The Councils of implementation of nonlinear computations

Résumé

the objective of this document is to give advices to an user wishing to carry out nonlinear computations with *Code_Aster*. The user of linear computations will find there also useful information and advices.

The aspects according to will be approached:
- mesh and modelization,
- loadings and boundary conditions,
- materials and constitutive laws,
- nonlinear resolution,
- management memory/time and tweaking,
- check out of the error and quality.

This document does not treat questions specific to the dynamics.

1 Mesh and modelization

the choice of the finite elements influences directly the solution. By choice, one hears the choice of the degree of the approximation element-finish as well as the choice of a formulation. The degree of the approximation being generally connected to the degree of the meshes, the paragraphs according to treat choice of the degree of the mesh and the choice of a formulation element-finish, with a particular item for the problems of incompressibility.

1.1 General information on the choice of the degree of the mesh and the choice of the finite elements

In a general way, the use of a linear mesh is advised in thermal and the use of a quadratic mesh is advised in mechanics. Command `CREA_MAILLAGE` (`LINE_QUAD` and `QUAD_LINE`) makes it possible to pass from a linear mesh to a quadratic mesh and vice versa, but does not allow to generate quadratic elements on curved board. It is preferable besides to directly generate the quadratic elements with the tool for mesh (the modulus of mesh of Salome-Meca for example). For the fracture mechanics, the use of elements of Barsoum in crack tip improves quality of the result (command `MODI_MAILLAGE` / `NOEUD_QUART`).

The choice of the finite elements is carried out in command `AFFE_MODELE`. In thermal, it is advised to use lumped elements (`_DIAG`). In mechanics, the finite elements classically used are the isoparametric elements (`3D`, `D_PLAN`, `C_PLAN`, `AXIS`). However, these elements are badly adapted to the quasi-incompressible problems. The choice of a formulation answering this problem is the subject of the next paragraph.

For more details, to see [U2.01.10] “Notice of use on the choice of the finite elements”

1.2 Choix of a formulation to deal with the quasi-incompressible problems

a problem is quasi-incompressible if the Poisson's ratio is close to 0.5 ($\nu > 0{,}45$) or if plastic strain rate is high. That results in an oscillation of the trace of the stresses. There exist 6 formulations for dealing with this problem in *Code_Aster* whose choice depends on the type of meshes, the model of strain and the ratio quality/cost computation.

The possible formulations are:

- in small strains: `_SI`, `_INCO`, `_INCO_UP`, `INCO_OSGS`
- in large deformation: `_INCO_GD`, `_INCO_LOG`.

All the formulations are compatible with modelizations `3D`, `D_PLAN` and `AXIS` but in plane stresses (`C_PLAN`) only the under-integrated formulation (`_SI`) is possible.

The table below has compatibility between quasi-incompressible and standard formulation of meshes (restricted with main meshes 3D).

	`_SI`	`_INCO`	`_INCO_UP`	`_INCO_OSGS`	`_INCO_GD`	`_INCO_LOG`
`HEXA20`	X	X	X		X	X
`PENTA15`		X	X		X	X
`PYRAM13`						
`TETRA10`	X	X	X		X	X
`HEXA8`	X			X		
`PENTA6`				X		
`PYRAM5`				X		
`TETRA4`			X	X		

Dans the general case, it is thus necessary to mix various formulations to cover all the types of meshes (for example, as no formulation manages meshes `PYRAM13`, it is also necessary to affect an isoparametric modelization).

If one remains in small strains, it is advised to use a mixture of formulations (mesh which can comprise several types of meshes):

	quality	cost
`MODELISATION= ("3D", "3D_INCO" , "3D_INCO_OSGS")`	+++	+++
`MODELISATION= ("3D", "3D_INCO_UP", "3D_INCO_OSGS")`	++	++
`MODELISATION= ("3D", "3D SI" , "3D_INCO_OSGS")`	+	+

Dans the case of large deformation, the choice also depends on the model of large deformation:

- the _INCO_GD `formulation` is compatible only with the model of strain `SIMO_MIEHE`
- the _INCO_LOG `formulation` is compatible only with the model of strain `GDEF_LOG`

Dans the case of large deformation, it is advised to use the following formulations (only for quadratic meshes!):

If `DEFORMATION='SIMO_MIEHE'`	`MODELISATION= ("3D", "3D_INCO_GD")`
If `DEFORMATION='GDEF_LOG'`	`MODELISATION= ("3D", "3D_INCO_LOG")`

Pour more details, to see [U2.01.10] "Notice of use on the choice of the finite elements"

2 Loadings, boundary conditions, Cœur

2.1 initial conditions of the problem

Les loadings and boundary conditions are important for a good modelization. Their influence on the results is most of the time direct, so that one can check their application by a simple preliminary computation, for example using `MECA_STATIQUE`, or by visualizing the boundary conditions *via* key word `CONCEPT` of `IMPR_RESU` to format `MED`.

