
Operator RECU_PARA_YACS

1 Goal

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

This operator is used by the macro-order `CALC_IFS_DNL` (cf. U7.06.01 documentation) which allows calculations fluid-structures coupled in non-linear transitory mode. For that, one comes to couple *Code_Aster*, for the structure part, with *Code_Saturne*, for the fluid field, via supervisor YACS of Salomé.

Calls to the order `RECU_PARA_YACS` can summarize itself in two big classes:

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

Contents

1 Goal.....	1
2 Syntax.....	3
3 Principle of operation.....	4
4 Type of data to be exchanged: keyword DATA.....	4
4.1 Initialization of the coupler: DATA = 'INITIALIZATION'.....	4
4.2 Recovery of the step of current time: DATA = 'NOT'.....	5
4.3 Recovery of the parameter of convergence: DATA = 'CONVERGENCE'.....	5
4.4 Management of the end of calculation: DATA = 'FINE'.....	5

2 Syntax

```
resu [listr8_sdaster] = RECU_PARA_YACS (  
    ♦ DATA = / 'INITIALIZATION', [KN]  
              / 'CONVERGENCE',  
              / 'FINE',  
              / 'NOT',  
  
    {So GIVEN = 'INITIALIZATION'  
              ♦ NOT = pasinit, [R]  
    }  
    {So GIVEN = / 'CONVERGENCE',  
              / 'FINE',  
              / 'NOT',  
              ♦ NUME_ORDRE_YACS = numeyacs [I]  
              ♦ INST = inst [R]  
              ♦ NOT = not [R]  
    }  
  
    ♦ INFORMATION = / 1,  
    [DEFECT] / 2,  
  
    ♦ TITLE = tx , [KN]  
)
```

3 Principle of operation

Method of coupling for the interaction fluid-structure based on *Code_Aster* and *Code_Saturne* require 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 order `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, speeds or efforts 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`, 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) which recovers, *via* YACS, fluid grids of the interfaces. All these orders are called by `CALC_IFS_DNL`.

The operator `RECU_PARA_YACS` allows the scalars data exchange in the two directions. It 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` are managed by the keyword `DATA` who allows to specify the type of data to be exchanged.

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

4 Type of data to be exchanged: keyword DATA

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

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 calculation 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 step of initial time which makes it possible the coupler to evaluate the first step of time.

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 moment, final moment, not running) *via* YACS.

4.1 Initialization of the coupler: `DATA = 'INITIALIZATION'`

To begin the coupled resolution, each computer code provides a step of initial time, which, on the level of `RECU_PARA_YACS` in the case `DATA = 'INITIALIZATION'`, will be given by the user with the keyword `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 entirety *a posteriori* in `CALC_IFS_DNL`):

- full number of step of time to calculate,
- maximum number of iterations to each step of time into implicit (convert in entirety),

- value limits convergence criteria,
- indicator of exit in postprocessing (converted into entirety and if it is worth 1 one will make a Ensign exit),
- pas d' filing (converted into entirety),
- initial moment,
- pas de time for the first step of calculation (minimal value of the initial steps of the two codes).

The format of Ensign exit is used for reasons of homogeneity with the exits of *Code_Saturne*. For more simplicity this exit is included in `CALC_IFS_DNL`. All postprocessing usual are of course realizable afterwards `CALC_IFS_DNL`.

4.2 Recovery of the step of current time: `DATA = 'NOT'`

The step of current time 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 the keyword `DATA`.

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

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

At exit one recovers only one reality: the step of updated time.

4.3 Recovery of the parameter of convergence: `DATA = '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 the step of current time.

4.4 Management of the end of calculation: `DATA = 'FINE'`

This option is currently not used by `CALC_IFS_DNL`.