

Implicit integration and clarifies nonlinear behavior models

Summarized

This document describes two integration methods (implicit and explicit) for the resolution of problems with nonlinear behaviors in operators `STAT_NON_LINE` [R5.03.01] and `DYNA_NON_LINE` [R5.05.05].

The implicit numerical method presented is that of Newton, with or without linear search.

The explicit numerical method is that of Runge Kutta of order 2 with adaptive step.

Contents

1	General information on the integration of the models of comportement3.....	
1.1	Algorithm of resolution of the problem.....	quasi- statique3
1.2	Goal of the intégration4.....	
1.3	Methods of résolution4.....	
2	Integration clarify of a behavior model incrémentale4.....	
2.1	Use of a method explicite5.....	
2.2	Control of the steps of temps5.....	
2.3	Influence on the implicit stage.....	
globale6	3 Integration of an incremental behavior model.	7.3.1
	implicit Integration by the method of Newton7.....	
3.2	Operator tangent8.....	
3.3	implicit Integration by the method of Newton with search linéaire9.....	
	3.3.1 Presentation of the general algorithm of search linéaire9.....	
	3.3.2 Implementation of the linear search in PLASTI11.....	
4	model Installation of new of behavior	12.4.1
	Methods possibles12.....	
4.2	model Installation of new of behavior by the method explicite12.....	
4.3	complete Establishment of a new behavior in PLASTI12.....	
4.4	implicit Establishment easy of a new model of behavior in environment PLASTI13.....	
5	Bibliographie14.....	
6	Description of the versions of the document14.....	

1 General information on the integration of the constitutive laws

1.1 Algorithm of resolution of the quasi-static problem

When the behavior of the materials composing structure is not linear, the resolution of the total equilibrium of structure at every moment is a nonlinear problem whose variational formulation in the case as of small strains can be written, in the quasi-static case, in the following form, deduced from the statement of the principle of the virtual works (cf [R5.03.01]):

$$\begin{cases} \mathbf{v}^T \cdot \mathbf{L}^{\text{int}}(\mathbf{u}, t) = \mathbf{v}^T \cdot \mathbf{L}^{\text{ext}}(t) & \forall \mathbf{v} \text{ tel que } \mathbf{B} \cdot \mathbf{v} = 0 \\ \mathbf{B} \cdot \mathbf{u} = \mathbf{u}^d(t) \end{cases} \quad (1)$$

Where:

- t represent the variable of time
- \mathbf{u} is the field of displacement taken from a reference configuration
- \mathbf{v} is the kinematically admissible field of virtual displacement
- the relation $\mathbf{B} \cdot \mathbf{u} = \mathbf{u}^d(t)$ corresponds to the boundary conditions imposed in displacements
- $\mathbf{L}^{\text{ext}}(t)$ is the external mechanical loading to which the structure is subjected
- \mathbf{L}^{int} represents the internal forces of the problem of quasi-static mechanics nonlinear. $\mathbf{L}^{\text{int}}(\mathbf{u}, t)$ is connected to the stress field $\boldsymbol{\sigma}$ by the operator work of the virtual strains \mathbf{Q}^T according to the following relation:

$$\mathbf{L}^{\text{int}}(\mathbf{u}, t) = \mathbf{Q}^T(\mathbf{u}) \cdot \boldsymbol{\sigma} \quad (2)$$

the dualisation of the boundary conditions of Dirichlet $\mathbf{B} \cdot \mathbf{u} = \mathbf{u}^d(t)$ leads to the following problem [R3.03.01]:

$$\begin{cases} \mathbf{L}^{\text{int}}(\mathbf{u}, t) + \mathbf{B}^T \cdot \boldsymbol{\lambda} = \mathbf{L}^{\text{ext}}(t) \\ \mathbf{B} \cdot \mathbf{u} = \mathbf{u}^d(t) \end{cases} \quad (3)$$

the formulation of the discretized quasi-static problem consists in expressing the equilibrium of structure for a continuation of times of computation $\{t_i\}_{1 \leq i \leq I}$ which parameterize the loading, one will note the quantities at time t_i by the index i (for example $\mathbf{L}^{\text{int}}(\mathbf{u}_i, t_i) = \mathbf{L}_i^{\text{int}}$):

$$\begin{cases} \mathbf{L}_i^{\text{int}} + \mathbf{B}^T \cdot \boldsymbol{\lambda}_i = \mathbf{L}_i^{\text{ext}} \\ \mathbf{B} \cdot \mathbf{u}_i = \mathbf{u}_i^d \end{cases} \quad (4)$$