With regard to the boundary conditions, it is often a question of introducing the minimum of it to lock displacements of rigid solid, therefore to avoid having floating body in structure, which would cause a null pivot or a singular matrix at the time of the resolution.

Certain boundary conditions are nonlinear, for example the unilateral contact. Their checking cannot thus be made using the first linear computation (`MECA_STATIQUE`): the various objects in contact should not have rigid body motion [U2.04.04] Notice of use of the contact. If it is the case, one can avoid the null pivots by adding discrete elements (springs) of low rigidity.

The loadings (other that blockings) can consist either of imposed conditions of displacement, or in imposed forces, or of standard imposed initial field.

In all the cases, it is important to represent reality well. One will be able to use with profit the conditions of symmetry or antisymetry [1].

In a general way, a condition of type imposed displacement does not provide the same results as a condition of the type forces imposed: a displacement imposed on part of the border imposes that displacement (imposed) is constant spaces some on this part, therefore brings rigidity, even of the singularities. Let us take the example of the punching of a solid mass: the displacement imposed on edge of the induced punch of the singularities of stresses of comparable nature that those which are met at the bottom of a crack.

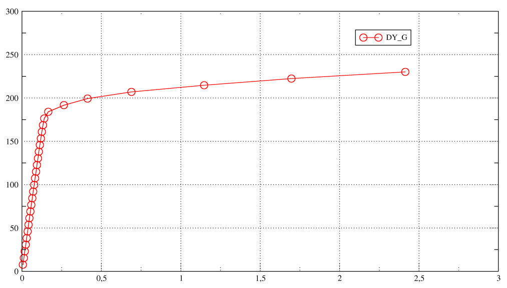
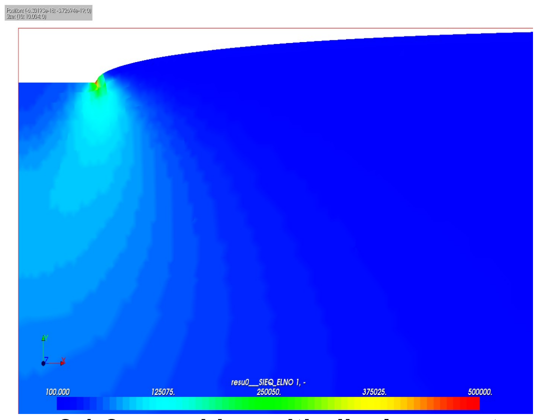


Figure 2.1-1: Example of curve force-déplacementFigure



2.1-2: punching with displacement imposed

Moreover, for nonlinear computations, the control of computation is not the same one: in the event of softening or of Yield-point load, the loading with imposed force can be illicit (beyond the limiting loading). Let us take for example punching (see Figure 2.1-1 and 2.1-2): the response in terms of resultant force on the punch according to displacement shows that the more the imposed force approaches the Yield-point load, the more convergence will be difficult, until being impossible beyond the Yield-point load. If the correct modelization requires to apply an imposed force, the problem can be solved *via* the control of the force imposed compared to the displacement of a point or a set of points (method of continuation or length of arc) (see [R5.03.80] Continuation methods of the loading).

Documentations of use relating to the application of the loadings and boundary conditions are [U4.44.01] AFFE_CHAR_MECA and [U4.44.03] AFFE_CHAR_CINE .

2.2 Precautions for the application of the loadings

It is useful for the application of the loadings to traverse the various key words of [U4.44.01] AFFE_CHAR_MECA. Attention with the vocabulary, in particular, the loadings of the type FORCE_FACE, PRES_REP, FORCE_CONTOUR, FORCE_ARETE, FORCE_INTERNE, FORCE_COQUE, FORCE_TUYAU correspond to distributed forces (linear, surface, voluminal) and **are expressed in unit of stresses** (for example Pa in unit S.I). Only the FORCE_NODALE are expressed in unit of forces (N), and the distributed forces of beams (FORCE_POUTRE) in units of force divided by a length.

