

Operator DEFI_LIST_INST

1 Goal

To define the list of moments of calculation, like its management for the iterative algorithms of resolution (orders `STAT_NON_LINE` and `DYNA_NON_LINE`):

- possibilities of recutting of the step of time in the event of failure,
- adaptability of the step of time.

Product a structure of data `list_inst`.

Contents

1 Goal.....	1
2 Syntax.....	3
3 Operands.....	6
3.1 Keyword DEFI_LIST.....	6
3.1.1 Operand METHOD.....	6
3.1.2 Operands LIST_INST and VALE.....	6
3.1.3 Operands PAS_MINI, PAS_MAXI, NB_PAS_MAXI.....	6
3.2 Keyword FAILURE.....	7
3.2.1 Definition of the causes of failure.....	7
3.2.1.1 Event 'ERROR'.....	8
3.2.1.2 Event 'DIVE_RESI'.....	8
3.2.1.3 Event 'DELTA_Grandeur'.....	8
3.2.1.4 Event 'COLLISION'.....	9
3.2.1.5 Event 'INTERPENETRATION'.....	9
3.2.1.6 Event 'INSTABILITY'.....	9
3.2.2 Definition of the actions to be carried out in the event of failure.....	10
3.2.2.1 Action 'STOP'.....	10
3.2.2.2 Action 'CUTTING'.....	10
3.2.2.3 Action 'AUTRE_PILOTAGE'.....	13
3.2.2.4 Action 'ITER_SUPPL'.....	13
3.2.2.5 Action 'ADAPT_COEF_PENA'.....	14
3.2.2.6 Action 'CONTINUES'.....	14
3.3 Keyword ADAPTATION.....	14
3.3.1 Operand EVENT.....	15
3.3.2 Operands NB_INCR_SEUIL, NOM_PARA, CRIT_COMP, VALE_I.....	15
3.3.3 Operand MODE_CALCUL_TPLUS.....	15
3.3.4 Operand PCENT_AUGM.....	16
3.3.5 Operands VALE_REF, NOM_CHAM, NOM_CMP.....	16
3.3.6 Operand NB_ITER_NEWTON_REF.....	16
3.4 Operand INFORMATION.....	16

2 Syntax

```
deflist [list_inst] = DEFI_LIST_INST(
```

1) words- keys for the definition a priori of the list of moment

```
CHALLENGE_LISTE = _F (
    ◆ METHOD = / 'MANUAL' , [DEFECT]
                / 'CAR' ,
    ◆ / LIST_INST = list, [l_r8]
      / VALE = vale, [R]
      / ◆ RESULT = resu, [result]
        ◆ SUBD_PAS = npas [I]

    # if METHOD = 'CAR'
        ◆ PAS_MINI = pasmin, [R]
        ◆ PAS_MAXI = pasmax, [R]
        ◆ NB_PAS_MAXI = / 1000000 [DEFECT]
                        / Nbpasmax, [I]
    )
```