This nonlinear problem is solved by a method of Newton (or quasi-Newton), cf [R5.03.01]. The system to be solved is written then, for each iteration n of the phase of correction:

$$\begin{cases} \mathbf{K}_i^n \cdot \delta \mathbf{u}_i^{n+1} + \mathbf{B}^T \cdot \delta \boldsymbol{\lambda}_i^{n+1} = \mathbf{L}_i^{\text{méca}} - \mathbf{L}_i^{\text{int}, n} - \mathbf{B}^T \cdot \boldsymbol{\lambda}_i^n \\ \mathbf{B} \cdot \delta \mathbf{u}_i^{n+1} = 0 \end{cases} \quad (5)$$

the vector of the internal forces $\mathbf{L}_i^{\text{int}, n}$ is calculated starting from the stresses $\boldsymbol{\sigma}_i^n$, those being calculated starting from displacements \mathbf{u}_i^n via the behavior model of the material. In fact, in the case

as of incremental behaviors , σ_i^n is calculated from $(\sigma_{i-1}, \alpha_{i-1})$ and of the increment of strain $\epsilon(\Delta \mathbf{u}_i^n)$ induced by the displacement increment since the beginning of the iterative process.

1.2 Goal of integration

This diagram requires, from the estimate of displacements to the iteration n , to calculate in each Gauss point:

the tensor of the stresses σ_i^n which checks the constitutive law, and which makes it possible and to $\mathbf{L}_i^{\text{int},n}$ calculate local variables specific to each behavior,

the operator of coherent tangent behavior $\left(\frac{\partial \sigma}{\partial \epsilon}\right)_i^n$, to obtain the tangent matrix:

$$\mathbf{K}_i^n = \sum_e \mathbf{K}_{e,i}^n \text{ with } \mathbf{K}_{e,i}^n = \int_{\Omega_e} \mathbf{Q}^T \left(\frac{\partial \sigma}{\partial \epsilon}\right)_i^n \mathbf{Q} d\Omega_e$$

1.3 Methods of resolution

For each finite element, in each point of integration, it is thus necessary to integrate the behavior, i.e. to calculate the tensor of the stresses and the local variables solutions of the problem as well as the tangent operator.

We present in the paragraphs according to two general algorithms allowing the computation of the solution: either the resolution by a method of Newton, based on an implicit discretization of the equations, or by a method of Runge-Kutta of order 2 with control of error, based on an explicit discretization of these equations.

Other methods of resolution, specific to each behavior, are also employed. It are founded on an optimization of the system of equations nonlinear, making it possible to reduce the number of equations to be solved, very often with only one equation. The method used then is a simple search for zero of function (see for example [R5.03.02], [R5.03.04], [R5.03.16]).

For all these algorithms, it is possible, in the event of nonconvergence, to carry out a local recutting of time step. ; for that, one adopts a linear interpolation of displacement and command variables (for example the temperature) during the time interval, which leads, for any increment of time with:

$$\forall \tau \in [t_{i-1}, t_i] \quad T(\tau) = T_{i-1} + \frac{\tau - t_{i-1}}{\Delta t} [T - T_{i-1}] \quad \text{and} \quad \epsilon_{ij}^k(\tau) = \epsilon_{ij}^k(\mathbf{u}_{i-1}) + \frac{\tau - t_{i-1}}{\Delta t} [\epsilon_{ij}^k(\mathbf{u}_i^n) - \epsilon_{ij}^k(\mathbf{u}_{i-1})]$$

2 explicit Integration of an incremental behavior model

to solve the nonlinear system relating to the constitutive law, one can use an explicit integration method of type Runge Kutta of order 2 (option "RUNGE_KUTTA" of operand ALGO_INTE of factor key word the COMP_INCR) for the incrémentaux problems of nonlinear behavior.

The relations concerned currently are:

VISCOCHAB	Behavior élasto-viscoplastique de Chaboche, in two centers, with effect of memory and restoration
VENDOCHAB	Behavior viscoplastique with damage of Chaboche
POLYCRISTAL	Behavior élasto-viscoplastique polycristalline homogénéisé
MONOCRISTAL	single-crystal viscoplastique Behaviors of metals
HAYHURST	Behavior viscoplastique with damage of Hayhurst

relation POLYCRISTAL is accessible only by the explicit method presented hereafter.

This kind of integration makes it possible to very easily establish a new model of behavior [bib2]. One describes the computation of the stress field from an increment of strain given while following the evolution of the local variables.

2.1 Use of a method clarifies

One supposes that the system to be solved can be expressed in the form:

$$\frac{dY}{dt} = F(Y, t; \Delta \varepsilon_i^n, \sigma_{i-1})$$

where Y all the local variables of the model represent, including the variation of plastic strain $\Delta \varepsilon^p$. Their derivatives in time are supposed to depend only on the local variables and the stresses at previous time, and the increase in total deflection provided by the total algorithm of Newton $\Delta \varepsilon_i^n$.