With regard to pressures (PRES_REP, FORCE_COQUE/PRES), the pressure applied is positive according to the contrary meaning of the norm to the element. It is strongly advised to reorientate these norms *via* the operator MODI_MAILLAGE, key words ORIE_PEAU_*, ORIE_NORM_COQUE.

For continuums 2D, 3D, the use of specific forces (FORCE_NODALE) is to be proscribed, because it always involves singularities.

The application of loads function of time can be carried out in two ways:

- either by defining constant loads, then by applying a multiplying coefficient function of time to these loads at the time of the resolution (key word FONC_MULT under EXCIT in STAT_NON_LINE),
- or by directly defining loads function of time via AFFE_CHAR_MECA_F.

If the loadings are of imposed displacements type, those can be introduced either by AFFE_CHAR_MECA (_F), or by AFFE_CHAR_CINE (_F). The use of this last command allows a time-saver computation which can be appreciable into nonlinear (because she does not add Lagrange multipliers, there are thus less equations to solve).

The command variables do not form part of commands AFFE_CHAR_MECA (_F). They are however, in many modelizations, comparable to loadings: for example thermal dilation can with it only generate stress states and strains leading to non-linearities of behavior. The command variables are applied via AFFE_MATERIAU.

It is necessary however to announce here a precaution of modelization concerning these command variables: indeed, as soon as computation is incremental, STAT_NON_LINE uses with each time step the increment of command variable for compute the corresponding strains (cf for example [R5.03.02] elastoplastic Intégration of the behavior models of Von Mises). In general, it preferable that at initial time, the structure is not forced, not is deformed. If it is not the case, it is necessary to add one preliminary time where the structure is at rest, see for example test FORMA30 [V7.20.101] FORMA30 - hollow Cylindre thermo-elastic. In the event of detection of stresses due to the command variables at initial time one obtains alarm:

```
------------------------------------------------------------------------
<A> <MECANONLINE2_97>:
- > the initial command variables induce incompatible stresses:
the initial state (before the first time of computation) is such as the command variables
(temperature, hydration, drying…) lead to not balanced stresses.

- > Risk & the Council:  in the case of an incremental resolution, one considers only the
variation of the command variables between previous time and current time. One  thus does not
take into account possible incompatible stresses due to these initial command variables.
To take account of these stresses you can:
-start from one former fictitious time where all the command variables are null or equal to the
values of reference
-to choose adapted values of reference
Pour more information, see the documentation of STAT_NON_LINE (U4.51.03) key word EXCIT, and
test FORMA30 (V7.20.101).
------------------------------------------------------------------------
```

2.3 The various types of boundary conditions

Les the simplest boundary conditions are mainly used to introduce conditions of symmetry, on the one hand, and to prevent rigid solid motions, on the other hand. When they are of standard imposed degree of freedom, one can use either AFFE_CHAR_MECA, or (to optimize time computation) AFFE_CHAR_CINE [U2.01.02] Notice of use of the boundary conditions treated by elimination.

A methodological advice: it is always preferable to rather apply the boundary conditions to mesh groups than to nodes groups or lists of nodes or meshes. Indeed, the mesh groups correspond to geometrical areas, and are preserved at the time of a refinement of the mesh (manual or using Homard), whereas the nodes groups are modified. This is why in AFFE_CHAR_MECA key words DDL_IMPO, FACE_IMPO, LIAISON_UNIF, as all the key words of AFFE_CHAR_CINE comprise key word GROUP_MA, to thus privilege.

It is sometimes necessary to impose linear relations between degrees of freedom. Key words LIAISON_* of AFFE_CHAR_MECA make it possible to introduce this kind of relation. The elementary relations can be defined using LIAISON_DDL, but that becomes tiresome if the equations to be written are numerous. Functionalities moreover high level are available, for example:

- LIAISON_SOLIDE to rigidify part of structure, (in small strains and small displacements only) [R3.03.02] Conditions of connection of solid body ;
- LIAISON_UNIF to ensure that part of the border will keep same displacement (unknown a priori); [U4.44.01] AFFE_CHAR_MECA
- LIAISON_MAIL, LIAISON_GROUP to connect two edges,
- and other specific key words of AFFE_CHAR_MECA.

Moreover it is possible to connect (with the energy meaning) modelizations of different nature, via LIAISON_ELEM and several options (in small strains and small displacements only)

- "2D_POU", "3D_POU", beams or discrete with elements 2D or 3D [R3.03.03] Raccords 2D-beam and 3D-beam;
- "COQ_TUYAU", "3D_TUYAU", elements pipe with plates and shells, 3D;
- "COQ_POU", beams or discrete with plates and shells [R3.03.06] Connection shell-beam.