2) words- keys for the behavior in the event of failure

```
◆ FAILURE = _F (
    ◆ EVENT = / 'ERROR' , [DEFECT]
                / 'DIVE_RESI' ,
                / 'DELTA_Grandeur'
                / 'COLLISION'
                / 'INTERPENETRATION'
                / 'INSTABILITY'

    # If EVENELIES = 'DELTA_Grandeur'
    ◆ VALE_REF = valref, [R]
    ◆ NOM_CHAM = / 'DEPL' ,
                / 'SIEF_ELGA' ,
                / 'VARI_ELGA' ,
    ◆ NOM_CMP = cmp , [TXT]
    ◆ ACTION = / 'CUTTING' , [DEFECT]
                / 'STOP' ,

    # If ACTION = 'CUTTING'
    ◆ SUBD_METHODE = / 'MANUAL' , [DEFECT]
                    / 'CAR' ,
        # If SUBD_METHODE = 'MANUAL'
        ◆ SUBD_PAS = 4. , [DEFECT]
                    not, [R]
        ◆ SUBD_NIVEAU = 3 , [DEFECT]
                    niv, [I]
        ◆ SUBD_PAS_MINI = 0. , [DEFECT]
                    pasmin, [R]
        # If SUBD_METHODE = 'CAR'
        ◆ SUBD_PAS_MINI = 0 , [DEFECT]
                    pasmin, [R]

    # If EVENELIES = 'ERROR'
    ◆ ACTION = / 'CUTTING' , [DEFECT]
                / 'STOP' ,
```

```
        / 'ITER_SUPPL' ,
        / 'AUTRE_PILOTAGE' ,

# If ACTION = 'CUTTING' / 'AUTRE_PILOTAGE'
◇ SUBD_METHODE      = / 'MANUAL' ,           [DEFECT]
                        / 'CAR' ,
# If SUBD_METHODE = 'MANUAL'
◇ SUBD_PAS          = 4 . ,                 [DEFECT]
                        not ,                [R]
◇ SUBD_NIVEAU       = 3 ,                   [DEFECT]
                        niv ,                [I]
◇ SUBD_PAS_MINI     = 0 . ,                 [DEFECT]
                        pasmin ,            [R]
# If SUBD_METHODE = 'CAR '
◇ SUBD_PAS_MINI     = 0 ,                   [DEFECT]
                        pasmin ,            [R]

# If ACTION = 'ITER_SUPPL '
◇ PCENT_ITER_PLUS   = 50 ,                   [DEFECT]
                        pcent ,            [I]
◇ SUBD_METHODE      = / 'MANUAL' ,           [DEFECT]
                        / 'CAR' ,
# If SUBD_METHODE = 'MANUAL'
◇ SUBD_PAS          = 4 . ,                 [DEFECT]
                        not ,                [R]
◇ SUBD_NIVEAU       = 3 ,                   [DEFECT]
                        niv ,                [I]
◇ SUBD_PAS_MINI     = 0 . ,                 [DEFECT]
                        pasmin ,            [R]
# If SUBD_METHODE = 'CAR '
◇ SUBD_PAS_MINI     = 0 ,                   [DEFECT]
                        pasmin ,            [R]

# If EVENELIES = 'COLLISION '
◇ ACTION            = / 'CUTTING' ,         [DEFECT]
                        / 'STOP' ,
# If ACTION = 'CUTTING'
◇ SUBD_METHODE      = / 'CAR' ,             [DEFECT]
                        / 'MANUAL' ,
# If SUBD_METHODE = 'MANUAL'
◇ SUBD_PAS          = 4 . ,                 [DEFECT]
                        not ,                [R]
◇ SUBD_NIVEAU       = 3 ,                   [DEFECT]
                        niv ,                [I]
◇ SUBD_PAS_MINI     = 0 . ,                 [DEFECT]
                        pasmin ,            [R]
# If SUBD_METHODE = 'CAR '
    ◆ SUBD_INST      = subd_inst ,          [R]
    ◆ SUBD_DURATION  = SUBD_DUREE ,        [R]

# If EVENELIES = 'INTERPENETRATION '
    ◆ PENE_MAXI      = penmax ,             [R]
◇ ACTION            = / 'ADAPT_COEF_PENA ' , [DEFECT]
                        / 'STOP' ,
# If ACTION = 'ADAPT_COEF_PENA '
◇ COEF_MAXI         = 1E12                  [DEFECT]
```

```

                                coemax ,                                [R]

# If EVENELIES = ` DIVE_RESI `                                [R]
◇ ACTION = / `CUTTING ` ,                                [DEFECT]
◇ SUBD_METHODE = / `MANUAL` ,                                [DEFECT]
                                / `CAR` ,

# If SUBD_METHODE = `MANUAL`
◇ SUBD_PAS = 4. ,                                [DEFECT]
                                not ,                                [R]
◇ SUBD_NIVEAU = 3 ,                                [DEFECT]
                                niv ,                                [I]
◇ SUBD_PAS_MINI = 0. ,                                [DEFECT]
                                pasmin ,                                [R]

# If SUBD_METHODE = ` CAR `
◇ SUBD_PAS_MINI = 0 ,                                [DEFECT]
                                pasmin ,                                [R]

# If EVENELIES = `INSTABILITY`
◇ ACTION = / `CONTINUES` ,                                [DEFECT]
                                / `STOP` ,

# 3) words- keys for the adaptation (if METHOD = `CAR`)

◇ ADAPTATION = _F (

◇ EVENT = / `THRESHOLD` ,                                [DEFECT]
                                / `TOUT_INST`
                                / `NO`

# if EVENT = `THRESHOLD`
◇ NB_INCR_SEUIL = /2 ,                                [DEFECT]
                                /nbincseuil ,                                [I]
◇ NOM_PARA = `NB_ITER_NEWTON`                                [DEFECT]
◇ CRIT_COMP = / `IT` ,                                [DEFECT]
                                = / `LT` ,
                                = / `GE` ,
                                = / `WP` ,
◇ VALE_I = nbvalseuil ,                                [I]
◇ MODE_CALCUL_TPLUS = / `FIXES` ,                                [DEFECT]
                                / `DELTA_Grandeur`
                                / `ITER_NEWTON`
                                / `IMPLEX`

# if MODE_CALCUL_TPLUS = `FIXES`
◇ PCENT_AUGM = /100. ,                                [DEFECT]
                                / pcent ,                                [R]

# if MODE_CALCUL_TPLUS = `DELTA_Grandeur`
◆ VALE_REF = valref ,                                [R]
◆ NOM_CHAM = / `DEPL` ,
                                / `SIEF_ELGA` ,
                                / `VARI_ELGA` ,
◆ NOM_CMP = cmp ,                                [TXT]

# if MODE_CALCUL_TPLUS = `ITER_NEWTON`
◆ NB_ITER_NEWTON_REF = nitref ,                                [I]

)

```

```
◇ INFORMATION      = / 1,          [DEFECT]
                    / 2,          [I]
)

```

3 Operands

3.1 Keyword DEFI_LIST

3.1.1 Operand METHOD

```
◇ METHOD      = / 'MANUEL',          [DEFECT]
               / 'CAR',

```

Cand operand makes it possible to choose the way of managing of the list of moments.

