPSTH (K)   = LIFO (K) - EPSTHE
        DEPSTH (K+3) = LIFO (K+3)
110   CONTINUOUS
      EPSMO = (DEPSTH (1) +DEPSTH (2) +DEPSTH (3))/3.D0
      C 115 K=1, NDIMSI
        DEPSDV (K)   = DEPSTH (K) - EPSMO * KRON (K)
115   CONTINUOUS
      SIGMO = (SIGM (1) +SIGM (2) +SIGM (3))/3.D0
      SIELEQ = 0.D0
      C 114 K=1, NDIMSI
        SIGDV (K) = SIGM (K) - SIGMO*KRON (K)
        SIGDV (K) = DEUXMU/DEUMUM*SIGDV (K)
        SIGEL (K) = SIGDV (K) + DEUXMU * DEPSDV (K)
        SIELEQ   = SIELEQ   + (SIGEL (K) - C/CM*VIM (K))** 2
114   CONTINUOUS
      SIGMO = TROISK/TROIKM * SIGMO
      SIELEQ = SQRT (1.5D0*SIELEQ)
      THRESHOLD  = SIELEQ - SIGY
      DP = 0.D0
      PLASTI=VIM (7)
```

- **CALCULATION OF THE CONSTRAINTS AND THE INTERNAL VARIABLES**

Expressions of the constraints and the internal variables (RAPH_MECA and FULL_MECA) are given in [R5.03.02]

```
      IF (OPTION (1:9) .EQ. 'RAPH_MECA' .OR.
     &     OPTION (1:9) .EQ. 'FULL_MECA') THEN
        IF (SEUIL.LT.0.D0) THEN
          VIP (7) = 0.D0
          DP = 0.D0
          SIELEQ = 1.D0
          A1 = 0.D0
          A2 = 0.D0
        ELSE
          VIP (7) = 1.D0
          DP = THRESHOLD (1.5D0* (DEUXMU+C))
          A1 = (DEUXMU/(DEUXMU+C)) * (SEUIL/SIELEQ)
          A2 = (C     /(DEUXMU+C)) * (SEUIL/SIELEQ)
        ENDIF
        PLASTI=VIP (7)
        C 160 K = 1, NDIMSI
          SIGDV (K) = SIGEL (K) - A1* (SIGEL (K) - VIM (K) *C/CM)
          SIGP (K)  = SIGDV (K) + (SIGMO + TROISK*EPSMO) *KRON (K)
          VIP (K)   = VIM (K) *C/CM   + A2* (SIGEL (K) - VIM (K) *C/CM)
160     CONTINUOUS
      ENDIF
```

- **CALCULATION OF TANGENT OPERATOR 'DSIPSEP': RIGI_MECA_TANG (of speed) or FULL_MECA (coherent)**

```
      IF (OPTION (1:14) .EQ. 'RIGI_MECA_TANG' .OR.
     &     OPTION (1:9)    .EQ. 'FULL_MECA') THEN
        CAL MATINI (6,6,0.D0, DSIDEP)
        C 120 K=1,6
          DSIDEP (K, K) = DEUXMU
120     CONTINUOUS
```

```
        IF (OPTION (1:14) .EQ. 'RIGI_MECA_TANG') THEN
          C 174 K = 1, NDIMSI
            SIGDV (K) = SIGDV (K) - VIM (K) *C/CM
174       CONTINUOUS
        ELSE
          C 175 K = 1, NDIMSI
            SIGDV (K) = SIGDV (K) - VIP (K)
175       CONTINUOUS
        ENDIF
        SIGEPS = 0.D0
        C 170 K = 1, NDIMSI
          SIGEPS = SIGEPS + SIGDV (K) *DEPSDV (K)
170     CONTINUOUS
        A1 = 1.D0/(1.D0+1.5D0* (DEUXMU+C)*DP/SIGY)
        A2 = (1.D0+1.5D0*C*DP/SIGY) *A1
        IF (PLASTI.GE.0.5D0.AND.SIGEPS.GE.0.D0) THEN
          COEFF = -1.5D0* (DEUXMU/SIGY) ** 2/(DEUXMU+C) * A1
          C 135 K=1, NDIMSI
            C 135 L=1, NDIMSI
              DSIDEP (K, L) = A2 * DSIDEP (K, L) + COEF*SIGDV (K) *SIGDV (L)
135       CONTINUOUS
          LAMBDA = AVERAGE + DEUXMU ** 2*A1*DP/SIGY/2.D0
        ENDIF
        C 130 K=1,3
          C 131 L=1,3
            DSIDEP (K, L) = DSIDEP (K, L) + AVERAGE
131       CONTINUOUS
130     CONTINUOUS
      ENDIF
```

5.4.3 Variables of order (external state variable)

The dependence of the parameters material to the variables of order (drying, temperature, etc) is taken into account by the use of the routine RCVALB . There exists in addition to the variables of orders specific to certain laws of behavior which must be ***calculated*** (in general on the level of the element) and not transmitted by the mechanism AFFE_MATERIAU/AFFE_VARC , it is about:

- ELTSIZE1 : calculation of the size of an element (for the law BETON_DOUBLE_DP);
- ELTSIZE2 : calculation of the size of an element (for the law ENDO_PORO_BETON);
- COORGA : coordinates of the point of Gauss of the element (for the law META_LEMA_ANI);
- GRADVELO : for the gradient the speed of deformation (for the law MONOCRYSTAL);
- HYGR : hygroscopy (starting from the function of desorption FONC_DESORP).

The laws of behavior using these variables use the keyword exte_vari in their catalogue. The addition of another variable of this style requires a specific development which cannot thus be here detailed.
On the other hand, one can to use variables defined higher. It is necessary:

- to impact the catalogue of the law of behavior and to add this variable in exte_vari ;
- to recover the value of this variable via the module calculation

Variables	Variable(S) in the module calculation
HYGR	ca_vext_hygrm_ ca_vext_hygrp_
ELTSIZE1	ca_vext_eltsize1_
ELTSIZE2	ca_vext_eltsize2_ (9)
COORGA	ca_vext_coorga_ (27.3)
GRADVELO	ca_vext_gradvelo_ (9)

For example:

```
use calcul_module, only: ca_vext_gradvelo_

real (kind=8):: L (3.3)
integer:: I, J

    C I = 1.3
        C J = 1.3
            L (I, J) =ca_vext_gradvelo_ (3* (i-1) +j)
        end C
    end C
```

6 Typical case of the MFront laws

In complement of the document [u2.10.02] which describes how to use a law of MFront behavior in prototype mode, one specifies in this paragraph the characteristics of the laws of behavior MFront which are officially (under quality assurance) integrated into code_aster.

The name of the MFront file must be identical in the name of the behavior described in the file.

When the official laws of MFront behavior are compiled, the procedure of construction must know which files are produced by the conversion of the MFront file into C++. Usually, two files are produced: one of the name of the named behavior and one `aster` + name of the behavior.
If another file is produced, it should be indicated in the MFront file for example (in PlasticityTH.mfront):
`//output Acier_ElasticYield-mfront`

6.1 Catalogue for DEFI_MATERIAU and RELATION

The catalogue of order of the parameters materials is automatically integrated in order DEFI_MATERIAU by the procedure of description (at the time DE the stage `waf installation`):
- the keyword factor is the name of the behavior (example: `AnisoLemaitre`),
- the name of the parameters materials corresponds to `@MaterialProperty`.

If the parameters are tables (example: `@MaterialProperty real has [3]`), they will be named `a_0`, `a_1` and `a_2`.
A second keyword factor suffix by `_FO` (example: `AnisoLemaitre_FO`) is created in which all the parameters are functions.

The name of the behavior is automatically added for the keyword RELATION.

6.2 Catalogue behavior

The description of the behavior is slightly different. C is neededréer an object of the type `LoiComportementMFront` and to indicate the name of the produced symbol during compilation.

7 Case of the laws of metallurgical behavior (CALC_META)

One considers here the laws of behavior allowing to calculate the metallurgical state in the operator `CALC_META` and not laws of mechanical behaviorS using these phases.

A metallurgical law of behavior is made upE of two parts:
- Lbe metallurgical phases on which ellE acts (`STEEL` or `ZIRC`);
- Llaw of metallurgical behavior has itself.

The addition of a new type of phase is very impacting:
- Impact in CALC_META and its peripherals;
- Impact in AFFE_MATERIAU (for the variables of order AFFE_VARC);
- Impact in the laws of mechanical behavior using the metallurgy.

7.1 Modification of the catalogue of order

One can add a new model of metallurgical law of behavior by binding it to the phases (STEEL or ZIRC) in the keyword factor BEHAVIOR of CALC_META. Should be impacted the simple keyword LOI_META.

7.2 Modification of the catalogue of behavior

A metallurgical law of behavior has its catalogue (in bibpyt/Behavior) who contains only little information (compared to the mechanical laws of behavior). PhasR example for the law WAECKEL :