Note: in certain cases, the boundary conditions or the loadings of type imposed displacement are not appropriate because they bring too local rigidity. It can be interesting to override these conditions by a connection between the part of the border concerned and a discrete element, to which will be applied the imposed conditions of displacement or a load vector force imposed. That means that displacements of the border will be equal on average to the displacement of the discrete element, without introducing secondary stresses. On the other hand, this process introduces additional degrees of freedom (Lagrange multipliers), the resolution of the linear systems can thus be more expensive.

The unilateral contact, with or without friction, is a kind of boundary conditions private individual. It is strongly nonlinear, and its processing in STAT_NON_LINE / DYNA_NON_LINE requires additional iterations to obtain convergence. It is strongly advised to consult the document [U2.04.04 Notice of use of the contact] to modelize these phenomena correctly.

2.4 The initial conditions

Dans AFFE_CHAR_MECA, it is possible to define loadings of average strains or mean stresses, overall uniform (key words PRE_EPSI, PRE_SIGM).

One should not confuse these loadings and the strains and initial stresses used into nonlinear, because these quantities do not intervene directly in the statement of the constitutive law, but only with the second member.

The initial stress fields, displacements and intern variables are with being provided directly in STAT_NON_LINE (key word ETAT_INIT). To build these fields, it can be useful to consult the document [U2.01.09] analytical Définition of an initial stress field and a field of intern variables.

3 Materials and Choix

3.1 behaviors of the constitutive law

the choice of the constitutive law is of course function of the material which one modelizes, but also phenomena with treating: for example, the same steel will be elastoplastic with low-temperature, and viscoplastic at high temperature. The values of the parameters of these models (in `DEFI_MATERIAU`) are often identified in a range of strain, velocity, temperature quite specific.

•For the elastoplastic behaviors, see [U2.04.03] Choix of the behavior élasto- (visco) - plastic
•Pour the models with damage (case of the concrete for example), to see [U2.05.06] Réalisation of computations of damage in quasi-static
•Pour the metallurgy, to see [U2.03.04] Notice of use for computations thermometallomecanic on steels
•Pour the porous environments in THM, to see [U2.04.05] Notice of use of model THM and [R7.01.11] Model of behavior THHM
•Pour the use of elements CZM, to see [U2.05.07] Notice of use of the models of cohesive areas

3.2 Word-keys of `COMP_INCR`

3.2.1 `DEFORMATION`

This key word makes it possible to define the assumptions used for the computation of the strains : by default, one considers small displacements and small strains. The type of strain used can have a great influence on computation as soon as a component of the strains exceeds a few % (typically 5%).

•**`PETIT`** : small strains, small displacements. Linearity of the operator strain.

•**`GROT_GDEP`** : allows to treat large rotations and large displacements, but while remaining in small strains. This is particularly useful for slender structures (modelized out of beams, shells, or 3D) and the study of buckling (see for example the test [V6.02.134] SSNL134 - Failure elastoplastic of the gantry of Lee).

With regard to the models very-elastics of the type `ELAS_VMIS` , they are not adapted to the large deformation (loss of existence of the solution, see the §2,1 of [R5.03.20] nonlinear elastic Behavior model in large displacements). It is necessary to use either a model of large deformation with `VMIS_ISOT` , or behavior `ELAS_HYPER` .

•**`GDEF_LOG`** : model large deformation, using a logarithmic curve strain measurement, and which makes it possible to use elastoplastic constitutive laws with isotropic or kinematical hardening (see the list of the behaviors in [U4.51.11] Comportements nonlinear). The relation stress-strain being hypo-elastic, this formulation is restricted with the weak elastic strain (but large deformation).

•**`SIMO_MIEHE`** : model large deformation of the constitutive laws lean on a criterion of Von Mises with isotropic hardening, and all the behaviours with isotropic hardening associated with an undergoing material of the metallurgical phase changes. The relation stress-strains elastic is very-elastic, which makes it possible to treat the elastic large deformation (for little that has a meaning for the material used).

•other formulations exist, to see [U4.51.11] .

3.2.2 `ALGO_INTE` , `ITER_INTE_MAXI` , `RESI_INTE_RELA` , `ITER_INTE_PAS`

Permet to specify the type of diagram of integration to solve the nonlinear equation or the system of equations formed by the constitutive equations of the models of behavior to intern variables. A method of by default resolution is planned for each behavior. However, it is possible to modify the method of by default resolution for a certain number of behaviors (see [U4.51.11]).