By default, management is manual (METHODE=' MANUEL'), which means that one will traverse the list of moments exactly defined by the user (see §3.1.2). In the event of under-cutting, new moments can be inserted.

In automatic management (METHODE=' AUTO'), the code automatically calculates the new moments of calculation. The ways of calculating of the new steps of time are defined under the keyword ADAPTATION. Moments of calculations defined by the user (see §3.1.2) are however respected.

3.1.2 Operands LIST_INST and VALE

```
◇ / LIST_INST      = list,          [l_r8]
  / VALE           = vale,          [R]
  / ◇ RESULT       = resu,          [result]
  / ◇ SUBD_PAS     = npas           [I]

```

The moments of calculation can be defined in 3 different ways:

- maybe by informing the keyword LIST_INST with a concept defined as a preliminary by the operator DEFI_LIST_REEL [U4.34.01],
- maybe by informing the keyword VALE with a list of realities,
- maybe by informing the keyword RESULT with a concept produced by STAT_NON_LINE or DYNA_NON_LINE. This keyword makes it possible to refine an already existing list of moments. Indeed, one defines a list of moments then npas **finer time** that the list of the moments really calculated in STAT_NON_LINE or DYNA_NON_LINE. It should be noted that this list of really calculated moments can be different from the list of the moments given as starter from STAT_NON_LINE or DYNA_NON_LINE (for example because of under-cutting) and of the list of the filed moments (if one does not file every moment). This functionality is useful to check the independence of the results to the temporal discretization. For example, to obtain a list of moments 2 times finer, one choose SUBD_PAS=2.

Caution! If the SD result resu comes from a calculation with recovery, it is not possible to re-use it in the order because one cannot extract from list of single moments (certain moments will be in double).

3.1.3 Operands PAS_MINI, PAS_MAXI, NB_PAS_MAXI

```
◇ PAS_MINI      = pasmin,          [R]
◇ PAS_MAXI      = pasmax,          [R]
◇ NB_PAS_MAXI   = Nbpasmax,        [I]

```

These operands are to be informed only in the event of automatic management of the list of the moments (METHOD = ' AUTO').

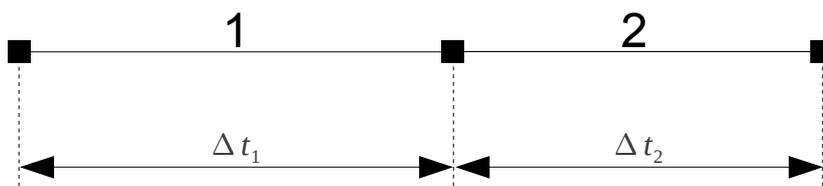


Figure 3-1: Automatic adaptation of the step of time

The operand `PAS_MINI` allows to specify the step of minimal time. So during the adaptation of the step of time, one is brought to treat a step of time lower than `pasmin`, then calculation stops. If this operand is not indicated, in the case general `PAS_MINI` is worth then 10^{-12} (it is the authorized minimal value besides); if it is not indicated and that `MODE_CALCUL_TPLUS='IMPLEX'`, then it `PAS_MINI` 1/1000ème first step of calculation (inevitably entered by the user is worth *via* `LIST_INST`).

$$\text{If } \Delta t_2 < \text{PAS_MINI} : \rightarrow \text{STOP}$$

Attention, the checking of `PAS_MINI` is not **not** carried out during the possible under-cutting of the step of time in the event of failure (it is `SUBD_PAS_MINI` who acts during under-cutting). The operand `PAS_MINI` does not act **that** for the calculation of a new step of time.

The operand `PAS_MAXI` allows to specify the step of maximum time, but its operation is different from that of `PAS_MINI`. Indeed, so during the adaptation of the step of time, one is brought to to treat a step of time higher than `pasmax`, then calculation does not stop (contrary to the operation of `PAS_MINI`), but the step of time is then limited to `pasmax`. If this operand is not indicated, this stage will not be made; if it is not indicated and that `MODE_CALCUL_TPLUS='IMPLEX'`, then it `PAS_MAXI` 10 times the first step of calculation (inevitably entered by the user is worth *via* `LIST_INST`).

$$\text{If } \Delta t_2 > \text{PAS_MAXI} : \rightarrow \Delta t_2 = \text{PAS_MAXI}$$

The operand `NB_PAS_MAXI` allows to specify the number of steps of time beyond whose calculation stops. The value by default is a million (it is also the authorized maximum value).

3.2 Keyword FAILURE

This keyword factor répétable makes it possible to manage the list of moments in the event of failure of convergence. With each occurrence of the keyword `FAILURE`, one defines a cause of failure (see § 3.2.1) and an action to be carried out if this cause is satisfied (see § 3.2.2). The actions can be different between each occurrence from the keyword `FAILURE`.

If several causes of failure are satisfied simultaneously, the tripping order of these failures is predetermined:

`ERROR' is priority on `DELTA_Grandeur'

3.2.1 Definition of the causes of failure

```

◇ EVENT = / `ERROR' , [DEFECT]
          / `DIVE_RESI'
          / `DELTA_Grandeur'
          / `COLLISION'
          / `INTERPENETRATION'
          / `INSTABILITY'
    
```

The operand `EVENT` allows to define the causes of failure.