This assumption rests in particular on the linear relation (Hooke's law) between the tensor of the stress at current time and the elastic part of the tensor of the strains: $\Delta \sigma = \mathbf{A}(\Delta \varepsilon_i^n - \Delta \varepsilon^p)$, which makes it possible to express σ according to $\Delta \varepsilon^p$ in the differential equations

$$\frac{dY}{dt} = F(Y, t; \sigma(\Delta \varepsilon_i^n, \sigma_{i-1}, Y))$$

One of the techniques simplest to implement to solve these differential equations is the use of explicit methods. So that they numerically remain effective, it is essential to associate an automatic control of step to them to preserve a good compromise cost accuracy.

Methods of explicit and encased Runge and Kutta [bib1], [bib2] are undoubtedly the simplest diagrams of integration respecting these criteria. Their principle is to associate two diagrams of integration of a different nature to control time step according to a necessary accuracy. According to the order of integration chosen, several algorithms are available and the simplest diagram is a method of order two.

One integrates according to the following diagram:

$$Y_{t+h} = Y^{(2)} \text{ if the criterion of accuracy is satisfied;}$$
$$Y^{(2)} = Y + \frac{h}{2} [F(Y, t) + F(Y^{(1)}, t+h)] \text{ with } Y^{(1)} = Y + hF(Y, t)$$

the difference between $Y^{(2)}$ (diagram in order 2) and $Y^{(1)}$ (diagram of order 1, Eulerian) provides an estimate of the error of integration and allows to control the size of time step h which is initialized with Δt_i for the first attempt. Thus, the method remains effective if the behavior remains elastic during the increment and one naturally has under local steps independent of the total increment to integrate, with a better accuracy, the evolution of the local variables to Gauss points where it not linearity of the behavior is most significant.

2.2 Control time step

strategy of the control of the step is defined from of the difference between the two integration methods: $\|Y^{(2)} - Y^{(1)}\|$ and of the accuracy required by the user η (key word: RESI_INTE_REL). The criterion selected is the following, where one notes $Y = (y_1, y_2, \dots, y_N)$:

$$\delta Y(t) = \sup_{j=1 \dots N} \left\{ \frac{|y_j^{(2)} - y_j^{(1)}|}{\max[\varepsilon, |y_j(t)|]} \right\} < \eta$$

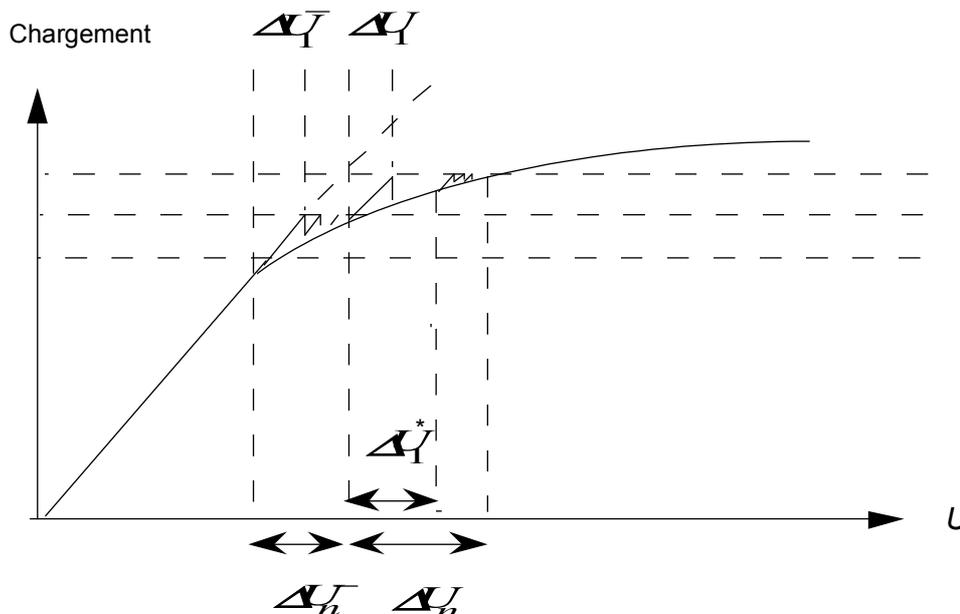
The parameter ε is fixed at 0,001. The accuracy of desired integration η must be coherent with the level of accuracy necessary for the total stage. The value by default $\eta = 10^{-6}$ often proves penalizing in terms of time computation. It can be often to be increased by an order or two, in particular for the models élasto-visco-plastics, without real loss of accuracy (to be seen on a case-by-case basis).

If the criterion is not checked, time step Re-is cut out according to a discovery method. When time step becomes too weak ($h < \text{accuracy machine}$), a return code is emitted, and the total recutting of time step is activated, if required by the user.

2.3 Influence on the total stage

In its current version, the method does not provide a tangent matrix; its evaluating is however possible (but expensive) by a technique of disturbances (activated under `COMP_INCR` key word `TYPE_MATR_TANG='PERTURBATION'`). If one uses the elastic matrix for the total equilibrium, the nombre of iterations of Newton to converge can become very important, which often brings to more significant number of increments so that the total resolution is facilitated.