In the case of an iterative resolution (i.e. if ALGO_INTE is not ANALYTIQUE), key words ITER_INTE_MAXI and RESI_INTE_RELA define the maximum number of iterations and the relative residue to integrate the behavior. If this integration fails, it is possible to subdivide the time step, either locally (key word ITER_INTE_PAS) or overall (command DEFI_LIST_INST). The default value of ITER_INTE_MAXI is 20. That can be insufficient for certain behaviors (MONOCRISTAL for example). Do not hesitate in this case to increase this parameter (100 for example). On the other hand it is strongly disadvised increasing RESI_INTE_RELA (10^{-6} by defect), under penalty of obtaining not converged solutions.

The local subdivision of the time step is average to improve the robustness of local integration, on the other hand it does not make it possible to provide a coherent tangent matrix (loss of the quadratic convergence of the total problem).

3.2.3 POST_ITER='CRIT_RUPT'

Definition of a rupture criterion in critical stress in postprocessing of the iterations of Newton, with each time step. If the greatest average principal stress in an element exceeds a given threshold σ_c, the Young modulus is divided with the following time step by a coefficient. These two coefficients are defined under key word CRIT_RUPT of operator DEFI_MATERIAU.

3.2.4 RESI_RADI_RELA

Mesure of the error due to the discretization in time, directly connected to the rotation of the norm on the surface of load. One computes the angle between the norm with the plasticity criterion at the beginning of the time step and the norm with the plasticity criterion computed at the end of the time step. The time step is cut out (*via* DEFI_LIST_INST) if the error is higher than the tolerance defined by the user.

This criterion is operational for the elastoplastic behaviors of Von Mises with hardening isotropic, kinematical linear and mixed and for the behaviors élasto-visco-plastics of Chaboche.

4 Nonlinear resolution

Les 3 great types of non-linearities in *Code_Aster* are the following:

- non-linearity related to the behavior of the material (for example plastic);
- non-linearity related to the geometry (for example in large displacements);
- non-linearity related to contact-rubbing.

The general advices of resolution of nonlinear problems are the subject of a specific document which one strongly recommends the reading: [U2.04.01] "the Councils of use of `STAT_NON_LINE` ".

Before any nonlinear computation, it is essential to make sure that computation functions correctly in linear elasticity. Non-linearities could then be added the ones after the others.

One briefly recalls the major advices on the temporal discretization and the numerical parameters of nonlinear resolution in the next paragraphs.

4.1 Non-linearity and temporal discretization

the resolution of a nonlinear problem generally requires to apply the external loading gradually, by increment of load. Thus time (or pseudo-time into quasi-static) is discretized in time step and to each time step an increment of load corresponds (see §2.25). The smaller the time step is, the less the problem is nonlinear thus easier to solve. In order to authorize the under-spanning of the time step in the event of failure of convergence, it is essential to use command `DEFI_LIST_INST` (for more details on management of the list of times, see the paragraph §3.1 document [U2.04.01])

Pour to check the coherence of the data (system of units, boundary conditions, characteristics elementary, effect of the command variables), it is always useful to carry out a first linear elastic design `(STAT_NON_LINE/RELATION='ELAS'`, or `MECA_STATIQUE)`, before any nonlinear study, then to add nonthe linearities ones after the others.

4.2 Numerical parameters for the nonlinear algorithm of resolution

By default, the nonlinear algorithm of resolution is based on the method of Newton-Raphson (`STAT_NON_LINE / METHODE='NEWTON'`).

It is advised to use a tangent matrix reactualized with each iteration of Newton: `REAC_ITER=1` in order to facilitate the convergence of the algorithm (see the paragraph §2.2 document [U2.04.01]).

It is strongly recommended not to increase the convergence criterion of the method of Newton (RESI_GLOB_RELA=10^{-6} by defaults). Other convergence criteria are also available (see the paragraph §3.3 document [U2.04.01]).

Note:

- For lenitive problems, *Code_Aster* makes it possible to use an alternative method with the method Newton-Raphson: it is the method `IMPLEX` which is a ruggedized but approximate method (see the paragraph §4.5 document [U2.04.01]).
- For a tweaking of the CPU time, *Code_Aster* makes it possible to use an inaccurate method of Newton (provided that the selected linear solver is an iterative solver): it is method `NEWTON_KRYLOV`.