3.2.1.1 Event 'ERROR'

By default, the cause of failure is `ERROR`. This cause gathers the following reasons:

- Going beyond the authorized maximum number of iterations of Newton.
- Failure during the local integration of the law of behavior. The local parameters of the algorithm used for the integration of the law of behavior are in [U4.51.11].
- Failure of the algorithm of Deborst for the plane constraints or the models 1D. One will find more details in [U4.51.11].
- Failure at the time of the resolution of the equation of piloting (see [U4.51.03]).
- Failure at the time of the resolution of the problem of discrete contact (see [R5.03.50]). There is failure when the maximum number of iterations of contact is exceeded or when the matrix of contact is singular.
- The matrix of the system is singular. The detection of singularity is based on the parameter `NPREC` keyword factor `SOLVEUR` (see [U4.50.01]).
- The iterative linear solver reached the maximum number of iterations of Krylov fixed by `NMAX_ITER` in the keyword factor `SOLVEUR` (see [U4.50.01]).
- With convergence, physical criteria are not satisfied during integration with the law of behavior (concept of "vent-driven")

Even if one finds no occurrence of `FAILURE/EVENT = 'ERROR'`, the code adds one with the current parameters automatically of them by default (i.e. the action 'CUTTING' in mode 'MANUAL' and with `SUBD_NIVEAU=4`).

3.2.1.2 Event 'DIVE_RESI'

The choice of the événement 'DIVE_RESI' allows to specify another possible cause of failure ("vent-driven"). There will be failure if the residue does not decrease sufficiently during iterations of Newton. More precisely, there will be failure if: $\min[R(i), R(i-1)] > R(i-2)$ for $i \geq 3$, where $R(j)$ is the residue (`RESI_GLOB_MAXI`) with the iteration of Newton j . If there is failure, then the step of time under-is cut out. The interest this functionality is that it is not necessary to wait until one arrives at the end of the iterations of Newton authorized Re-to cut out the step of time in the event of "notable" not-convergence. One realizes of not-convergence more quickly, a time-saver is thus hoped.

Note:

The activation of this functionality with the discrete contact is not advised because in this kind of calculation the residue does not decrease in general in a monotonous way.

3.2.1.3 Event 'DELTA_Grandeur'

The choice of the événement 'DELTA_Grandeur' allows to specify another possible cause of failure ("vent-driven"). There will be failure if the increment of a component of a field exceeds a fixed threshold, **with convergence**. It is then necessary to define the name of the field by the operand `NOM_CHAM` (field of displacement, constraints or internal variables), the name of the component by the operand `NOM_CMP` and the value of the threshold of reference by the operand `VALE_REF`.

```
# if EVENT = 'DELTA_Grandeur'
  ◆ VALE_REF = valref , [R]
  ◆ NOM_CHAM = / 'DEPL' ,
                / 'SIEF_ELGA' ,
                / 'VARI_ELGA' ,
  ◆ NOM_CMP = cmp , [TXT]
```

More precisely, are v the component `NOM_CMP` field `NOM_CHAM` and $valref$ the value of reference `VALE_REF`.

If the field `NOM_CHAM` is a field with the nodes and N_{no} is the whole of all the nodes of the grid, then there will be failure so with convergence:

$$\max(\Delta|v^i|, \forall i \in N_{no}) > valref$$

If the field `NOM_CHAM` is a field at the points of Gauss, $N_{sp, pg, el}$ is the whole of all the under-points of all the points of Gauss of all the elements of the model, then there will be failure so with convergence:

$$\max(\Delta|v^i|, \forall i \in N_{sp, pg, el}) > valref$$

The increment Δ is to be taken with the direction increment between two moments.

N°1 example with a law `VMIS_ISOT_LINE` :

```
ECHEC= _F ( EVENT      = 'DELTA_GRANDEUR',
           VALE_REF    = 0.1e-2,
           NOM_CHAM    = 'VARI_ELGA',
           NOM_CMP     = 'V1' ),
```

One will have failure if the increment of the plasticity cumulated between two steps of time exceeds 0.1%.

N°2 example:

```
ECHEC= (
  _F ( EVENT      = 'DELTA_GRANDEUR',
      VALE_REF    = 5.e-2,
      NOM_CHAM    = 'DEPL',
      NOM_CMP     = 'DX' ),
  _F ( EVENT      = 'DELTA_GRANDEUR',
      VALE_REF    = 5.e-2,
      NOM_CHAM    = 'DEPL',
      NOM_CMP     = 'DY' ),
)
```

One will have failure if the standard (standard sup) of displacement (in 2D) varies of more than 5th-2 between two steps of time.

Note: in the two preceding examples, although `EVENEMENT='ERREUR'` is not present, it will be nevertheless activated into hard.

3.2.1.4 Event 'COLLISION'

This event makes it possible to cut out the step of time if the contact is detected (impact). It is a particularly useful functionality in dynamics. Indeed, in this case, an unsuited temporal discretization will reveal inopportune oscillations of displacement and speed at the time of the shock. To minimize these digital artifacts, one in general recommends to finely cut out the step of time at the time of the collision, which is not possible that for the typical cases where this moment is detectable *a priori* by the user. In the case general, one can use this functionality so that cutting is done automatically.

