Implicit integration and clarifies nonlinear relations of behaviors

Summary

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

The implicit digital method presented is that of Newton, with or without linear research.

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

Contents

1 General information on the integration of the laws of behavior

1.1 Algorithm of resolution of the quasi-static problem

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

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

Where:

- t represent the variable of moment
- $\mathbf{u}$ is the field of displacement taken starting from a configuration of reference
- $\mathbf{v}$ is the field of virtual displacement kinematically acceptable
- the relation $\mathbf{B}.\mathbf{u}=\mathbf{u}^d(t)$ corresponds to the boundary conditions imposed in displacements
- $\mathbf{L}^{\text{ext}}(t)$ is the external mechanical loading to which the structure is subjected
- $\mathbf{L}^{\text{int}}$ represent 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 of the work of the virtual deformations $\mathbf{Q}^T$ according to the following relation:

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

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

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

The formulation of the discretized quasi-static problem consists in expressing the balance of the structure for a continuation of moments of calculation $\left(t_i\right)_{1\le i\le I}$ who parameterize the loading, one will note the quantities at the moment t_i by the index i (for example $\mathbf{L}^{\text{int}}(\mathbf{u_i},t_i)=\mathbf{L_i}^{\text{int}}$) :

$$\begin{cases} \mathbf{L}_i^{\text{int}}+\mathbf{B}^T.\boldsymbol{\lambda}_i=\mathbf{L}_i^{\text{ext}} \\ \mathbf{B}.\mathbf{u}_i=\mathbf{u}_i^d \end{cases} \tag{4}$$

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 phase of correction:

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

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

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

1.2 Goal of integration

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

- the tensor of the constraints σ_i^n who checks the law of behavior, and who allows to calculate $\mathbf{L}_i^{\text{int},n}$ and internal 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:

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

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 constraints and the variables internal solutions of the problem as well as the tangent operator.

We present in the paragraphs according to two algorithms generals allowing the calculation 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 nonlinear equations, 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 the step of time. ; for that, one adopts a linear interpolation of displacement and variables of order (for example the temperature) during the time interval, which leads, for any increment of time with:

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

2 Integration clarifies of an incremental relation of behavior

To solve the nonlinear system relating to the law of behavior, one can use a method of integration explicit of type Runge Kutta of order 2 (option 'RUNGE_KUTTA'L' operand ALGO_INTE keyword factor BEHAVIOR) for the incrémentaux problems of nonlinear behavior.

The relations concerned currently are:

VISCOCHAB	Élasto-viscoplastic behavior of Chaboche, in two centers, with effect of memory and restoration
VENDOCHAB	Viscoplastic behavior with damage of Chaboche
POLYCRYSTAL	Homogenized polycrystalline élasto-viscoplastic behavior
MONOCRYSTAL	Single-crystal viscoplastic behaviors of metals
HAYHURST	Viscoplastic behavior with damage of Hayhurst

The relation POLYCRYSTAL 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 calculation of the stress field starting from an increment of deformation given while following the evolution of the internal variables.

2.1 Use of an explicit method

It is supposed 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 represent the whole of the internal variables of the model, including the variation of plastic deformation $\Delta\varepsilon^p$.
Their derivative in time are supposed to depend only on the internal variables and the constraints at the previous moment, and of the increase in total deflection provided by the total algorithm of Newton $\Delta\varepsilon_i^n$.

This assumption rests in particular on Lhas linear relation (law of Hooke) between the tensor of the constraint at the current moment and the elastic part of the tensor of the deformations: $\Delta\boldsymbol{\sigma}=\mathbf{A}(\Delta\boldsymbol{\varepsilon}_\mathbf{i}^\mathbf{n}-\Delta\boldsymbol{\varepsilon}^\mathbf{p})$, which makes it possible to express σ in function of $\Delta\varepsilon^p$ in the differential equations $\frac{dY}{dt}=F(Y,t;\boldsymbol{\sigma}(\Delta\varepsilon_i^n,\sigma_{i-1},Y))$

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 precision.