To improve convergence, it is possible to initialize the iterations of Newton from the solution previously calculated: instead of considering the displacement increment for the first iteration at time running (obtained via the elastic matrix), one initializes the displacement increment searched for the first iteration ΔU_1 from the value ΔU_n , solution with the preceding increment (cf appears following page). Key word `PREDICTION='EXTRAPOLE'` (under `NEWTON`) [U4.51.03] allows to initialize with the converged increment of the preceding step (balanced by the ratio of time step). This estimate is projected on the field of kinematically admissible displacements so that the final solution checks the boundary conditions of Dirichlet well.



3 Implicit integration of an incremental behavior model.

To solve the nonlinear systems of equations relating to the incrementaux problems of nonlinear behavior (operand `RELATION` of the key word factor `COMP_INCR`). it is possible to use an implicit integration method of Newton-Raphson type, with which one can associate a linear stage of search (resp. option "NEWTON" and option "NEWTON_RELI" of operand `ALGO_INTE` of factor key word the `CONVERGENCE`).

For more details to consult the document [U4.51.03] user's manual.

The relations concerned are for example:

VISCOCHAB	Behavior élasto-viscoplastique of J.L. Chaboche, in two centers, with effect of memory and restoration
MONOCRISTAL	single-crystal viscoplastic Behaviors of metals
IRRAD3M	Behavior elastoplastic under irradiation of the stainless steels
...	

relations `VISCOCHAB` and `MONOCRISTAL` can also be integrated by the method clarified exposed with the §22.

3.1 Implicit integration by the method of Newton

environment `PLASTI` makes it possible in `Code_Aster` to integrate by a method of Newton, in a systematic way, models of nonlinear behavior. The approach to solve the equations of the model can be schematized in the following way: knowing all the variables (forced, local variables) at previous time t_{i-1} , and the increase in total deflection provided to the point considered by the total algorithm of Newton $\epsilon(\Delta \mathbf{u}_i^n)$, also noted $\Delta \epsilon_i^n$, it is necessary to find the solution of the system:

$$f(\Delta Y) = 0 = \begin{bmatrix} g(\Delta y) \\ l(\Delta y) \\ i(\Delta y) \\ j(\Delta y) \\ k(\Delta y) \end{bmatrix} \quad \text{with} \quad \Delta y = \begin{pmatrix} \Delta \sigma \\ \Delta \epsilon^p \\ \Delta X_1 \\ \Delta X_2 \\ \Delta p \end{pmatrix} = y_i - y_{i-1}.$$

The first equation represents for example the elastic relation stress-strain $g(\Delta y) = \Delta \sigma - \mathbf{A}(\Delta \epsilon_i^n - \Delta \epsilon^p)$, the second expressing yielding, and the following correspond to the laws of evolution of the various local variables, vectorial or scalar, noted here ΔX_1 , ΔX_2 and Δp .

The number of equations to be solved varies obviously according to the behavior (see for example behaviors `MONOCRISTAL` [R5.03.11], `VISCOCHAB` [R5.03.12], `IRRAD3M` [R5.03.23]).

One solves this system by the method of Newton proposed in environment `PLASTI`, that is to say:

$$\begin{cases} \frac{\partial f^l}{\partial \Delta y_k} d(\Delta y_k) = -f^l(\Delta y_k) \\ \Delta y_{k+1} = \Delta y_k + d(\Delta y_k) \end{cases} \quad \text{while reiterating in } k \text{ until convergence.}$$

It remains to calculate the jacobian matrix of the system: $\frac{df}{dy_k}$. It has the following form:

$$J = \begin{bmatrix} \frac{\partial g}{\partial \Delta \sigma} & \frac{\partial g}{\partial \Delta \varepsilon^p} & \frac{\partial g}{\partial \Delta X_1} & \frac{\partial g}{\partial \Delta X_2} & \frac{\partial g}{\partial \Delta p} \\ \frac{\partial l}{\partial \Delta \sigma} & \frac{\partial l}{\partial \Delta \varepsilon^p} & \frac{\partial l}{\partial \Delta X_1} & \frac{\partial l}{\partial \Delta X_2} & \frac{\partial l}{\partial \Delta p} \\ \frac{\partial i}{\partial \Delta \sigma} & \frac{\partial i}{\partial \Delta \varepsilon^p} & \frac{\partial i}{\partial \Delta X_1} & \frac{\partial i}{\partial \Delta X_2} & \frac{\partial i}{\partial \Delta p} \\ \frac{\partial j}{\partial \Delta \sigma} & \frac{\partial j}{\partial \Delta \varepsilon^p} & \frac{\partial j}{\partial \Delta X_1} & \frac{\partial j}{\partial \Delta X_2} & \frac{\partial j}{\partial \Delta p} \\ \frac{\partial k}{\partial \Delta \sigma} & \frac{\partial k}{\partial \Delta \varepsilon^p} & \frac{\partial k}{\partial \Delta X_1} & \frac{\partial k}{\partial \Delta X_2} & \frac{\partial k}{\partial \Delta p} \end{bmatrix}$$

The stopping criteria of the iterations relate, according to the behaviors, to the nullity of the residue:

$$\frac{f(y_k)}{f(y_0)} < \varepsilon \text{ or the stationarity of the solution: } |y_{k+1} - y_k| < \varepsilon.$$

The initial values y_0 are selected null by default, but can take particular values according to the behaviors.

