Operator CALC_IFS_DNL

1 Drank

the object of this command is to allow the computations fluid-structures coupled in nonlinear transient regime. For that, one comes to couple *Code_Aster*, for the structure part, in *Code_Saturne*, for the fluid field, *via* supervisor YACS of Salomé.

The method of coupling is of type partitionné Neuman-Dirichlet. To solve the problem structure, one bases oneself on the operator DYNA_NON_LINE, which one very largely takes again syntax.

Contents

2 Syntax

Syntax specific to the YEWS:

```
♦  PAS_INIT          =pdtinit                        [R]
♦GROUP_MA_IFS      =    lgrmaifs,               [l_gr_maille]
♦  NOM_CMP_IFS       =    lcompifs,                  [l_Kn]
◊UNITE_NOEUD       =/ulnoeud        ,               [I]
                         /  81,                      [DEFAULT]
◊UNITE_ELEM         =/ulelem         ,               [I]
                         /  82,                      [DEFAULT]

   )
```

3 Principle of operation

the method of coupling is of type partitionné Neuman-Dirichlet. To solve the problem structure, one bases oneself on operator DYNA_NON_LINE of *Code_Aster.* The fluid field will be solved with *Code_Saturne*. The meshes with the interface not being obligatorily in conformity, it is necessary to use an operator of projection of fields: one has chooses to use PROJ_CHAMP. Who will manage all the stages of projection (*Code_Saturne* will not have thus any projection to make in-house).

One can summarize the algorithm of coupling as follows:
with each time step, *Code_Aster* sends displacement and velocity calculated to *Code_Saturne* , which in deduced a strain from the fluid grid and solves the fluid problem top (of description ALE). The forces fluid with the walls are then sent towards *Code_Aster* which can then solve the new problem structure on a step.

In this simple form the algorithm is explicit and that time step imposes one small enough for reasons of conditional stability [R5.05.05]. In practice it is not inevitably very penalizing because the resolution of the fluid problem often claims one time step rather small.
It is possible to define a implicitée version of the method of coupling. It is enough, at each time step to introduce an iterative process of type not fixes. That makes it possible to use time step the larger, but with a overcost of computation related to the iterations of fixed point.

All the data exchanged (they are scalar or vectorial) between the two computer codes pass by calls YACS. The use of the coupling YEWS thus passes obligatorily through Salomé who will control the two codes: *Code_Aster* and *Code_Saturne*. One cannot thus carry out this kind of computation by means of classically the interfaces of launching of *Code_Aster* : astk or as_run.
This documentation is restricted to only describe the use with the meaning *Code_Aster* .

The resolution of the structure part is done thanks to operator DYNA_NON_LINE, which explains why the syntax of CALC_IFS_DNL is in very great identical part. One thus re-examines with U4.53.01 documentation for all the key word commun runs with DYNA_NON_LINE.

The only differences in syntax, which are detailed in this documentation, are related to:

- the management of time: one does not use factor key word the INCREMENT, because control in time is managed by the coupler itself
- the definition of the characteristics of the interface fluid-structure.

CALC_IFS_DNL produces a concept of the usual evol_noli type.

4 Definition of the temporal discretization

control in time is in fact off-set out of *Code_Aster*. More precisely, the coupler will evaluate at every moment computation the current step and will provide it to the two codes which are *Code_Aster* and *Code_Saturne*. In practice, each one of these codes provides only one time step initial which makes it possible the coupler time step to evaluate the first.

In the same way, information of initial time and final time of study are defined in the level of the coupler itself and not in the command file.

Code_Aster will recover all this information (initial time, final moment, not running) *via* YACS.

4.1 Key word PAS_INIT

Defines the time step initial one for the coupling YEWS, within the meaning of time step relevant for computation structure alone. *Code_Saturne* defines in the same way its clean time step initial and the coupler then will return to the two codes the time step initial one which will be really used for the coupled resolution. In practice, this time step coupled will be the minimum of both time step coming from the codes, in order to observe and quality the stability conditions of the solution on each of the two fields.

If one uses an explicit time scheme in *Code_Aster*, then, obviously, the time step initial one will have to observe the Flow condition (or CFL, [U4.53.01] and [R5.05.05]).

5 Definition of the interface fluid-structure

the user must specify the interface fluid-structure. It is advisable to recall that with this interface, the fluid and solid meshes are not inevitably in conformity. Moreover, *Code_Aster* managing all the stages of projection between the fluid and solid, it is necessary to give him the information of the fluid mesh.

5.1.1 Operand GROUP_MA_IFS

This operand makes it possible to define the mesh group of the solid mesh which is with the interface fluid-structure.

5.1.2 Operand NOM_CMP_IFS

One specifies which components of the force will be transmitted to the interface fluid-structure, in the absolute coordinate system.

For example:
`NOM_CMP_IFS = (“FX”, “FY”, “FZ”),`
to have complete transmission of the forces in 3D.
If one wishes to transmit only certain components, it is enough to exclude the unutilised components. One can thus carry out conditions of sliding to the wall.

5.1.3 Operands UNITE_NOEUD and UNITE_ELEM

These operands make it possible to define the unit logical of the files containing the meshes corresponding to the interface fluid-structure resulting from the fluid mesh. The resolution by Code_Saturne being done in finished volumes, it is necessary to define two mesh distinct for projections from fields to be exchanged.

In the Neuman-Dirichlet coupling, the code structure provides to the fluid code displacements and velocity with the interface. They are thus data with the nodes of the solid mesh which one projects on the nodes of the fluid mesh. The mesh fluid of the interface is recovered of *Code_Saturne*, *via* YACS and will be written, with the format of Aster mesh in the file having the logical unit `UNITE_NOEUD` (which is worth 81 per default). One can thus also recover this mesh in postprocessing if need be.

The second phase of the coupling is done in the other meaning: the fluid code provides to the code structure the forces with the interface (according to the components given with `NOM_CMP_IFS`). More precisely, *Code_Saturne* being a code in finished volumes, the calculated surface forces are constant by face and what *Code_Aster* recovers is in fact the resultants by face of the elements, expressed with the nodes mediums. So that *Code_Aster* can project on the mesh structure, it is thus necessary to have the mesh of the nodes mediums of the fluid mesh for the interface. This mesh is also recovered via YACS and one writes it with the Aster format in the logical unit `UNITE_ELEM` (which is worth 82 per default).