Procedure DEBUT

1 Drank

Affecter the resources memory, disc and files.

The execution consists of a set of commands starting with DEBUT and ending in FIN [U4.11.02], (see also procedure POURSUITE [U4.11.03]).

The command DEBUT which is carried out, as of its reading by Superviseur, carries out the following tasks:

- definition of the characteristics of data bases (managed by JEVEUX) and assignment of the associated files,
- reading of the catalogues of the elements and the commands.

The apparently complex syntax of this procedure should not worry the user; its call with the operands by default, sufficient in most case, is: DEBUT ()

Les operands are to be used in the case of studies requiring a more important size of the files “data bases” or to divert the various files on numbers of logical unit different from the numbers affected by defect.

The commands placed before DEBUT, if they are syntactically correct, are ignored.

Contents

2 Syntaxe

```
DEBUT
      (  ◊PAR_LOT          =  /  "OUI",                              [DEFECT]
                               /  "NON",
         ◊IMPR_MACRO       =  /  "NON",                              [DEFECT]
                              /  "OUI",

         ◊LANG   = Lang,                                             [TXM]

         ◊BASE   = _F (       ♦FICHIER   = /  "GLOBALE",
                                           /  "VOLATILE",
                                           /  |LONG_ENRE = lenr,    [I]
                                              |NMAX_ENRE = nenr,    [I]
                                              |LONG_REPE = lrep,    [I]
                       ),

         ◊CODE   =_F       ( ♦NOM    =nom      code,                  [K8]
                           ♦NIV_PUB_WEB=          / "INTERNET",
                                                  / "INTRANET",
                           ◊VISU_EFICAS=          /  "OUI",          [DEFECT]
                                                  /  "NON",
                           ),

         ◊ERREUR    = _F   (ERREUR_F=      /  "ABORT",              [DEFECT]
                                           /  "EXCEPTION",
                              ),
         ◊IGNORE_ALARM =l_vale  ,                                  [l_Kn]

         ◊DEBUG  =     _F   ( JXVERI =  /  "OUI",
                                        /  "NON",
                                  ENVIMA =     'TEST',              [l_Kn]
                              ◊JEVEUX    = /  "OUI",
                                           /  "NON",
                              ◊SDVERI    = /  "OUI",
                                           "NON",
                              ◊HIST_ETAPE   = /  "NON",
                                              "OUI",
                                 ),
         ◊MESURE_TEMPS = _F (
                              ◊NIVE_DETAIL  = / 0                  [DEFAUT]
                                              / 2/1/3


                              ◊MOYENNE   =    / "OUI"              [DEFECT]
                                              / "NON"
                              ),
         ◊MEMOIRE   =_F    ( ◊TAILLE_BLOC=          /  800. ,
[DEFAUT]
                                                    /tbloc ,       [R]
                       ),

         ◊CATALOGUE        = _F (   ♦ FICHIER=       nfic,
[l_Kn]
                              ◊ UNITE=      unit,                 [I]
                          ),

         ◊RESERVE_CPU   =_F    ( / VALE=vale                        [R]
                                 / POURCENTAGE=    pcent           [R]

                                 ◊ BORNE=     /  bv,               [R]
```

```
                                                  /180   .              [DEFAUT]
                                        ),
        )
```

3 Affichages

At the beginning of the execution of Code_Aster, a heading is displayed. One finds there:
- the identification specifies version used: model number, date of the last amendments,
- the date and the hour of the beginning of the execution,
- the name, architecture, the operating system of the machine,
- the language used for the display of the messages,
- the type of parallelism available (MPI/OpenMP), the number of allocated processors,
- the version of the libraries used (when it is available) for hdf5, med, mumps, Scotch tape,
- then several information on the distribution of the memory.
  For example:

```
Limiting memory for the execution: 256.00 Mo
consumed by initialization: 148.68 Mo
by the objects of the command set: 17.48 Mo
remains for the dynamic allocation: 89.84 Mo
Limiting Taille of the files of exchange: 48.00 Go
```

Ce qui means:
- `256 Mo` is the quantity of memory required by the user, it is the total quantity which one should not exceed.
- `148.68 Mo` is consumed simply by booting the execution (loading of executable, the associated dynamic libraries, etc).
- `17.48 Mo` is consumed by the reading of the command file (Remarque: in mode PAR_LOT='NON', the command set being read progressively, this value will be then null).
- `89.84 Mo` is the quantity of memory available (at this time) for the objects of computation (equalizes to 256-148.68-17.48). It is thus seen that computation cannot begin if this value is too low.