More precisely, the tests of stop and the initial values are:

Behaviors	Stopping criteria	Initialization
VISCOCHAB	$ y_{k+1} - y_k < \varepsilon y_{k+1} $	Solution clarifies and modification of the initialization in the event of bad convergence
MONOCRISTAL	$ f(y_k) < \varepsilon f(y_0) $	elastic Solution
null	$ f(y_k) < \varepsilon f(y_0) $	IRRAD3M Solution

the method used time step allows a local recutting of, either systematic, or in the event of nonconvergence.

3.2 Tangent operator

the tangent operator can be obtained directly starting from the preceding system. Indeed, the formed system by the equations of the model with convergence, $(f(\Delta y_{cv}) = 0)$ is checked at the end of the increment. For a small variation of f , by regarding this time the increase in total deflection $\Delta \varepsilon_i^n$ variable and not as parameter, the system remains with the equilibrium and one checks $df = 0$, i.e.:

$$\frac{\partial f}{\partial \Delta \sigma} \delta \Delta \sigma + \frac{\partial f}{\partial \Delta \varepsilon_i^n} \delta \Delta \varepsilon_i^n + \frac{\partial f}{\partial \Delta \varepsilon^p} \delta \Delta \varepsilon^p + \frac{\partial f}{\partial \Delta X_1} \delta \Delta X_1 + \frac{\partial f}{\partial \Delta X_2} \delta \Delta X_2 + \frac{\partial f}{\partial \Delta p} \delta \Delta p = 0$$

This system can be still written:

$$\frac{\partial f}{\partial y} \delta(y) = X, \text{ avec } Y = \begin{bmatrix} \Delta \sigma \\ \Delta Z_s \end{bmatrix} \text{ et } X = \begin{bmatrix} \delta \Delta \varepsilon_i^n \\ 0 \end{bmatrix}$$

where ΔZ represents the unknowns complementary to $\Delta \sigma$ in y .

By writing the jacobian matrix in the form:

$$J \cdot \delta Y = \begin{bmatrix} Y_0 & Y_1 \\ Y_2 & Y_3 \end{bmatrix} \begin{bmatrix} \Delta \sigma \\ \Delta Z \end{bmatrix}$$

then while operating by successive eliminations and substitutions, the required tangent operator can be written directly:

$$\left(\frac{\partial \Delta \sigma}{\partial \Delta \varepsilon} \right)_i^n = (Y_0 - Y_1 Y_3^{-1} Y_2)^{-1}$$

The preceding equations show that one is led to re-use the same jacobian matrix J as previously to evaluate the tangent operator.

3.3 Implicit integration by the method of Newton with linear search

In order to improve the robustness of the method of Newton, one gives the opportunity of activating the linear search.

3.3.1 Presentation of the general algorithm of linear search

Is a system of equations nonlinear to solve: $f(x) = 0$

One associates with f a functional calculus F .

The method of Newton with linear search consists in finding with each iteration a step of advance following the direction of descent which minimizes (with the strong meaning or the weak meaning) the functional calculus. That is brought back in a sequence of the following stages:

computation of the direction of descent: $d_k = -[\nabla f(x_k)]^{-1} \cdot f(x_k)$

computation of the step of advance: ρ such as $F(x_k + \rho \cdot d_k) < F(x_k)$, the best being ρ which minimizes the functional $\rho = \operatorname{argmin}[F(x_k + \rho \cdot d_k)]$

actualization of the unknowns: $x_{k+1} = x_k + \rho d_k$

One associates with f the functional calculus $F(\rho) = \frac{1}{2} \|f(x_k + \rho \cdot d_k)\|^2$.

Contrary in the linear search of total Newton where one determines ρ by the resolution of $F'(\rho) = 0$, one seeks here one $\rho \in [0, 1]$, $\rho \neq 0$ which decreases sufficiently the functional calculus F . One will higher than not seek a step of advance not Newton ($\rho = 1$). The principle is described in [1].