3.2.1.5 Event 'INTERPENETRATION'

◆ `PENE_MAXI = penmax , [R]`

This event detects the interpenetration of a node in a surface when the unilateral contact of regularized type (penalization) is activated. The user gives a value of the maximum penetration which he wants to obtain (parameter `PENE_MAXI`) and codes it adapt then the coefficient of penalization (`ACTION='ADAPT_COEF_PENA'`) or stops (`ACTION='ARRET'`) to respect this constraint. This functionality is available only for the contact in `FORMULATION='DISCRETE'` (see `DEFI_CONTACT [U4.44.11]`).

3.2.1.6 Event 'INSTABILITY'

This event makes it possible to manage the course of a non-linear calculation when an instability is detected at the time of the call to CRIT_STAB (cf. [U4.51.03] and [U4.53.01]). If this keyword CRIT_STAB this event is not used then cannot occur.

If an instability is detected, an action will then be started, according to two possibilities:

- ACTION=' CONTINUE' : it is the value by default and one will continue calculation, even in the event of instability,
- ACTION=' ARRET' : if an instability is collected, then one will adopt the non-linear resolution properly (the base will be saved and reusable in continuation).

More details on the criterion of instability and its use are given in documentations [R7.05.01] and [U2.08.04].

3.2.2 Definition of the actions to be carried out in the event of failure

```

◇ ACTION = / 'CUTTING' , [DEFECT]
            / 'STOP' ,
            / 'ITER_SUPPL' ,
            / 'AUTRE_PILOTAGE' ,
            / 'ADAPT_COEF_PENA'
            / 'CONTINUES'
    
```

According to the event, only certain actions are possible, see table below:

ACTION →	STOP	CUTTING	ITER_SUPPL	AUTRE_PILOTAGE	ADAPT_COEF_PENA	CONTINUOUS
EVENT ↓						
ERROR	YES	YES	YES	YES	NOT	NOT
DIVE_RESI	NOT	YES	NOT	NOT	NOT	NOT
DELTA_GRADEUR	YES	YES	NOT	NOT	NOT	NOT
COLLISION	YES	YES	NOT	NOT	NOT	NOT
INTERPENETRATION	YES	NOT	NOT	NOT	YES	NOT
INSTABILITY	YES	NOT	NOT	NOT	NOT	YES

For example, in the case more the general (EVENEMENT=' ERREUR'), there are five possibilities:

- To cut out the step of time (ACTION=' DECOUPE');
- To authorize additional iterations of Newton (ACTION=' ITER_SUPPL');
- To change the solution resulting from piloting (ACTION=' AUTRE_PILOTAGE');
- To stop (ACTION=' ARRET').

3.2.2.1 Action 'STOP'

If one chooses ACTION = 'STOP', calculation stops properly: the base is saved and will be then reusable in continuation.

3.2.2.2 Action 'CUTTING'

```

◇ SUBD_METHODE = / 'MANUAL'
                  / 'CAR'
# If ACTION = 'CUTTING' AND IF SUBD_METHODE = 'CAR'
  ◆ SUBD_INST = subd_inst, [R]
  ◆ SUBD_DURATION = SUBD_DUREE, [R]
# If SUBD_METHODE = 'MANUAL'
  ◇ SUBD_PAS = 4. , [DEFECT]
              not, [R]
  ◇ SUBD_NIVEAU = 3 , [DEFECT]
                 niv, [I]
  ◇ SUBD_PAS_MINI = 0. , [DEFECT]
    
```

```

pasmin, [R]
# If SUBD_METHODE = 'CAR'
  ◊ SUBD_PAS_MINI = 0. , [DEFECT]
pasmin, [R]
    
```

This action consists in remarking the step of time by cutting out it. There exist two methods of cutting of the step of time: mode CAR and mode HANDBOOK.

Cutting in mode HANDBOOK

In mode HANDBOOK, the step of time is cut out in a uniform way. It is thus necessary to specify the number of cuttings using the keyword SUBD_PAS.

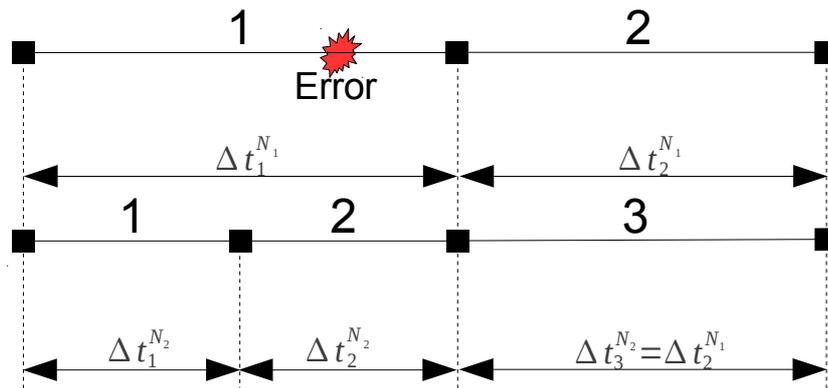


Figure 3-2: Cutting in mode MANUEL with SUBD_PAS=2

One sees an example on the figure 3-2. The event ERROR starts during the step of time 1, initially length $\Delta t_1^{N_1}$. This step is then cut out in two steps (SUBD_PAS = 2) equal lengths such as $\Delta t_1^{N_2} = \Delta t_2^{N_2} = \frac{\Delta t_1^{N_1}}{2}$. The following step (number 2 in the initial configuration) becomes step 3 then, its length remains unchanged $\Delta t_3^{N_2} = \Delta t_2^{N_1}$.