During the execution, according to the dynamic allocations carried out, when this value varies of more than 10 % (upwards or downwards), a message of this type informs the user:

```
The memory currently consumed except JEVEUX is of 214.08 Mo.
The limit of dynamic allocation JEVEUX is fixed at 41.92 Mo.
```

At the end of the execution, an assessment indicates if same computation can be started again with less memory:

```
The memory requested from launching is over-estimated, it is of 256 Mo.
The peak report used is of 216.02 Mo.
```

or so more memory is necessary (indeed according to the platforms, the maximum limit can be exceeded without the system stopping computation):

```
The memory requested from launching is underestimated, it is of 256 Mo.
The peak report used is of 273.22 Mo.
```

4 Operands

4.1 Opérande PAR_LOT

PAR_LOT =

Mode of processing of the commands:

"OUI": (default choice); the supervisor analyzes **all** the commands before asking for the execution of it.

"NON": after having analyzed a command the supervisor asks for his execution then passes to the analysis (and the execution) of the following command (processing orders by command).

4.2 Key word IMPR_MACRO

IMPR_MACRO =

Autorise or not displays produced by the macros in the file of message. The reading of the files of message can be painful when it contains the totality of the echoes of the subcommands generated by macro itself. By default, only the echo of the commands explicitly called by the user in his command set will appear.

4.3 Mot-clé LANG

It makes it possible to choose the language of display of the messages transmitted by the code.
If the key word is not indicated, in fact the environment variables determines the language of the messages (reference: http://www.gnu.org/software/gettext/manual/gettext.html#Users). One can for example define in the file ~/.bashrc : LANG=fr_FR.UTF-8 export.
Encoding (UTF-8 or ISO-8859-1) makes it possible to correctly display the accentuated characters.
The LANG key word expects a value in two letters, for example "FR" (for French) or "IN" (for English).
When a language is selected (that it is by the environment or LANG), still it is necessary that the file of the translated messages (file .mo) is available. This file is expected under this name:

$ASTER_ROOT/share/local/"Lang `/LC_MESSAGES/aster_ "version ` .mo

where $ASTER_ROOT is the main directory of *Code_Aster* (e.g.: /aster or /opt/aster), Lang is the name in small letters of the language (e.g. in, Fr, of...) and version is the name of the version of *Code_Aster* used (stable e.g., testing, unstable).
If the file of translation cannot be read, it is French who is used.

Remarque

Même if the file of translation exists, when a message was not translated, there is displayed in French (language used of the messages in the source code).

4.4 Key word BASE

BASE =

the functionality of this key word is to redefine the values of the parameters of the files of random accesses associated with “data bases” if one does not wish to use those built-in by defect.

Values by default of the parameters associated with data bases.

GLOBALE		
NMAX_ENRE	62914	
LONG_ENRE	100	Kmots
LONG_REPE	2000	
VOLATILE		
NMAX_ENRE	62914	
LONG_ENRE	100	Kmots
LONG_REPE	the 2000	

word is worth 8 bytes out of platform 64 bits under LINUX 64, TRU64 and IRIX 64.4 bytes out of platform 32 bits under SOLARIS, HP-UX and WINDOWS - NT, LINUX.

Under Linux 64, with the default values, procedure DEBUT will allocate a file of random access of to more the 62914 records of 100 Kmots (the K is worth 1024) for base "GLOBALE".