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 the step of time according to a necessary precision. 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:

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

The difference enters $Y^{(2)}$ (diagram of order 2) and $Y^{(1)}$ (diagram of order 1, Euler) provides an estimate of the error of integration and allows to control the size of the step of time h who 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 precision, the evolution of the internal variables at the points of Gauss where nonthe linearity of the behavior is most significant.

2.2 Control of the step of time

The strategy of the control of the step is defined on the standard basis of the difference between the two methods of integration: $\|Y^{(2)}-Y^{(1)}\|$ and of the precision required by the user η (keyword: RESI_INTE_RELA). The criterion selected is the following, where one notes $Y=(y_1,y_2,\ldots y_N)$:

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

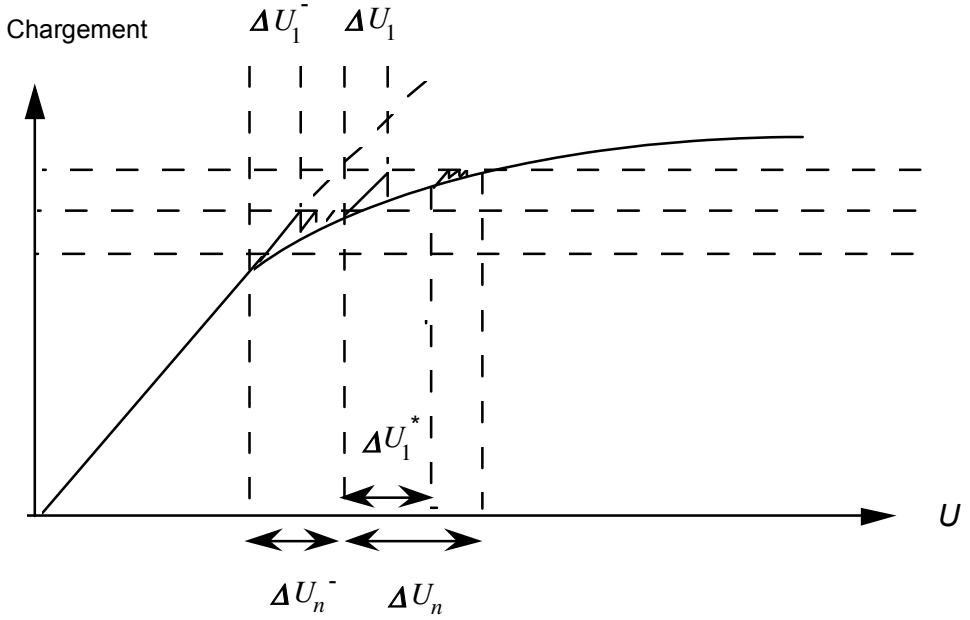
The parameter ε is fixed at 0.001. Precision of desired integration η must be coherent with the level of precision necessary for the total stage. The value by default $\eta = 10^{-6}$ often prove penalizing in terms of time calculation. It can be often to be increased by an order or two, in particular for the models élasto-visco-plastics, without real loss of precision (to be seen on a case-by-case basis).

If the criterion is not checked, the step of time Re-is cut out according to a discovery method. When the step of time becomes too weak ($h <$ precision machine), a code return is emitted, and the total recutting of the step of time 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 evaluation is however possible (but expensive) by a technique of disturbances (activated under BEHAVIOR the keyword TYPE_MATR_TANG=' PERTURBATION') . If one uses the elastic matrix for total balance, the iteration count 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 starting from the solution previously calculated: instead of considering the increment of displacement for the first iteration at the moment running (obtained via the elastic matrix), one initializes the increment of displacement searched for the first iteration ΔU_1 starting from the value ΔU_n^-, solution with the preceding increment (cf appears following page). The keyword PREDICTION='EXTRAPOLE' (under NEWTON) [U4.51.03] allows to initialize with the converged increment of the preceding step (balanced by the report of the steps of time). This estimate is projected on the field of displacements kinematically acceptable so that the final solution checks well the boundary conditions of Dirichlet.