Not	1	2	3
Initial - 1	$\Delta t_1^{N_1}$	$\Delta t_2^{N_1}$	
Cutting - 2	$\Delta t_1^{N_2}$	$\Delta t_2^{N_2}$	$\Delta t_3^{N_2}$

Concept of level of cutting

The cutting of the step of time is recursive: if the event ERROR starts during a step already cut out, one redécoupe. One sees an example on the figure 3-3.

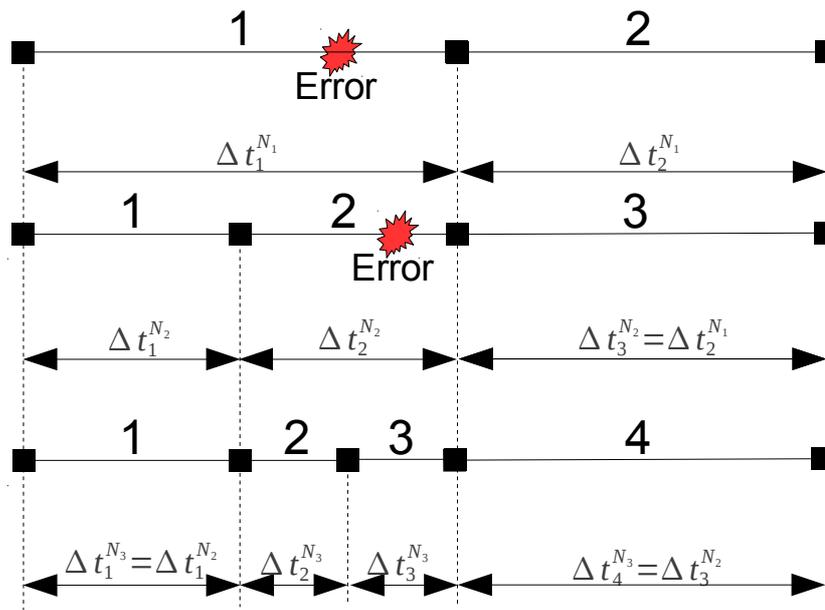


Figure 3-4: Recursive cutting in MANUAL mode with SUBD_PAS=2

After the first cutting, the failure occurs in (new) the second step of time. One redécoupe this one with $\Delta t_2^{N_3} = \Delta t_3^{N_3} = \frac{\Delta t_2^{N_2}}{2}$. The step of next time remains unchanged $\Delta t_4^{N_3} = \Delta t_3^{N_2}$, the first step of time is converged and thus remains also unchanged $\Delta t_1^{N_3} = \Delta t_1^{N_2}$. What gives:

Not	1	2	3	4
Initial – 1	$\Delta t_1^{N_1}$	$\Delta t_2^{N_1}$		
Cutting – 2	$\Delta t_1^{N_2}$	$\Delta t_2^{N_2}$	$\Delta t_3^{N_2}$	
Cutting – 3	$\Delta t_1^{N_3}$	$\Delta t_2^{N_3}$	$\Delta t_3^{N_3}$	$\Delta t_4^{N_3}$

There cannot be a variation of more than one level between two steps of time.

Cutting in mode CAR

The automatic mode of cutting uses two possibilities:

- If the event release is a collision (EVENEMENT=' COLLISION'), then the step of time is cut out while having for objective to be a step of time length `subd_inst`, maintained for the length of time `subd_duration`. If the moment of cutting is too close to the collision ("shaving" contact too), cutting will not be effective, it is then necessary to change the initial discretization or the parameter `subd_inst` (if the contact becomes shaving after cutting);
- If not, one tries a linear extrapolation of the value of the residues.

If one notes *iter* the iteration of Newton and $resi^{ext}$ the value of the extrapolated residue (RESI_GLOB_RELA and/or RESI_GLOB_MAXI), then one seeks a_1 , a_2 and b such as:

$$resi^{ext} = \frac{(a_1 \cdot iter + a_2)}{b} \tag{1}$$

Here the algorithm used:

```
For iter = deb, fin
    xx = log(resi_iter+1)
```

```
If iter > fin - 3
    poids = 2
If not
    poids = 1
One calculates
    xn = xn + poids
    sx = sx + poids.xx
    sy = sy + poids.iter
    sxx = sxx + poids.(xx2)
    syx = syx + poids.xx.iter
Fin Pour
    a1 = sxx.sy - sx.syx
    a2 = -(sx.syx) + syx.xn
    b = -sx2 + sxx.xn
```

To make this extrapolation, one at least asks for three iterations of Newton and one considers that the two last iterations have twice more weight than the first.