```
law = LoiComportement (
    name            = 'WAECKEL',
    lc_type        = ('MODELE_METALLURGIQUE',),
    Doc.            = """ standard metallurgical Model for steel """,
    num_lc         = 2,
    nb_vari        = 3,
    nom_vari       = ('TAILLE_GRAIN', 'TEMP', 'TEMP_MARTENSITE',
        ),
    mc_mater       = ('META_ACIER',),
    modeling   = ('3D', 'AXIS', 'D_PLAN',),
)
```

One cannot modify modeling (because the metallurgy always acts on these three modelings). The rest changes as in the §3.3 .

A metallurgical law of behavior is made up of a model (LOI_META) and of a phase (RELATION).
The number and the name of the internal variables are bound by a system of KIT. Lphase has will give the list of the phases standards (steel or Zircaloy), the type will possibly make it possible to add other "internal variables typically" (aujourd'today, for the model of Waeckel, one has the temperature, time of transformation, the size of the grains, etc).

To determine Lroutine called for calculation has, it is it even principle that for laws of behavior mechanics, it will be the addition of the nature of the phases (steelZircaloy) and type of model used.

The mecanism is of type nzcomp.F90+lzxxxx.F90 (like nmcomp.F90+lcxxxx.F90).
- Steel: num_lc = 20000
- Zircaloy: num_lc = 30000

The model of Wackel has num_lc = 2, therefore the metallurgical law of behavior of steel with the model of Waeckel will call the routine lz20002. F90

8 Validation and maintenance

The reflection on the tests of validation and identification can be started very early, before even the development itself. One can distinguish:
- command files (often simple, using for example SIMU_POINT_MAT) allowing to identify certain parameters (using MACR_RECAL).
- Lbe tests to be restored: it must make it possible to validate the behavior in all its aspects, for a range of value of parameters covering the field of validity correctly. Certain tests are almost obligatory:

- tests of robustness and invariance compared to a rotation, a change of unit (COMP001, COMP002, COMP003)
- tests checking the good taking into account of the variables of orders (COMP008, COMP010)

Moreover the developer can have to correct or analyze its behavior following an anomaly met at the time of a study. In this case, besides the usual techniques of debugging, it is possible to use a functionality put in work in the event of failure of integration of the behavior: first points in failure produce a small command file making it possible to play again the scene with SIMU_POINT_MAT, for better analyzing the problem, testing other methods. It is enough to recopy the definition of material; all other data (forced, deformations, initial internal variables, and increment of deformation make it possible to simulate the behavior at the moment and the point where the failure occurred. For example for the test forma03c:

```
#-----------------------------------------------------------------------
# test to analyze the failure of integration on the mesh <M83>, not <3>
#-----------------------------------------------------------------------
BEGINNING ()

# to recopy DEFI_MATERIAU (...)
# DIAGRAM TRACTION
CTRAC = LIRE_FONCTION (UNITE=21, NOM_PARA=' EPSI', PROL_DROITE=' CONSTANT',)

MAT=DEFI_MATERIAU (ELAS=_F (E=200000., NU=0.3), TRACTION=_F (SIGM=CTRAC))

LISTE=DEFI_LISTE_REEL (DEBUT=1.944000000000000E+02,
                  INTERVALLE=_F (JUSQU_A= 2.187000000000000E+02, NOMBRE=1))

DEFLIST = DEFI_LIST_INST ( DEFI_LIST=_F (LIST_INST=LISTE,),
                           ECHEC=_F (SUBD_NIVEAU=10, SUBD_PAS=4),)

EXX=DEFI_FONCTION (NOM_PARA=' INST',
                   VALE= (1.944000000000000E+02, -9.407813329102166E-04,
                          2.187000000000000E+02,7.373776084156800E+06))

EYY=DEFI_FONCTION (NOM_PARA=' INST',
                   VALE= (1.944000000000000E+02, 1.718362911427018E-04,
                          2.187000000000000E+02, -5.221483651275956E+06))

EZZ=DEFI_FONCTION (NOM_PARA=' INST',
                   VALE= (1.944000000000000E+02, 0.000000000000000E+00,
                          2.187000000000000E+02, - 9.224110429927666E+05))

EXY=DEFI_FONCTION (NOM_PARA=' INST',
                   VALE= (1.944000000000000E+02, 3.621278829822514E-04,
                          2.187000000000000E+02, -4.710098874951571E+06))

RESU=SIMU_POINT_MAT (INFO=1, MATER=MAT, INCREMENT=_F (LIST_INST=DEFLIST),
                     EPSI_IMPOSE=_F (EPXX=EXX, EPYY=EYY, EPZZ=EZZ, EPXY=EXY),
                     SUPPORT=' ELEMENT', MODELISATION=' C_PLAN',
                     EPSI_INIT=_F (   EPXX=-9.407813329102166E-04,
                                   EPYY=1.718362911427018E-04,
                                   EPZZ= 0.000000000000000E+00,
                                   EPXY=3.621278829822514E-04,
                                   EPXZ=0.000000000000000E+00,
                                   EPYZ= 0.000000000000000E+00),
                     SIGM_INIT=_F (   SIXX=-1.887253339740370E+02,
                                   SIYY=-2.497632316081525E+01,
                                   SIZZ= 0.000000000000000E+00,
                                   SIXY=7.537194347798921E+01,
                                   SIXZ=0.000000000000000E+00,
                                   SIYZ= 0.000000000000000E+00),
                     VARI_INIT=_F (VALE= (3.931095545774485E-05,
                                          1.000000000000000E+00,)),
```

```
                        COMPORTEMENT=_F          (RELATION='          VMIS_ISOT_TRAC',
                           ITER_INTE_MAXI=20,),
                        NEWTON=_F (REAC_ITER=1),)

    END ()
```