5 Management memory/time and tweaking

Avant de speech of tweaking, the first question which one pose is how choose the parameters of execution in `Astk` : total memory and time so that the first launching of computation finishes correctly. Then, once computation will have passed for the first time, it will be always possible to try to optimize it according to information which one will have recovered of the 1st computation.

5.1 First execution of a computation

the knowledge of the memory size which one lays out is necessary. The response depends of course on the object computer: centralized server, local machine…. On a local machine, it is always possible to authorize the maximum memory available whereas on the centralized server, it can be convenient not to ask too much memory under penalty of having to wait until the associated class of job is released.

Estimate of the memory necessary : into quasi-static, the total memory is generally function of the size of the linear systems to solve. For the isoparametric finite elements, it is rather easy to consider the size total of these systems. The number of degrees of freedom can be estimated by the relation: (many nodes of the mesh) X (dimension of the problem (2 or 3))
This relation does not take into account the ddls relating to the dualized boundary conditions. But generally, this share is weak. If the finite elements are not isoparametric (mixed formulations for example), this estimate is more delicate.

In any case, the size of the linear systems is displayed with each resolution in the file `.mess` : `full number of equations` or `the matrix is of size N equations`. It is **important to know this number** (although with him only, it is not always a completely reliable indicator, because the memory and the time necessary with the resolution of a linear system depend on good of other parameters: cut bandwidth, many non-zero terms in the matrix, renumerotor… but that would take us along too far). Thus if one cannot estimate the size of the linear system, it is interesting **for the first time to launch the study in degraded mode** : reading of the mesh, assignment of element-finish, the material and resolution (`MECA_STATIQUE` or `STAT_NON_LINE`) with direct solver `MUMPS`. Launch the study simplified in interactive mode with interactive follow-up, then once stop computation information on the known size of the system.

In the idéal1Idéal[1], it is necessary to have approximately 30 Go to make pass a computation to a million degrees of freedom (case of the cube with a grid in `HEXA8`, whose face is embedded by dualisation), the memory minimale2Minimale[2] being 9 Go. The table below gives orders of magnitude of the “ideal” memory (or optimal) and minimal to solve a system of size given (with direct solver `MUMPS`).

Many degrees of freedom	Mémoire minimal	Mémoire “ideal”
200000	1 Go	4 Go
400000	2.5 Go	9 Go
600000	4 Go	15 Go
800000	6 Go	23 Go
1000000	9 Go	31 Go

Une fois the estimated necessary memory, one can launch complete computation with direct solver `MUMPS`. Following this 1st computation, it is very interesting to look at:

1 meaning here management of memory `IN_CORE`
2 wanting to say here for a management `OUT_OF_CORE`

- In the file `.mess` total memory (`JEVEUX` + python + external bookshops) used by computation: this information is displayed at the end of the file: see `MAXIMUM OF MEMOIRE UTILISEE PAR LE PROCESS`;
- In the file `.resu` : the time of resolution (`MECA_STATIQUE` / `STAT_NON_LINE`) compared to the total time of computation.

5.2 Tweaking of a computation

the tweaking of a computation on the aspects time and memory is an interesting operation but a little delicate because there does not exist miraculous formula. Moreover it is not always easy to make the distinction between tweaking of the memory and tweaking of the CPU time, therefore in the next paragraphs, one gives runways of tweaking “in the broad sense” and one returns towards the adequate documents.

Parallelism
the use of parallelism is certainly the easiest solution and surest to decrease the CPU time (and sometimes also memory necessary). Parallelism consists in executing operations on several processors at the same time. The installation of parallelism is effective only if the time spent in `STAT_NON_LINE` is dominating compared to the total time of computation. For that, it is enough to choose an adequate solver (example: `MUMPS`, `PETSC`), a version MPI of *Code_Aster* and to specify the number of processors in `Astk` (finely `Options`/`mpi_nbcpu`). On the server centralized `aster4`, one advises to start by choosing `mpi_nbcpu=2`, to observe the saving of time then to start again with `mpi_nbcpu=4`, then possibly `mpi_nbcpu=8`. It is possible to choose `mpi_nbcpu=16` but one is likely to wait a long time before all these processors are available, therefore before computation boots (all depends on the load machine!).
Many advices on parallelism are given in the document [U2.08.06] “Notice of use of parallelism”. Finer information over the times spent in each part of the resolution is also displayed, and can be used for better parameterizing computation (time, memory, many processors…). One returns for that to documentation [U1.03.03] “Indicateurs of performance of a computation (time/memory)”.