Confidence interval:

To know if one ρ decreases sufficiently the functional calculus F , one uses the rule of Armijo. This one makes it possible to characterize a confidence interval by saying that ρ is appropriate if $F(\rho) \leq F(0) + \omega F'(0)\rho$ where ω is a built-in reality lower than 1. This statement can be represented graphically by Appear 3.3.1-a : the curve in dotted lines blue is the graph of the functional calculus $h(\rho) = (7\rho - 4)^4 + (7\rho - 5)^3$, that is purple is the tangent in the beginning, of slope $F'(0)$ and that in dotted lines red is the line of slope $\omega F'(0)$ with $\omega = 0,1$. In this example, ρ sufficiently the functional calculus decreases if $\rho \in [0; 0,871]$.

Principle of the computation of ρ :

One starts initially by testing the step of Newton ($\rho = 1$). If this step satisfied the rule with Armijo, it is retained and the iterations of Newton are continued. If not, one makes a quadratic interpolation knowing $F(0)$, $F'(0)$ and $F(1)$. New then is determined $\rho = \rho_q$. If this step satisfied the rule with Armijo, it is retained and the iterations of Newton are continued. If not, one carries out a cubic interpolation knowing $F(0)$, $F'(0)$, $F(1)$ and $F(\rho_q)$. New then is determined $\rho = \rho_c$. If this step satisfied the rule with Armijo, it is retained and the iterations of Newton are continued. If not, one continues the cubic interpolations between the 0 and the 2 last ρ . The paragraph below details the computation of ρ .

Details of the computation of ρ :

One starts initially by testing the step of Newton ($\rho = 1$). At this stage, one knows $F(0) = \frac{1}{2} \|f(x_k)\|^2$ and $F'(0) = -\|f(x_k)\|^2$.

One calculates $F(1) = \frac{1}{2} \|f(x_k + d_k)\|^2$.

If the rule of Armijo is not satisfied, i.e. if $F(1) > F(0) + \omega F'(0)$, then one models $F(\rho)$ by a polynomial of order 2 (quadratic interpolation):

$$q(\rho) = [F(1) - F(0) - F'(0)]\rho^2 + F'(0)\rho + F(0)$$

The minimum of this polynomial is obtained for $\rho_q = \frac{-F'(0)}{2[F(1) - F(0) - F'(0)]}$. Let us note that if $F(1)$ is much larger than $F(0)$, then the minimum of the polynomial is reached into 0. However one does not wish to have a step of too small advance, under penalty of having to do one very a large number of iterations of Newton. For that, one is imposed $\rho_{min} > 0$ (for example $\rho_{min} = 0,1$). One can also impose a higher limit ρ_{max} (for example $\rho_{max} = 0,6$). Thus, so $\rho_q < \rho_{min}$ then $\rho_q = \rho_{min}$ and so $\rho_q > \rho_{max}$ then $\rho_q = \rho_{max}$.

Then one calculates $F(\rho_q) = \frac{1}{2} \|f(x_k + \rho_q \cdot d_k)\|^2$.

If the rule of Armijo is not satisfied, i.e. if $F(\rho_q) > F(0) + \omega F'(0)\rho_q$, then one models $F(\rho)$ by a polynomial of order 3 (cubic interpolation):

$$c(\rho) = a\rho^3 + b\rho^2 + F'(0)\rho + F(0)$$

with

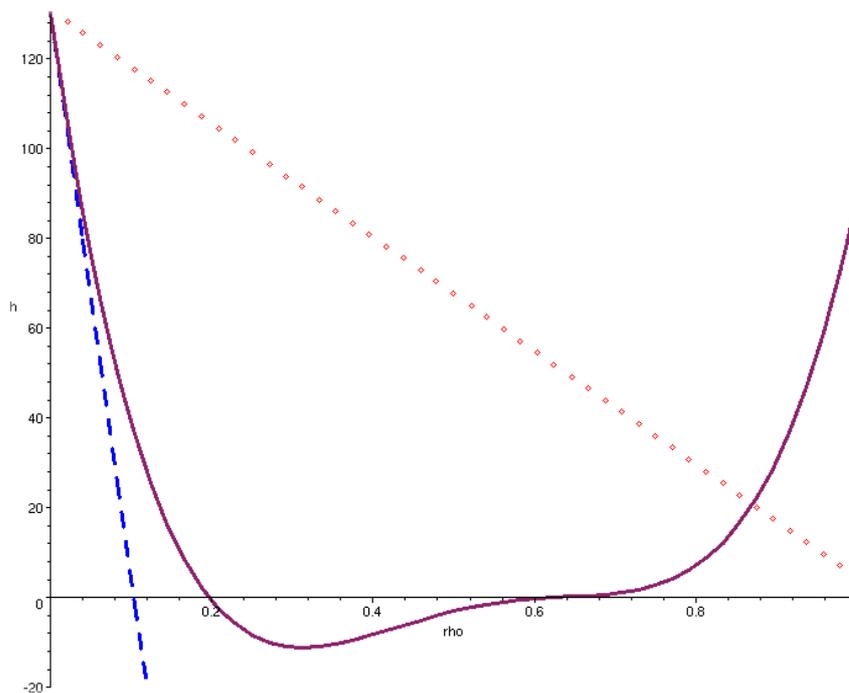
$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\rho_1 - \rho_2} \begin{bmatrix} \frac{1}{\rho_1^2} & \frac{-1}{\rho_2^2} \\ \frac{-\rho_2}{\rho_1^2} & \frac{\rho_1}{\rho_2^2} \end{bmatrix} \begin{bmatrix} F(\rho_1) - F(0) - F'(0)\rho_1 \\ F(\rho_2) - F(0) - F'(0)\rho_2 \end{bmatrix}$$

where ρ_1 and ρ_2 are the two last values failed of ρ .

