Structure of data sd_eigensolver

Summary:

This document describes the structure of data EIGENSOLVER. This one clarifies with the modal problem dealt (GEP and QEP) as well as the parameters contiguous to the selected modal solvor.

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.
Copyright 2021 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Contents

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.
Copyright 2021 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

1 General information

This object of the type EIGENOLVER has as a function to store and convey between the various routines of the code (for the moment that in-house of the order CALC_MODES[1]), information describing a modal problem with the direction *Code_Aster* (GEP or QEP) as well as the parameters contiguous to the selected modal solvor (SORENSEN, TRI_DIAG, JACOBI or QZ). This object is created on the volatile basis.

It should be created and only fills *via* VPINIS . The coherence of its parameters and those of certain subjacent objects (e.g. them matrices defining the problem) are controlled *via* VPVERS . This routine also utilizes, optionnellement, the definition with the direction Python of this object (sd_eigensolver.py) *via* a call to the routine cheksd . In CALC_MODES this option is always activated.

Other routines dedicated (standard “method” to the direction C ++) allow moreover to handle this object:
- To entirely read again and update certain auxiliary variables: VPLECS .
- To see a given coefficient of this object: VPLECI .
- To write a given coefficient of this object: VPECRI .

Normally, all the fields of this object, once they were filled and checked, should not be modified any more. The only exceptions concern, for the moment, in CALC_MODES:
- ESVK (2) and ESVK (3) : in order to invert the matrices of work if OPTION= 'PLUS_GRANDE',
- ESVI (1) and ESVI (2) : recalculation/correction amongst modes and of the size of the space of projection.

During a modification of this structure of data it is thus necessary to take care of:
- *to put in coherence, if necessary, sources mentioned above,*
- *to update, if necessary, the catalogue sd_eigensolver.py ,*
- *to update documentations (this Doc. D and if necessary Doc. U concerning modal calculation),*
- *to enrich or modify, if necessary, some CAS-tests.*

2 Tree structure

```
SOLVEUR (K19)   :: =record
  ♦    '.ESVK'   :   OJB   S V K24  long=20  (initialized with '')
  ♦    '.ESVR'   :   OJB   S V R    long=15  (initialized with r8vide ())
  ♦    '.ESVI'   :   OJB   S V I    long=15  (initialized with isnnem ())
```

3 Contents of the basic objects

3.1 Vector ESVK

ESVK:

Parameters generals describing the modal problem

ESVK (1) : type of result ('DYNAMICS', 'MODE_FLAMB' or 'GENERAL').

ESVK (2) : name of the provided matrix *via* the keyword MATR_RIGI if DYNAMICS/MODE_FLAMB, MATR_A if GENERAL.

ESVK (3) : name of the provided matrix *via* the keyword MATR_MASS if DYNAMICS, MATR_RIGI_GEOM if MODE_FLAMB , MATR_B if GENERAL .

1 Routine op0045. F90.

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.
Copyright 2021 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

ESVK (4) : name of the provided matrix *via* the keyword MATR_AMOR if DYNAMICS, MATR_B if GENERAL.
ESVK (5) : type of calculation. The possible values are:
'PLUS_PETITE', 'CENTER', 'BAND' or 'ALL' (if MODE_FLAMB or GENERAL),
Idem more 'PLUS_GRANDE' (if DYNAMICS).
ESVK (6) : name of the modal solvor ('SORENSEN', 'TRI_DIAG', 'JACOBI' or 'QZ').
ESVK (7) : value of the option of pre-captures rigid modes ('WITHOUT' or 'MODE_RIGIDE').
ESVK (8) : Comportement to adopt in the event of band of empty calculation, option 'STOP_BANDE_VIDE'('YES' or 'NOT').
ESVK (9) : name of the table generated by INFO_MODE (option TABLE_FREQ/TABLE_CHAR_CRIT).
ESVK (10) : C omportement to adopt in the event of error, option ' STOP_ERREUR '('YES' or ' NOT ').
ESVK (11) : activation of the test of Sturm , option ' STURM '('YES' or ' NOT ').
ESVK (12) with ESVK (15): unutilised.

Parameters specific to the modal solveurs (cf. ESVK (6))
ESVK (16) : type of approach in QEP ('R', 'I' or 'It).
if ESVK (6)= ' QZ'
ESVK (17) : type of method QZ ('QZ_SIMPLE', 'QZ_EQUI' or 'QZ_QR').
If not:
ESVK (17) : unutilised.
ESVK (18) with ESVK (20) : unutilised.

3.2 Vector ESVI

ESVI:
Parameters generals describing the modal problem
ESVI (1) : many modes to be calculated.
ESVI (2) : value of the parameter DIM_SOUS_ESPACE.
ESVI (3) : value of the parameter COEF_DIM_ESPACE.
ESVI (4) : value of the parameter NMAX_ITER_SHIFT.
ESVI (5) : many frequencies or critical loads seizures by the user.
ESVI (6) with ESVI (10) : unutilised.

Parameters specific to the modal solveurs (cf. ESVK (6))
if ESVK (6)= ' TRI_DIAG'
ESVI (11) : value of NMAX_ITER_ORTHO .
ESVI (12) : value of NMAX_ITER_QR .
ESVI (13) with ESVI (15) : unutilised.
if ESVK (6)= ' JACOBI'
ESVI (11) : value of NMAX_ITER_BATHE .
ESVI (12) : value of NMAX_ITER_JACOBI .
ESVI (13) with ESVI (15) : unutilised.
if ESVK (6)= ' SORENSEN'
ESVI (11) : value of NMAX_ITER_SORENSEN .
ESVI (12) with ESVI (15) : unutilised.
if ESVK (6)= ' QZ'
ESVI (11) with ESVI (15) : unutilised.

3.3 Vector ESVR

ESVR:

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.
Copyright 2021 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Parameters generals describing the modal problem

ESVR (1) : first terminal of research seized by the user (keyword FREQ or CHAR_CRIT).
ESVR (2) : second terminal of research seized by the user (keyword FREQ or CHAR_CRIT).
ESVR (3) : value of the parameter CALC_FREQ/PREC_SHIFT.
ESVR (4) : value defining "the zero modal one": (2.pi.SEUIL_FREQ) ** 2 if DYNAMICS, SEUIL_CHAR_CRIT if not.
ESVR (5) : value of the parameter VERI_MODE/PREC_SHIFT .
ESVR (6) : value of the parameter VERI_MODE/SEUIL .
ESVR (7) with ESVR (10) : unutilised.

Parameters specific to the modal solveurs (cf. ESVK (6))

if ESVK (6)= ' TRI_DIAG'
ESVR (11) : value of PREC_ORTHO .
ESVR (12) : value of PREC_LANCZOS .
ESVR (13) with ESVR (15) : unutilised.
if ESVK (6)= ' JACOBI'
ESVR (11) : value of PREC_BATHE .
ESVR (12) : value of PREC_JACOBI .
ESVR (13) with ESVR (15) : unutilised.
if ESVK (6)= ' SORENSEN'
ESVR (11) : value of PREC_SOREN .
ESVR (12) : value of PARA_ORTHO_SOREN .
ESVR (13) with ESVR (15) : unutilised.
if ESVK (6)= ' QZ'
ESVR (11) with ESVR (15) : unutilised.

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.
Copyright 2021 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)