3 Implicit integration of an incremental relation of behavior.

To solve the systems of nonlinear equations relating to the incrémentaux problems of nonlinear behavior (operand RELATION keyword factor BEHAVIOR). it is possible to use a method of integration implicit of Newton-Raphson type, with which one can associate a linear stage of research (respectively option 'NEWTON' and option 'NEWTON_RELI' operand ALGO_INTE keyword factor CONVERGENCE).

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

The relations concerned are for example:

VISCOCHAB	Élasto-viscoplastic behavior of J.L. Chaboche, in two centers, with effect of memory and restoration
MONOCRYSTAL	Single-crystal viscoplastic behaviors of metals
IRRAD3M	Elastoplastic behaviour under irradiation of the stainless steels
…	

Relations VISCOCHAB and MONOCRYSTAL can also be integrated by the explicit method exposed to the §2.

3.1 Implicit integration by the method of Newton

Environment PLASTI allows in *Code_Aster* to integrate by a method of Newton, 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, variables internal) at the previous moment t_{i-1}, and the increase in total deflection provided to the point considered by the total algorithm of Newton $\boldsymbol{\varepsilon}(\Delta \mathbf{u}_i^n)$, noted too $\Delta \varepsilon_i^n$, it is necessary to find the solution of the system:

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

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

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

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

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

It remains to calculate the matrix jacobienne system: $\dfrac{df}{dy_k}$. 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 criterion of stop of the iterations relates, according to the behaviors, to the nullity of the residue: $\frac{f(y_k)}{f(y_0)}<\varepsilon$ or stationnarity of the solution: $|y_{k+1}-y_k|<\varepsilon$.

Initial values y_0 by defaults are selected worthless, but can take particular values according to the behaviors.

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

Behaviors	Criterion of stop	Initialization
VISCOCHAB	$\|y_{k+1}-y_k\|<\varepsilon\|y_{k+1}\|$	Explicit solution and modification of initialization in the event of bad convergence
MONOCRYSTAL	$\|f(y_k)\|<\varepsilon\|f(y_0)\|$	Worthless solution
IRRAD3M	$\|f(y_k)\|<\varepsilon\|f(y_0)\|$	Elastic solution

The method used allows a local recutting of the step of time, 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, $\left(f\left(\Delta y_{CV}\right)=0\right)$ is checked at the end of the increment. For a small variation of f, by considering this time the increase in total deflection $\Delta \varepsilon_i^n$ like variable and not like parameter, the system remains with balance and one checks $df=0$, i.e.:

$$\frac{\partial f}{\partial \Delta \boldsymbol{\sigma}}\delta\Delta\boldsymbol{\sigma}+\frac{\partial f}{\partial \Delta \boldsymbol{\varepsilon}_{\mathbf{i}}^{\mathbf{n}}}\delta\Delta\boldsymbol{\varepsilon}_{\mathbf{i}}^{\mathbf{n}}+\frac{\partial f}{\partial \Delta \boldsymbol{\varepsilon}^{\mathbf{p}}}\delta\Delta\boldsymbol{\varepsilon}^{\mathbf{p}}+\frac{\partial f}{\partial \Delta X_1}\delta\Delta X_1+\frac{\partial f}{\partial \Delta X_2}\delta\Delta X_2+\frac{\partial f}{\partial \Delta p}\delta\Delta p=0$$

This system can be still written:

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

where ΔZ represent the unknown factors complementary to $\Delta\boldsymbol{\sigma}$ in y .

By writing the matrix jacobienne in the form:

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

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

$$\left(\frac{\partial\Delta\boldsymbol{\sigma}}{\partial\Delta\boldsymbol{\varepsilon}}\right)_i^n=\left(Y_0-Y_1Y_3^{-1}Y_2\right)^{-1}$$

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

3.3 Implicit integration by the method of Newton with linear research

So to improve the robustness of the method of Newton, one gives the opportunity of activating linear research.