The linear solver
Suivant the size of the matrix, the use of an iterative solver (as `PETSC`) instead of a direct solver will allow a saving of time. For information, it is considered that an iterative solver is faster than a direct solver (by default) starting from approximately 200 000 equations in 3D. But that depends on the type of cubic[3]. The counterpart of the use of an iterative solver is an unguaranteed robustness. It is to be noted that the use of `PETSC` requires a version MPI of *Code_Aster* (even if one uses one processor). Starting from a certain size of problem (a few million ddls), the use of a direct solver becomes impossible and it is essential to use an iterative solver.
In the event of failure of the resolution by an iterative solver, the levers of actions are the choice of another preconditioner (direct preconditioner single precision by defect `LDLT_SP`, incomplete preconditioner `LDLT_INC`,…), the choice of another iterative algorithm (`GMRES` by defect, `CG`, `CR`,…).
Many advices on the choice of the solver are given in [U2.08.03] “Notice of use of the linear solvers” .

The management of the total base
En cas de overflow of the limiting memory, one can exploit the filing of the results. Filing (key word factor `ARCHIVAGE` of `STAT_NON_LINE`) makes it possible to appreciably reduce the size of the bases by selecting saved times. By default, all is archived (even times resulting from the under-spanning of the time step). In the phase of installation of a study, that can prove to be useful, but too expensive in memory. The observation and the follow-up of certain quantities can meet the need for filing efficiently (see §3.4 [U2.04.01] “the Councils of use of STAT_NON_LINE”). Once the study installation, one can restrict filing with a restricted number of time steps (urgent of postprocessing for example).
To reduce the overall dimension of the data structure `result`, it is possible to choose the fields a posteriori to be archived either by indicating the fields to be preserved, or by indicating the fields to be excluded (command `EXTR_RESU`). The extraction can also be done on part of the mesh or model. Command `DETRUIRE` can also be used to destroy a concept. Following these operations of extraction or suppression, it is necessary to specify `RETASSAGE='OUI'` in command `FIN` in order to recover the disk space indeed associated with the total base.

3 structure3Un is penalizing for the direct solver, whereas a thin structure or slender gives an advantage to the direct solver

To reduce the size of the files result to format MED (`.rmed`): use `IMPR_RESU/RESTREINT`.
In an objective of a continuation, one can in certain cases replace the base by the printing of fields MED. The “continuation” will begin then with commands `DEBUT` then `LIRE_RESU`.

6 Quality control

In a general way, the total quality of a study depends on several factors. Among the causes which can deteriorate the quality of a study, one can distinguish in particular:

- uncertainties on the data,
- the choices of modelization (transition of the physique to numerical),
- the errors of discretization, once the data established and the modelization selected.

6.1 Uncertainties on the data

Sur this point uncertainties can be multiple because they can relate to each following aspect.

- Material characteristics: once the constitutive law chosen, and its parameters correctly identified on adapted tests of characterization, it can remain an uncertainty on the values of the parameters of this model, which had with the variability of the materials.
- The geometry exact of structure to be modelized can comprise uncertainties: real dimensions are not those envisaged with the design, or the degradation of the shape of a part leads to a badly controlled geometry (apart from any aspect of discretization).
- Loadings (including thermals or due to other command variables: irradiation, hydration, drying, etc.) and the boundary conditions can present many uncertainties.
- Initial conditions, in particular those due to manufacture (residual stresses, initial hardening).

For each dubious parameter evoked above, it will be sometimes necessary to carry out an analysis of sensitivity:

- that is to say the user carries out some simulations for extreme values of the dubious parameters, which is enough in the case of a monotonous variation to the result to these parameters (but it is not always the case: for example the dependence of the stresses to the extreme values of temperature, with coefficients material which are function, it is not commonplace.)
  On this subject let us announce the possibility of carrying out very simply parametric computations with *Code_Aster* (cf [U2.08.07] Distribution of parametric computations)
- that is to say in the case of highly variable parameters, it carries out an analysis mechanic-reliability engineer, with OpenTurns (http://www.openturns.org) coupled to *Code_Aster*, in the platform Salome-Meca [SV1.04.01] Administrative of chained study Openturns/Aster).

6.2 Choice of modelization and checking of the Même

data if the data are reliable, the quality of the results also depends on choices carried out by the user relating to simulation to carry out. The paragraphs precedent describe these choices. Let us point out some general advices:

- Beyond the mesh, it can be useful to check the data of computation, such as for example, to visualize the boundary conditions and loadings (meaning and place of application), the assignments of the materials, and the elementary characteristics (directional sense of the beams, thickness, sections). For that, use `IMPR_RESU/CONCEPT`.