Note:

The real size of the file is dynamic; it depends on the volume of information to store indeed. But this size is limited by the conditions of operating and a parameter preset among the values characterizing the platform. On the platform of reference Linux 64 bits, the maximum size is fixed

at 48 Go. This value can be modified while passing an argument on the command line of executable behind the key word "- max_base size" where size is an actual value measured out of Mo.
On the platforms 32 bits, the maximum size is fixed at 2,047 Go (2 147,483,647), but the code manages several files to go beyond this limit when the parameter "- max_base" passed in argument.
For the Globale base, which can be saved and re-used in data of a computation, the maximum size in "POURSUITE" is preserved such as it is if the parameter "- max_base" is not used, but perhaps redefined with the need for this manner.

4.4.1 Operand FICHIER

♦FICHIER =

Symbolic name of the base considered.

4.4.2 Operands LONG_ENRE / NMAX_ENRE / LONG_REPE

Définition of the parameters of data base (files of random access).

/ | LONG_ENRE = lenr

lenr is the length of the records in Kmots of the files of random accesses used.

Note:

The manager of memory JEVEUX uses this parameter to determine two types of objects: the large objects which will be cut out in as many records as necessary, and the small objects which will be accumulated in a buffer of the size of a record before being discharged.

| NMAX_ENRE = nenr

nenr is the number of records per defect, this value is given starting from LONG_ENRE and of an operating parameter on the platform of reference Linux 64 fixed at 48 Go (51 539 607 552 bytes) for the maximum size of the file associated with a data base, if this value were not modified by the use of the key word - max_base on the line of ordering of the executable one.

Note:

Two operands LONG_ENRE and NMAX_ENRE must be used with precaution, a bad use which can lead to the brutal stop of the program by saturation of the files of random access. Coherence between the maximum size of the file and the value resulting from the product of two parameters LONG_ENRE and NMAX_ENRE is checked at the beginning of execution.

| LONG_REPE = lrep

lrep is the initial length of the directory (maximum number of addressable objects by JEVEUX), it is managed dynamically by the manager of memory which extends the size of the directory and all the associated system objects as needs.

Note:

The choice by the user to modify these various parameters determines in a final way certain characteristics of GLOBALE data base which cannot be modified in POURSUITE any more.

4.5 Key word CODE

CODE =

Définition of a name for the group of a study. This key word is intended only for the command files of the tests of non regression managed with the source code.
The presence of this key word starts the emission of a message of information and automatically positions the mode of deboggage DEBUG (JXVERI = "OUI",) which implements checks on objects JEVEUX, which can bring a overcost to the execution. The behavior in the event of error can be modified.

4.5.1 Operand NOM

♦NOM = name codes

Nom of identification of the study, this name is with more than 8 characters.

4.5.2 Operand NIV_PUB_WEB

♦ NIV_PUB_WEB = "INTRANET"

Indicateur of level of publication. Meaning that the test is only diffusable on the internal network.

NIV_PUB_WEB = "INTERNET"

Indique which the test is diffusable such as it is on the external network.

VISU_EFICAS = "OUI"

Indique which the command file can be open without problem with tools EFICAS. This key word is primarily used for the tests and at ends of receipt of the new versions of the tools.

VISU_EFICAS = "NON"

Signale the presence of python source in the command file not allowing its edition with tools EFICAS.

4.6 Key word ERREUR

Permet to modify the behavior of the code in the event of <F> error.

4.6.1 Operand ERREUR_F

En cas de error, the code stops the normal execution of the command set.
By default, an exception is then raised (for the detailed definition of a Python exception, one will refer to the documentation of Python or that of the supervisor, *cf* [U1.03.01]). In this case, the code carries out the command FIN (*cf* [U4.11.02]) which closes the base then in order to allow the possible continuation of computation. It will be noticed that, although the initial error is known as "fatal" (<F>), the diagnosis is <S>_ERROR since the exception "is recovered" by FIN. This base will be then recopied by the driver of studies. This is the behavior when ERREUR_F='EXCEPTION".

If ERREUR_F='ABORT", that means that one explicitly asks the code definitively to stop the execution of the command set in the event of fatal error (<F>). Command FIN is not carried out, the base is thus not closed correctly, it is not recopied and no resumption of computation is possible.