The minimal point of $c(\rho)$ is:

$$\rho_c = \frac{-b + \sqrt{b^2 - 3a F'(0)}}{3a}$$

The cubic interpolations are continued as long as the rule of Armijo is not satisfied. In practice, one limits to 2 the number of cubic interpolations.



Appear 3.3.1-a : illustration of the rule of Armijo

3.3.2 Put in work of the linear search in PLASTI

the system to solve is:

$$f(\Delta y) = 0$$

The following iterations of Newton are made:

update of the variables at time “+”

computation of the residue $f(\Delta y_i) = 0$

computation of Jacobienne $\frac{\partial f}{\partial y}(\Delta y_i)$

resolution of the linear system $\frac{\partial f}{\partial y}(\Delta y_i) \cdot \delta \Delta y_i = -f(\Delta y_i)$

searches linear: computation of ρ

actualization of the solution: $\Delta y_{i+1} = \Delta y_i + \rho \delta \Delta y_i$

For the linear stage of search:

The functional calculus is introduced $F(\rho) = \frac{1}{2} \|f(y_i + \rho \cdot \delta \Delta y_i)\|^2$.

The parameters of the linear search are fixed into tough:

Parameter of the rule of Armijo	$\omega = 0,1$
Limit min for the folding back	$\rho_{min} = 0,1$
Limit max for the folding back	$\rho_{max} = 0,5$
Nombre of iterations of cubic interpolations	$it_{max} = 2$

Note:

| Certain authors take values even smaller for ω (for example 10^{-4}).

4 Model installation of new of behavior

4.1 possible Methods

By means of preceding architectures, a new model of behavior can be developed relatively easily:

- the simplest method is explicit integration, by the method of `RUNGE_KUTTA` (see §22);
- implicit integration supplements by the method of `NEWTON` in environment `PLASTI` (see §33) requires a little more developments, in particular the computation of the jacobian matrix;
- it is however possible to directly use the development carried out for explicit integration (2 routines), without additional routine, to carry out computations with implicit integration, the jacobian matrix being then calculated by disturbance.

4.2 Model installation of new of behavior by the explicit method

to establish a model of behavior `XXX`, usable with `ALGO_INTE_=' RUNGE_KUTTA '`, it is enough to write 2 routines:

- a routine recovery of the material characteristics, called by the routine of shunting `LCMATE`
- a routine calculating the derivatives temporal of the local variables, called by the routine of shunting `LCDVIN`.

The update of the catalogs relates to only `DEFI_MATERIAU` and `C_COMP_INCR`, as well as the catalog of the constitutive law (cf [D5.04.01]).

4.3 Complete establishment of a new behavior in `PLASTI`

Without making a precise data-processing description of the developments (see on this subject [D5.04.01]), one can however describe the architecture of `PLASTI` and the places where to intervene to integrate a new behavior.

General architecture of `PLASTI` :

Call to the routines recovering the material parameters (at time t_i because possibly depending on the command variables) : `Specific LCMATE =>` routines of recovery of the material, communes with `RUNGE_KUTTA` (§3)

elastic Integration on Δt : `LCELAS =>` by default, isotropic or anisotropic linear elasticity; if necessary, call to specific routines for elastic design `XXXELA`.

Evaluating of the threshold for elastic integration: `LCCNVX =>` call to the routines of evaluating of threshold `XXXCVX`

If $seuil > 0$, then resolution by Newton: `LCPLAS` which calls `LCPLNL`, which carries out the loop of Newton. Its structure is the following one:

Initialization: `LCINIT =>` by default initialization to zero: $y^0=0$; if necessary, call to the specific routines of initialization `XXXINI`

Iterations of Newton:

Incrementing of the solution $y^k = y_{i-1} + \Delta y^k$

computation of residue $-f(\Delta y^k)$ `LCRESI =>` call to the routines calculating residue `XXXRES`

computation of the jacobian matrix $\frac{df}{dy_k}$: LCJACB => call to specific routines XXXJAC

resolution of the linear system $\frac{\partial f^l}{\partial \Delta y_k} d(\Delta y_k) = -f^l(\Delta y_k)$ by the method of Gauss

incrementing of $\Delta y_{k+1} = \Delta y_k + d(\Delta y_k)$

test of convergence LCCONV => by default, test on the relative error on the solution $|y_{k+1} - y_k| < \varepsilon |y_{k+1}|$; if necessary, call to specific routines XXXCVG

so not convergence, poursuite of the iterations, except if the maximum number is reached, in which case one leaves PLASTI with a return code, time step to activate local or total recutting (according to the user data: ITER_INTE_PAS for local recutting and DEFI_LIST_INST for total recutting).