3.3.1 Presentation of the algorithm linear general of research

That is to say a system of non-linear equations to solve: $f(x)=0$

One associates with f a functional calculus F.

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

- calculation of the direction of descent: $d_k=-[\nabla f(x_k)]^{-1}.f(x_k)$
- calculation of the step of advance: ρ such as $F(x_k+\rho.d_k)<F(x_k)$, the best being it ρ who minimizes the functional calculus $\rho=\operatorname{argmin}[F(x_k+\rho.d_k)]$
- actualization of the unknown factors: $x_{k+1}=x_k+\rho d_k$

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

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

Confidence interval:

To know if one ρ decrease 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 fixed reality lower than 1. This expression can be represented graphically by Figure 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 decrease sufficiently the functional calculus if $\rho\in[0;0,871]$.

Principle of the calculation 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 0 and the 2 last ρ. The paragraph below details the calculation of ρ.

Details of the calculation 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 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 an upper limit ρ_{max} (for example $\rho_{max}=0,6$). Thus, if $\rho_q<\rho_{min}$ then $\rho_q=\rho_{min}$ and if $\rho_q>\rho_{max}$ then $\rho_q=\rho_{max}$.

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

If the rule of Armijo is not satisfied, i.e. if $F(\rho_q)>F(0)+\omega F'(0)\rho_q$, then one 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.

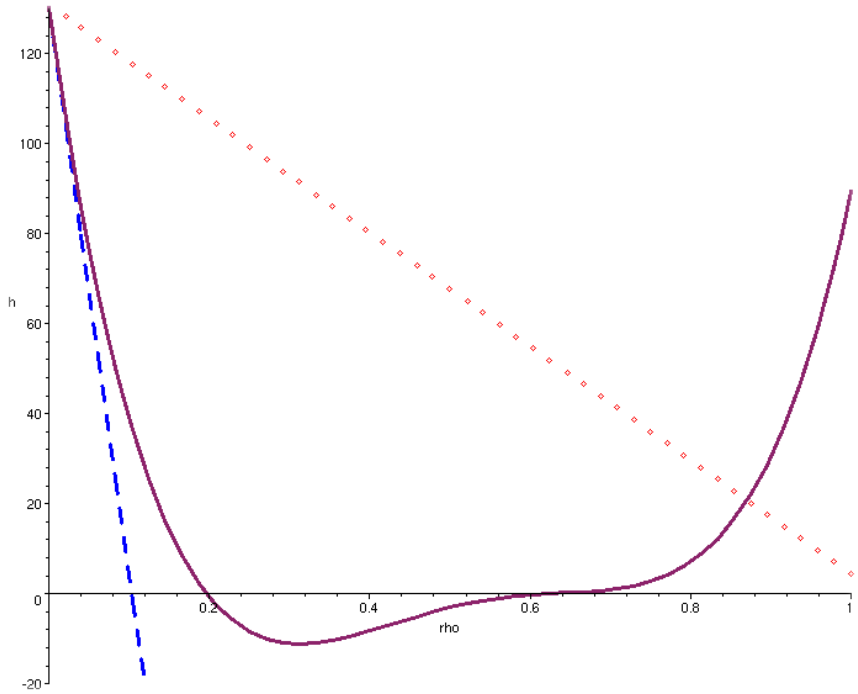


Figure 3.3.1-a : illustration of the rule of Armijo

3.3.2 Implementation of linear research in PLASTI

The system to be solved is:

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

The following iterations of Newton are made:

- update of the variables at the moment "+"
- calculation of the residue $f(\Delta y_i)=0$
- calculation of Jacobienne $\frac{\partial f}{\partial y}(\Delta y_i)$
- resolution of the linear system $\frac{\partial f}{\partial y}(\Delta y_i).\delta\Delta y_i=-f(\Delta y_i)$
- linear research: calculation of ρ
- actualization of the solution: $\Delta y_{i+1}=\Delta y_i+\rho\,\delta\Delta y_i$

