
Operator RECU_PARA_YACS

1 Drank

the object of this command is to manage certain exchanges of parameters for the coupling YEWS by calls YACS. The exchanged data are smalls (scalar): one does not exchange fields.

This operator is used by the macro-command `CALC_IFS_DNL` (cf U7.06.01 documentation) which allows 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 calls to command `RECU_PARA_YACS` can be summarized in two big classes:

- initialization of the algorithm of coupling,
- exchange, with each step (or iteration) of parameters reactualized like time step or an indicator of convergence.

Contents

1 But1.....	
2 Syntaxe3.....	
3 Principle of fonctionnement4.....	
4 Type of data to exchange: key word DONNEES4.....	
4.1 Initialization of the coupler: DONNEES = "INITIALIZATION".....	4
4.2 Recovery of time step running: DONNEES = "NOT"	5.4.3
Recovery of the parameter of convergence: DONNEES = "CONVERGENCE"	5.4.4
Management of the end of computation: DONNEES = "FIN"	5

2 Syntax

```
resu      [listr8_sdaster] = RECU_PARA_YACS      (
    ♦      DONNEES=/                "INITIALIZATION",                [kN]
                                /"CONVERGENCE",
                                /"FIN",
                                /"NOT",

{If      DONNEES=' INITIALIZATION"
    ♦      NOT                =pasinit      ,                [R]
}
{If DONNEES=/                "CONVERGENCE",
    /    "FIN",
    /    "NOT",
    ♦      NUME_ORDRE_YACS =numeyacs                [I]
    ♦      INST                =inst                [R]
    ♦      NOT                =pas                [R]
}

    ♦INFO=/                1      ,
[DEFAULT]
                /2      ,

    ♦TITER=                tx      ,                [kN]

)
```

3 Principle of operation

the method of coupling for the interaction fluid-structure based on *Code_Aster* and *Code_Saturne* requires the data exchange between these codes. Indeed, it is not about a monolithic approach where all the coupled problem would be solved in a single computer code: one couples two codes, each one being confined with his speciality. This coupling is managed by the command `CALC_IFS_DNL` (U7.06.01).

All the data to be exchanged use protocol YACS of Salomé.

These data can be of two different nature:

- parameters of smalls (of the scalars, for example),
- fields (displacements, velocities or forces with the interfaces, for example).

In order to keep a good modularity, favourable with the evolutions, different operators were thus developed, each one treating one of the types of data to be exchanged.

The scalar data are handled by `RECU_PARA_YACS`, the fields by `ENV_CINE_YACS` (U7.07.01) and `MODI_CHAR_YACS` (U7.08.02). One can also indicate the operator `IMPR_MAIL_YACS` (U7.08.03) who recovers, *via* YACS, the fluid meshes of the interfaces. All these commands are called by `CALC_IFS_DNL`.

Operator `RECU_PARA_YACS` allows the scalars data exchange in the two meanings. He generates a list of realities which will contain all the parameters recovered following the communications with YACS.

The various use potential of `RECU_PARA_YACS` is managed by the key word `DONNEES` which makes it possible to specify the type of data to be exchanged.

Exchanges YACS require specific arguments of entry: time running, time step and the call number (which one can compare to a sequence number). This is why one finds the corresponding key words like arguments of this operator.

4 Type of data to be exchanged: key word `DONNEES`

This key word makes it possible to differentiate the uses of `RECU_PARA_YACS`, which one uses at the beginning of the transient and each time step.

It is pointed out that all the management of time is off-set out of *Code_Aster*. It is thus necessary, for the initialization of the coupler, to exchange several parameters between the codes.

Moreover, the coupler vaensuite to evaluate computation at every moment the current step and to 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 the necessary information (initial time, final moment, not running) *via* YACS.

4.1 Initialization of the coupler: `DONNEES = "INITIALIZATION"`

to begin the coupled resolution, each computer code provides time step initial, which, on the level of `RECU_PARA_YACS` in the case `DONNEES = "INITIALIZATION"`, will be given by the user with the key word `NOT`. One thus sends with YACS the value thus entered, named `pasinit`.

All the parameters necessary to the definition of the coupling are then recovered, by receptions YACS and are gathered in the list of realities which is produced by the operator `RECU_PARA_YACS`.

This list is composed of 7 realities (of which some are converted into integer *a posteriori* in `CALC_IFS_DNL`):

- nombre total of time step with calculating,
- maximum number of iterations to each time step into implicit (convert in integer),
- limiting value of the convergence criterion,
- indicator of output in postprocessing (converted into integer and if it is worth 1 one will make a Enight output),
- not archivage (converted into integer),
- initial time,
- time step for the first computation step (minimal value of the initial steps of the two codes).

The format of Enight output is used for reasons of homogeneity with the outputs of *Code_Saturne*. For more simplicity this output is included in `CALC_IFS_DNL`. All postprocessing usual are of course realizable after `CALC_IFS_DNL`.

4.2 Recovery of time step running: `DONNEES = "NOT"`

time step running is not known *a priori* in *Code_Aster* : it is furnished information by the coupler starting from the data of the codes.

Thus with each step, *Code_Aster* must recover this information, which is done with the call to `RECU_PARA_YACS` with the value "NOT" for key word `DONNEES`.

That thus requires an exchange YACS which, as all exchanges YACS needs certain arguments as starter (except for the initialization):

- time running given by the key word `INST`,
- the last time step known with the key word `NOT`,
- the call number YACS (sequence number managed by `CALC_IFS_DNL`).

In output one recovers only one reality: the time step updated one.

4.3 Recovery of the parameter of convergence: `DONNEES = "CONVERGENCE"`

operation is very close to the preceding case: with each step one recovers an indicator of convergence calculated apart from *Code_Aster*. The arguments of entry are the same ones as for the recovery of time step running.

4.4 Management of the end of computation: `DONNEES = "FIN"`

This option is currently not used by `CALC_IFS_DNL`.