If convergence, incrementing $y^k = y_{i-1} + \Delta y^k$ and output.

To establish a model of behavior XXX, it is thus necessary to write has minimum the routines:

XXXMAT: recovery of the material characteristics, called by LCMATE

XXXCVX for the evaluating of initial threshold

XXXRES : computation of the residue, called by LCRESI

XXXJAC : computation of the jacobian matrix, called by LCJACB.

One can also modify the following routines, if need be, to define in a particular way: the elasticity XXXELA called by LCELAS

initialization XXXINI called by LCMAT

the test of convergence XXXCVG, called by LCCONV.

The update of the catalogs relates to only DEFI_MATERIAU and C_COMP_INCR, as well as the catalog of the constitutive law (see [D5.04.01]).

4.4 Easy implicit establishment of a new model of behavior in environment PLASTI

It is possible to directly use the 2 routines (recovery of the material characteristics, derived from the local variables) used with ALGO_INTE_='RUNGE_KUTTA' to carry out an implicit integration. Indeed, the system of equations differentials solved by RUNGE_KUTTA can be written:

$$\sigma = \sigma_{i-1} + \Delta \sigma = \sigma_{i-1} + \mathbf{A}(\Delta \varepsilon_i^n - \Delta \varepsilon^{\text{th}} - \Delta \varepsilon^{\text{p}}(Y)) = G(Y)$$
$$\frac{dY}{dt} = F(Y, t; \sigma)$$

where Y all the local variables of the model represent. It is supposed here that the first values of Y are the components of the variation of plastic strain $\Delta \varepsilon^{\text{p}}$. The relation between the tensor of the stresses and the elastic part of the tensor of the strains is generally linear, but can be evaluated in a nonlinear way by a specific statement.

Once programmed the routine making it possible to calculate $\frac{dY}{dt} = F(Y, t; \sigma)$, it is possible to use it for an implicit integration, which consists in solving (cf 3.1):

$$R(\Delta Z) = 0 = \begin{bmatrix} R_1(\Delta Z) \\ R_2(\Delta Z) \end{bmatrix} \text{ with } \Delta Z = \begin{pmatrix} \Delta \sigma \\ \Delta Y \end{pmatrix} = Z_i - Z_{i-1}.$$

- The first system of equations represents the elastic relation stress-strain: $R_1(\Delta Z) = \sigma - G(Y) = 0$;
- The second expresses the laws of evolution of the various local variables:
 $R_2(\Delta Z) = \Delta Y - \Delta t \cdot F(Y, t; \sigma) = 0$, where G and F are calculated by the explicit routine.

To solve this system by the method of Newton proposed in environment PLASTI, that is to say:

$$\begin{cases} \frac{\partial f^l}{\partial \Delta Z_k} d(\Delta Z_k) = -f^l(\Delta Z_k) \\ \Delta Z_{k+1} = \Delta Z_k + d(\Delta Z_k) \end{cases}, \text{ IL is necessary to have the jacobian matrix } \frac{\partial f^l}{\partial \Delta Z_k}.$$

Its computation can be carried out automatically by disturbance: it is enough to use `ALGO_INTE='NEWTON_PERT'`.

Moreover, the tangent matrix is then calculated directly from the jacobian matrix according to the method described with the § 3.2.

In short, this process allows, with the two only routines necessary to explicit integration, to use an implicit integration, and to profit from a tangent matrix.

The update of the catalogs relates to only `DEFI_MATERIAU` and `C_COMP_INCR`, as well as the catalog of the constitutive law (cf [D5.04.01]).

The implementation of this method is illustrated in test SSNV225, for a creep model.

5 Bibliography

- 1) CROUZEIX Mr., MIGNOT A.L.: "Numerical Analysis of the differential equations", Masson, 1989.
- 2) CAILLETAUD G., PILVIN P.: "Use of polycrystalline models for the caclul by finite elements", European Review of the finite elements, vol.3, n°4, pp. 515-541, 1994.
- 3) ZIANI Mr., "Acceleration of the convergence of the methods of the Newton type for the resolution of the nonlinear systems", doctorate, 2008.

6 Description of the versions of the document

Version Aster	Author (S) Organization (S)	Description of modifications
4	P. GEYER, E. LORENTZ, C. VOGEL (EDF/RNE/EMA, IMA/MMN)	initial Text (integration clarifies)
10	S. GENIAUT, J.M. PROIX	Addition of implicit integration, and the linear search.
11.2	J.M. PROIX	Addition of the §4.44.4