Control of cutting

If convergence is difficult and that the event starts several times of continuation, involving the recursive step division of time, one can stop the process of two manners in mode `HANDBOOK` :

- With the keyword `SUBD_NIVEAU`, cutting will stop as soon as cutting is made more `niv` time.
- With the keyword `SUBD_PAS_MINI`, cutting will stop as soon as the step of time is lower than `pasmin`.

In mode `CAR`, only the keyword `SUBD_PAS_MINI` is available.

Notice concerning the calculation of buckling:

During elastoplastic calculation of buckling, it can happen that the tangent matrix of the system is singular during iterations of Newton. While Re-cutting out the step of time, one can pass these hard points.

Characteristics of the operand `SUBD_NIVEAU` :

- The operand `SUBD_NIVEAU` is a concept which should be total with all the occurrences of keyword `FAILURE`. Thus, this operand should be defined only once. In practice, it is not the case and nothing prevents from defining a value of `SUBD_NIVEAU` different with each occurrence from the keyword `FAILURE`. To secure this case there, the algorithm will retain with final like global value and single of `SUBD_NIVEAU` **maximum** `SUBD_NIVEAU` defined by the user.

3.2.2.3 Action 'AUTRE_PILOTAGE'

The action 'AUTRE_PILOTAGE' allows to define an action in a calculation which calls on the piloting of the loading. During piloting, one can have to choose between two solutions (see documentation [R5.03.80]) which one often discriminates using a criterion `RESIDUE`.

If the user selects the action 'AUTRE_PILOTAGE' and that convergence fails with the choice of solution which minimizes the residue, one start again the step of current time, after having reset it correctly, but without subdividing. With the first iteration where the choice arises, one will choose the solution of maximum residue at the time of this second attempt. One subdivides only if the two attempts failed.

This action will thus be undertaken only if the maximum number of iterations of Newton is reached and if it is the first attempt.

3.2.2.4 Action 'ITER_SUPPL'

```

◇ PCENT_ITER_PLUS = 50 , [DEFECT]
                    PCENT , [I]
◇ SUBD_METHODE = / 'MANUAL' , [DEFECT]
                / 'CAR' ,
    
```

The action 'ITER_SUPPL' allows to make iterations of Newton beyond the maximum given by ITER_GLOB_MAXI (or ITER_GLOB_ELAS). It prevents that calculation fails for some iterations of Newton. For that, thanks to the linear extrapolation of the residues (see § 3.2.2.2), it is possible to estimate the iteration count of Newton necessary to convergence.

The iteration count additional is limited by the keyword PCENT_ITER_PLUS. If the estimated number of iterations is higher than pcent*ITER_GLOB_*, then it is considered that one failed. In the same way if extrapolation does not succeed. In these two cases, it is possible to activate the cutting of the step of time (keyword SUBD_METHODE).

3.2.2.5 Action 'ADAPT_COEF_PENA'

```

◇ COEF_MAXI = 1E12 [DEFECT]
             coemax, [R]
    
```

If the event INTERPENETRATION starts, this action consists in automatically increasing the value of the coefficient of penalization (see operator DEFILIST_CONTACT [U4.44.11]) so as to decrease this interpenetration. The parameter COEF_MAXI allows to specify the maximum value taken by the coefficient of penalization when the automatic adaptation mechanism is used, this in order to avoid problems of conditioning of the matrix (see the precautions of use of the methods penalized in [U2.04.04]).

Currently the mechanism does nothing but increase this coefficient and never makes it decrease, even if the interpenetration does not appear any more after adaptation, it thus tends to transform a problem of contact with interpenetration into problem without contact. It is thus advisable to choose a coefficient of penalization initial weaker than necessary (see operator DEFILIST_CONTACT [U4.44.11]).

3.2.2.6 Action 'CONTINUES'

If the event INSTABILITY starts, this action consists in trying to continue non-linear calculation. It is the value by default for this event, which is the only one with being able to combine with this action.

3.3 Keyword ADAPTATION

The adaptation of the step of time consists in calculating a new step of time Δt^{i+1} , starting from information of the step of current time Δt^i (and of the steps of previous times Δt^{i-1} , Δt^{i-2} , ...).

The goal is thus to calculate $\Delta t^{i+1} = c \cdot \Delta t^i$, where c is a real coefficient.

The process is the following:

For the occurrence k keyword ADAPTATION, one defines an event and a way of calculating of Δt^{i+1} . If the event is checked, then one calculates c^k with the selected way of calculating.

A summary table of this type is obtained:

Occurrence n°1	$c^1 = 4,2$
Occurrence n°2	event not checked
Occurrence n°3	$c^3 = 1,7$
Occurrence n°4	$c^4 = 3,9$

Then, one chooses $c = \min(c^k)$ among the occurrences whose event is checked. In this example, the new step of time will be worth $\Delta t^{i+1} = 1,7 \cdot \Delta t^i$.

To use `MODE_CALCUL_TPLUS='IMPLEX'`, it is necessary that this keyword is used only once.

3.3.1 Operand EVENT

```
◇ EVENT = / 'THRESHOLD', [DEFECT]
          / 'TOUT_INST'
          / 'NO'
```

This operand makes it possible to specify the triggering function for the adaptation of the step of time.