- Other checks can be carried out directly in the command file; for example, to check the coherence of the system of units, the boundary conditions, the elementary characteristics, it is always useful to carry out a first elastic design, before any nonlinear study.

- To check the material characteristics, it can be useful to carry out with the setting in data of *Code_Aster* a test simpler than the study, for example on material point (`SIMU_POINT_MAT`). Is the curve of traction and compression found?

- The computation of the mass (or the volume) of structure also forms part of the simple but sometimes useful checks.

6.3 To control the errors due to the spatial discretization

Sur this point, we point out some general information here, but the reading of [U2.08.01] Utilisation of the indicators of error and associated strategies of mesh adaptation more than is advised.

Independently of any desire to carry out mesh adaptation, we advise to check the initial mesh with command `MACR_INFO_MAIL` [U7.03.02] Macro-command `MACR_INFO_MAIL`. This command makes it possible to realize, with few expenses, the following checks:

- check the agreement of the mesh with the initial geometry (in dimension, volume);
- list `THE GROUP_MA` and `GROUP_NO`, for a good modelization of the boundary conditions;
- diagnose possible problems (connexity, holes, interpenetration of meshes);
- provide quality standards of the meshes (evaluated mesh by mesh).

To more easily control the effect of the quality of the mesh, an automatic strategy of mesh adaptation is operational: it lean on the software of Homard mesh adaptation, which can be called directly since the command file of *Code_Aster* or in the Salome-Meca platform, and uses either of the indicators of error making it possible to control the adaptation, or the values of a field, or the jump of a field from one element to another. Several motivations appear to adapt a mesh:

- The mesh is very intricate to realize: one starts from a simple version and one entrusts to an automatic process the responsibility to refine it of.
- One wants to make sure of the convergence of the numerical solution: rather than to realize with the hand of the increasingly fine meshes, one lets the software seek itself the places where it would be necessary to refine the mesh to increase the accuracy of the result.
- The conditions of computation change during its unfolding: the areas which must be with a grid finely move. If one nets fine everywhere as of the beginning, the mesh is too large. While adapting progressively, (refinement - coarsening) the mesh will be fine only at the places necessary: its size will be reduced and the quality of the solution will be good.

The areas to be refined can be identified:

- Maybe with an indicator of error. Let us note that simplest (ZZ1, of Zhu-Zienkiewicz type) are very robust, available in 2D and 3D, and even if they do not provide best raising an error, their variations are enough to control the adaptation. The estimators in quantity of interest are more relevant to provide a limit of the made error.
- Maybe with a field (strain, intern variable) relevant. Attention, in plasticity, the von Mises stress is not it always. One can use:
  - either values extreme of this field (for example ask to refine the 5% of elements which has the values strongest),
  - or control the adaptation by the jump of the values of this field with the border between two finite elements.
- Maybe by boxes (uniform refinement in geometrical areas).

For more information, to see [U2.08.01] Utilisation of the indicators of error and strategies of mesh adaptation associated

6.4 Erreurs with discretization in Pour

time most nonlinearities (behavior, contact, large deformation) the temporal discretization influences the result. Results of convergence exist (extremely fortunately) in all the conventional cases (élasto-visco-plasticity, contact), but it remains nevertheless to be made sure that for a selected time step the solution is sufficiently close to the continuous solution in time.

The simplest solution is similar to uniform refinement for the mesh adaptation: it consists to uniformly refine the time step on all the transient and to start again `STAT_NON_LINE`. This can be automated thanks to a functionality of `DEFI_LIST_INST` (see [U2.04.01] and [U4.34.03]).

Another criterion of control of the time step is the subdivision according to a quantity of interest: `DEFI_LIST_INST / DELTA_GRANDEUR` allows redécouper indeed the time step if the maximum variation of a given quantity (for example a component of plastic strain) is higher than a provided threshold.

Moreover, for the elastoplastic behaviors, it is possible to consider directly the error due to the discretization in time, in a way similar to `RESI_RADI_RELA` (cf.3.2.48). If this criterion were not taken into account during computation, it is possible of the compute in postprocessing, in `CALC_CHAMP` : component `ERR_RADI` of option `DERA_ELGA` at every moment contains the estimate of error and in each point of integration.

7 References

[1] “Structural mechanics”, p.658, F.Voldoire; Y.Bamberger, Presses of the ENPC 2008.