Pour

remarks the execution of the benchmarks by the developers, the stop by ABORT is automatic and by defect. This is activated by the presence of the key word factor CODE (except if ERREUR_F specifies another thing).

In the event of lack of time CPU, of memory, for all errors of the type <S> and the exceptions, the behavior that is described when ERREUR_F='EXCEPTION' .

4.7 Key word IGNORE_ALARM

IGNORE_ALARM =

Permet with the user to remove the display of certain alarms (of which he knows the origin) in order to more easily identify other alarms which could appear.
During the execution of command FIN, one systematically displays a summary chart of the alarms emitted during the execution (and the number of occurrences). The alarms ignored by the user are preceded by "*" to distinguish them (and they appear even if they were not emitted).
Alarms are indicated starting from the bill of materials appearing between the characters < and >, for example: IGNORE_ALARME = ("MED_2", "SUPERVIS_40",…).

4.8 Key word DEBUG

DEBUG =

Option of déboggage (reserved for the developers and the maintenance of the code).

4.8.1 Operand JXVERI

JXVERI =

Permet to control the integrity of the segments of the memory between two executions of consecutive commands. By defect the execution is carried out without "DEBUG". This option is systematically activated in the presence of key word CODE.

4.8.2 Operand ENVIMA

ENVIMA = "TEST"

Permet to print in file RESULTAT the values of the parameters preset in software package ENVIMA characterizing the machine [D6.01.01].

4.8.3 Operand JEVEUX

◊JEVEUX =

Permet to activate the operating mode in debug of the manager of memory JEVEUX: unloadings on disc not differed and assignment from the segments values to an indefinite value [D6.02.01].

4.8.4 Operand SDVERI

SDVERI = "NON'

The use of this key word is bound for the developers. Attention, this functionality little cause a considerable overcost during the execution.
This key word starts the checking of data structures produced by the operators. It is used in the frame of the procedures of development of the code in the tests of non regression. If key word CODE is present, this key word takes the default value "OUI'.

4.8.5 Operand HIST_ETAPE

HIST_ETAPE = "NON'

This key word makes it possible to preserve all the history of the stages/commands used. This is greedy in memory and must be used only for quite particular cases (the commands which require it indicates it in their documentation).
By default, this history is not preserved.

4.9 Key word MESURE_TEMPS

key word MESURE_TEMPS makes it possible to choose the level of detail of the impressions of times CPU which will be displayed in the file of messages by the commands carrying out of elementary computations, of the resolutions of systems linear, the unloading of objects on disc or communications MPI.

4.9.1 Operand NIVE_DETAIL

Par defect, at the end of each command, one will print a line of the type:

```
#1.Resolution.des.systemes.lineaires         CPU. (USER+SYST/SYST/ELAPS):  7.52 0.79 11.22
# 2.Calculs.elementaires.et.as semblages       CPU. (USER+SYST/SYST/ELAPS): 15.07 0.70 15.77
```

◊NIVE_DETAIL=0 no printing.
=1 by default printings.
=2 more detailed printings:

```
#1.Resolution.des.systemes.lineaires          CPU (USER+SYST/SYST/ELAPS):  7.72 0.82  8.72
#1.1.Numerotation, .connectivity .de.la.ma trice CPU (USER+SYST/SYST/ELAPS):  0.21 0.02  0.31
#1.2.FACTORISATION.SYMBOLIC                 CPU (USER+SYST/SYST/ELAPS):  0.58 0.05  1.28
#1.3.Factorisation.numerique. (ou.precond.)    CPU (USER+SYST/SYST/ELAPS):  6.78 0.73  7.71
#1.4.Resolution                              CPU (USER+SYST/SYST/ELAPS):  0.15 0.02  0.35
# 2.Calculs.elementaires.et.as semblages       CPU (USER+SYST/SYST/ELAPS): 28.87 0.64 29.47
#2.1.Routine.calcul                          CPU (USER+SYST/SYST/ELAPS): 26.61 0.56 26.61
#2.1.1.Routines.te00ij                       CPU (USER+SYST/SYST/ELAPS): 24.58 0.07 25.78
#2.2.Assemblages                             CPU (USER+SYST/SYST/ELAPS):  2.26 0.08  3.36
#2.2.1.Assemblage.matrices                   CPU (USER+SYST/SYST/ELAPS):  2.02 0.06  3.12
#2.2.2.Assemblage.seconds.membres            CPU (USER+SYST/SYST/ELAPS):  0.24 0.02  0.37
```