If `EVENEMENT='SEUIL'`, then the event is checked only if the threshold is crossed.

If `EVENEMENT='TOUT_INST'`, then the event is checked at every moment; this case is obligatory with the method `IMPLEX`.

If `EVENEMENT='AUCUN'`, then the event is never checked.

3.3.2 Operands NB_INCR_SEUIL, NOM_PARA, CRIT_COMP, VALE_I

```
◇ NB_INCR_SEUIL = /2, [DEFECT]
                  /nbincseuil, [I]
◇ NOM_PARA = 'NB_ITER_NEWTON' [DEFECT]
◇ CRIT_COMP = / 'IT', [DEFECT]
              = / 'LT',
              = / 'GE',
              = / 'WP',
◇ VALE_I = nbvalseuil, [I]
```

These operands are to be informed only in the event of event of type threshold (`EVENEMENT='THRESHOLD'`). The event is checked if one has `nbincseuil` time of continuation a “green light”, a green light being characterized by the condition `NOM_PARA CRIT_COMP VALE_I`.

For example if `NB_INCR_SEUIL=2`, `NOM_PARA='NB_ITER_NEWTON'`, `CRIT_COMP='LE'` and `VALE_I=5`, then the event is checked if one made 2 times of continuation less than 5 iterations of Newton.

Note:

There is no value by default for `VALE_I` in the catalogue of the operator `DEFI_LIST_INST`. However, if its value is not indicated, one will take for `VALE_I` half of the maximum number of iterations of Newton (declared in `STAT_NON_LINE/CONVERGENCE/ITER_GLOB_MAXI`).

3.3.3 Operand MODE_CALCUL_TPLUS

```
◇ MODE_CALCUL_TPLUS = / 'FIXES', [DEFECT]
                     / 'DELTA_GRANDEUR'
                     / 'ITER_NEWTON'
                     / 'IMPLEX'
```

This operand is used to specify the way of calculating of the new step of time (in fact, of the coefficient c).

If `MODE_CALCUL_TPLUS='FIXE'`, then the coefficient c is fixed. Its value is given thanks to the operand `PCENT_AUGM`. The new step of time will be worth then $\Delta t^{i+1} = \left(1 + \frac{\text{pcent_augm}}{100}\right) \Delta t^i$.

If `MODE_CALCUL_TPLUS=' DELTA_Grandeur'`, then the coefficient c is related to the variation during the step of time of a selected size. The following formula is used:

$$c = \min_{i \in N} \left(\frac{V_{ref}}{|\Delta V_i|} \right)$$
 with N the whole of the nodes or the points of Gauss for which ΔV_i is not null. The value of reference V_{ref} is chosen by the user.

If the field ΔV is identically null, then one considers that the criterion (the event) is not checked.

If `MODE_CALCUL_TPLUS=' ITER_NEWTON'`, then the coefficient c is related to the variation during the step of time amongst iterations of Newton. The following formula is used:

$$c = \sqrt{\frac{N_{ref}}{N_{it} + 1}}$$
 where N_{it} is the iteration count of Newton carried out and N_{ref} a value of reference

chosen by the user. It is pointed out that the prediction is regarded as the iteration of Newton n°0, which means that if one converges as of the prediction then $c = \sqrt{N_{ref}}$.

If `MODE_CALCUL_TPLUS=' IMPLEX'`, then the coefficient c is related to the internal increments of variables (cumulated plastic deformation or damage) of the preceding step (see [R5.03.81]). In this case, the keyword `ADAPTATION` must be used only once. The coefficient c inevitably will lie between 0.5 and 1.2. Moreover, by definition of the method `IMPLEX` there will be no cause of failure.

3.3.4 Operand `PCENT_AUGM`

◇ `PCENT_AUGM` = /100. , [DEFECT]
/pcent , [R]

This operand is used to specify the percentage of increase in the step of time if `MODE_CALCUL_TPLUS = 'FIXES'`. By default, one increases by 100%, i.e. one doubles the step value of time. The value can be negative and must be strictly higher than -100.

3.3.5 Operands `VALE_REF`, `NOM_CHAM`, `NOM_CMP`

◆ `VALE_REF` = valref, [R]
◆ `NOM_CHAM` = /'DEPL',
/'SIEF_ELGA',
/'VARI_ELGA',
◆ `NOM_CMP` = cmp, [K8]

These operands are used to specify the calculation of the coefficient c of increase in the step of time if `MODE_CALCUL_TPLUS = 'DELTA_Grandeur'` (cf §3.3.3) with `VALE_REF` the value of reference V_{ref} and `NOM_CHAM` the name of the field of size considered.

3.3.6 Operand `NB_ITER_NEWTON_REF`

◆ `NB_ITER_NEWTON_REF` = nitref, [I]

This operand is used to inform the iteration count as Newton of reference, noted N_{ref} (cf §3.3.3) if `MODE_CALCUL_TPLUS = 'ITER_NEWTON'`.

3.4 Operand `INFORMATION`

◇ `INFORMATION`

For the moment, this operand is not useful.