For the linear stage of research:

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

The parameters of linear research are fixed into hard:

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	$\rho_{max}=0,5$
Iteration count of interpolations cubic	$it_{max}=2$

Note:

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

4 Establishment of a new model of behavior

4.1 Possible methods

By using 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 §2) ;
– implicit integration supplements by the method of NEWTON in the environment PLASTI (see §3) request a little more developments, in particular the calculation of the matrix jacobienne;
– it is however possible to directly use the development carried out for explicit integration (2 routines), without additional routine, to carry out calculations with implicit integration, the matrix jacobienne being then calculated by disturbance.

4.2 Establishment of a new model 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 data material, called by the routine of shunting LCMATE
- a routine calculating the derivatives temporal of the internal variables, called by the routine of shunting LCDVIN.

The update of the catalogues only concerns DEFI_MATERIAU and C_COMPORTEMENT, as well as the catalogue of the law of behavior (cf [D5.04.01]).

4.3 Establishment supplements 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 places where to intervene to integrate a new behavior.

General architecture of PLASTI :

• Call to the routines recovering the parameters material (at the moment t_i because possibly depending on the variables of orders) : LCMATE => specific routines of recovery of 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 the elastic design XXXELA.

• Evaluation of the threshold for elastic integration: LCCNVX => call to the routines of evaluation of the threshold XXXCVX

• If $seuil > 0$, then resolution by Newton: LCPLAS who 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 $y^k = y_{i-1} + \Delta y^k$

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

- calculation of the matrix jacobienne $\frac{df}{dy_k}$: LCJACB => call to the 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 the specific routines XXXCVG
  - so not convergence, continuation of the iterations, except if the maximum number is reached, in which case one leaves PLASTI with a code return, to activate local or total recutting step of time (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 exit.

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

- XXXMAT : recovery of the data material, called by LCMATE
- XXXCVX for the evaluation of the initial threshold
- XXXRES : calculation of the residue, called by LCRESI
- XXXJAC : calculation of the matrix jacobienne, called by LCJACB.
- One can also modify the following routines, if need be, to define in a particular way:
- elasticity XXXELA called by LCELAS
- initialization XXXINI called by LCINIT
- the test of convergence XXXCVG, called by LCCONV.

The update of the catalogues only concerns DEFI_MATERIAU and C_COMPORTEMENT, as well as the catalogue of the law of behavior (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 data material, derived from the internal variables) used with ALGO_INTE_=' RUNGE_KUTTA' to carry out an implicit integration. Indeed, the system of differential equation solved by RUNGE_KUTTA can be written:

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

where Y represent the whole of the internal variables of the model. It is supposed here that the first values of Y are the components of the variation of plastic deformation $\Delta\varepsilon^p$. 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 allowing to calculate $\frac{dY}{dt} = F(Y, t; \boldsymbol{\sigma})$, it is possible to use it for an implicit integration, which consists in solving (cf. 3.1) :

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

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

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

$$\begin{cases} \dfrac{\partial f^l}{\partial \Delta Z_k} d(\Delta Z_k)=-f^l(\Delta Z_k) \\ \Delta Z_{k+1}=\Delta Z_k+d(\Delta Z_k) \end{cases}$$, I L is necessary to have the matrix jacobienne $\dfrac{\partial f^l}{\partial \Delta Z_k}$.

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

Moreover, the tangent matrix is then calculated directly starting from the matrix jacobienne 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 catalogues only concerns DEFI_MATERIAU and C_COMPORTEMENT, as well as the catalogue of the law of behavior (cf [D5.04.01]).

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

5 Bibliography

1) CROUZEIX MR., MIGNOT A.L.: “Digital 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 non-linear systems”, doctorate, 2008.

6 Description of the versions of the document

Version Aster	Author (S) Organization (S)	Description of the 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 linear research.
11.2	*J.M. PROIX*	Addition of the §4.4