=3 more detailed printings and incremental printing for each time step

.

During parallel computations (MPI), the time spent in the communications is also displayed:

```
#4 Communications MPI                       CPU (USER+SYST/SYST/ELAPS)    : 12.67  0.50 12.68
```

4.9.2 Operand MOYENNE

◊MOYENNE= 'OUI" display of the statistics (defect)
= 'NON" not of display of the statistics

key word MOYENNE makes it possible to exclusively control the display of additional statistics for parallel computations. It is the average of measurements on all the processors as well as the standard deviation of these measurements.

By default each time displayed is supplemented as follows:

```
#1 Résolution.des.systèmes.linéaires       CPU (USER+SYST/SYST/ELAPS):  0.29  0.00  0.35
       (average….diff. .procs)            CPU (USER+SYST/SYST/ELAPS):  0.30  0.00  0.47
       (variation-type.diff. .procs)            CPU (USER+SYST/SYST/ELAPS):  0.01  0.00  0.05
```

4.10 Key word MEMOIRE

the assignment of various data structures is a dynamic allocation, the user indicates the limits of resource during launching of executable in the interface of access.

4.10.1 Operand TAILLE_GROUP_ELEM

TAILLE_GROUP_ELEM = tgrel [defect: 1000]

This parameter gives the maximum number of finite elements of the same type which will be gathered in a group of elements.
This parameter influences the performances memory and CPU of elementary computations and the assemblies.
When tgrel is increased, one must in general save time CPU. On the other hand, objects JEVEUX are larger, which can require more memory.

4.10.2 Operand TAILLE_BLOC

TAILLE_BLOC = tbloc [defect: 800.]

This parameter gives the size of the blocks of the matrixes factorized for solver LDLT. This size is given in kiloR8 (1 kiloR8 = 1024 realities). This parameter influences the number of operations of input/output and thus over the time of assembly and resolution. By defect this value is fixed at 800 kiloR8, that is to say 8 records per defect on the file of random access associated with base JEVEUX.

4.11 Key word CATALOGUE

This key word is reserved for the developers, it is used at the time of the operation of compilation of the catalogues of elements to obtain the file in the form of base JEVEUX.

4.11.1 Operand FICHIER

♦ FICHIER = nfic

can take only value" CATAELEM"

4.11.2 Opérande UNITE

◊ UNITE = logical

Numéro unit of unit associated with the catalogues of elements. In the procedures of construction of the catalogue of elements one uses like value 4. The file fort.4 is obtained starting from the directory contents of the catalo sources using a procedure python.

4.12 Key word RESERVE_CPU

Permet to reserve a share of the time CPU allotted to the job to finish the execution in the event of stop properly for lack of time CPU detected by a Aster command. This mechanism is useful only in the case of an execution batch of *Code_Aster*. The value of this reserve can be indicated in absolute value or in the form of a percentage of total time CPU. This value is limited by the value of the key word LIMITS.
When key word CODE is present, i.e. for all the tests of non regression, one imposes systematically a reserve of time 10 second old CPU if key word RESERVE_CPU is absent.

4.12.1 Operand VALE

Valeur expressed in seconds withdrawn from the total time CPU, over which certain total commands is based to stop the execution properly.

4.12.2 Operand POURCENTAGE

Pourcentage withdrawn from the total time CPU, over which certain total commands is based to stop the execution properly.

4.12.3 Operand LIMITS

Valeur maximum of the reserve of time, being worth by defect 180 seconds.