An example of matrix distributed for PETSc

Summary:

To use the solveurs of the PETSc bookstore in parallel, one must provide an assembled matrix distributed on the processors available according to a model of distribution imposed by PETSc (distribution by group of lines contiguëS). The model of distribution of the data *Code_Aster* rest on a distribution of the meshs between the processors. It leads to an assembled matrix which does not follow the model of PETSc distribution. One explains here while following a simple physical example various classifications of the degrees of freedom: local and total classification for Code_Aster then PETSc classification.

1 Introduction

The keyword `SOLVEUR 'MATR_DISTRIBUEE'` indicate if the matrix of the linear problem to solve is supplements on all the processors (`MATR_DISTRIBUEE=' NON'`), or distributed well on these processors (`MATR_DISTRIBUEE=' OUI'`). Dyears the two cases, the vector second member are complete on all the processors.

The distribution of the meshs by processor is defined in the model. There are several possible ways D'to carry out this distribution: it is the keyword `PARALLELISM` who selects the type of distribution.

Each processor assembles the elementary contributions of the degrees of freedom which it has, i.e. of the degrees of freedom which belong to meshs who were affected to himES. As one partitionné meshs (and not degrees of freedom), a degree of freedom can be had by several processors.

Each processor thus assembles a block of the total matrix. Each degree of freedom has a local index (on this processor) and a total index (in the total model).
When one uses a solvor PETSc, one passes an object of the Chechmate type to him, which is a matrix distributed with the model of distribution expected by PETSc. This distribution is a distribution per blocks of lines of matrix. One thus carries out a new distribution of the values of the matrix between the processors available for calculation in order to build a matrix distributed of PETSc type. Each degree of freedom thus has besides its local index and of its index total (in the model Aster) a total index in the PETSc matrix.

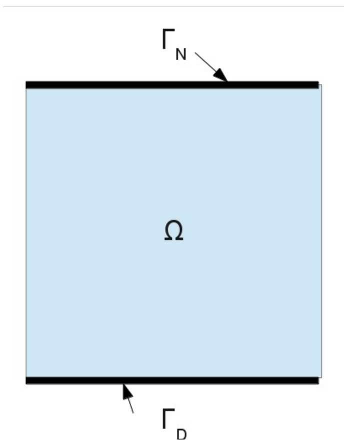
The object of this documentation is to clarify these various classifications starting from a simple example. It supplements documentation [D4.01.03].

2 Model problem

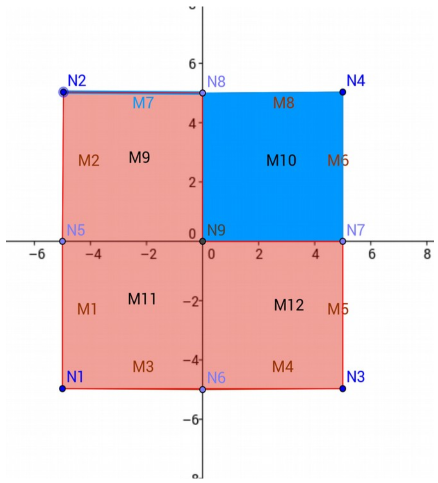
One considers a problem 2D static elasticity, distributed on two processors.

2.1 Geometry

The elastic range is a square of extreme points (- 50.50) and (50.50). This field in 4 quadrangles is netted.



(A) Geometry: one imposes a uniform pressure on the border Γ_{NR} and a condition of embedding on the border Γ_D



(b) Grid: Lbe names of the tops (N1 with N9) and meshs (M1 with M12) are in conformity with Code_Aster. There are 8 meshs 1D and 4 meshs 2D. The blue meshs M7, M8 and M10 are assigned to processor 0. The red meshs M9, M11 and M12 are assigned to processor 1

2.2 Properties of material

- $E=1{,}0\ 10^{11}\ N/m^2$
- $\nu=0{,}3$

2.3 Modeling 2D

The model is affected on the 4 quadrangular meshs (group of meshs 2D all) and on the 2 segments which constitute the higher edge of the field (group of meshs 1D up).

```
MODEL=AFFE_MODELE (MAILLAGE=MA,
                   AFFE=_F (GROUP_MA= ('all', 'up',),
                            PHENOMENE=' MECANIQUE',
                            MODELISATION=' C_PLAN',),
                   PARALLELISM =_F ( DISTRIBUTION = ' SOUS_DOMAINE', ),
                   );
```

A kind of distribution of the meshs was also defined (by under-field).

2.4 Limiting conditions and loading

One puts pressure distributed on the higher edge of the square, of ends (- 50.50) (50.50):

```
PRESSION=AFFE_CHAR_MECA (MODELE=MODEL,
                         PRES_REP=_F (GROUP_MA=' up',
                                      PRES=1000000000,),);
```

The base of the square is embedded: the condition is applied *DX*=0, *DY*=0 on the segment of ends $(-50,-50)$ $(50,-50)$.

This embedding is applied in two ways:

- modélisatioN a: aveC AFFE_CHAR_CINE
```
ENCASTR=AFFE_CHAR_CINE (MODELE=MODEL,
                        MECA_IMPO=_F (GROUP_NO=' bottom',
                                      DX=0,
                                      DY=0,),);
```
- modeling B: with AFFE_CHAR_MECA
```
ENCASTR=AFFE_CHAR_MECA (MODELE=MODEL,
                        DDL_IMPO=_F (GROUP_NO=' bottom',
                                     DX=0,
                                     DY=0,),);
```

2.5 Distribution of the problem

During the creation of the model, one chose a distribution of elementary calculations by under-field. One specifies with the solvor that the matrix is distributed:

```
MECA_STATIQUE (MODELE=MODEL,
               CHAM_MATER=AFMAT,
               EXCIT= (_F (CHARGE=PRESSION,),
                       _F (CHARGE=ENCASTR,),),
               SOLVEUR=_F (METHODE=' PETSC',
                           MATR_DISTRIBUEE=' OUI',
                           ALGORITHME=' GMRES',),);
```

Calculation is carried out on 2 processors.

3 General information on the linear system

One wants to solve a linear problem of elasticity. One recalls his variational formulation to the paragraph 3.1.

The discrete problem associated with the continuous problem is different according to the method used to apply the limiting condition of Dirichlet on the basis of field. One will describe it later on for each modeling. One gathers in this section the part common to modelings A and B.

3.1 Continuous problem

The constraints check the equation $-\nabla\cdot\sigma=0$ in the elastic range. Displacement u is fixed on the lower border of the field Γ_D by the condition of Dirichlet $u=u_0$. Here, $u_0=0$. On the higher border Γ_N, one applies the loading of Neumann $\sigma\cdot n=p$, for a pressure p uniform.

One writes the variational formulation of the problem. One defines:

- space closely connected $V=\{v, v\in(H^1(\Omega))^2, v=u_0 sur\,\Gamma_D\}$,
- associated vector space (displacements kinematically acceptable) $V^0=\{v, v\in(H^1(\Omega))^2, v=0\,sur\,\Gamma_D\}$

One seeks $u\in V$, such as $\forall v\in V^0$, $a(u,v)=(f,v)$, where a is the bilinear form of linear elasticity and f to the loading “pressure corresponds” on Γ_N.

3.2 Classification of the equations of the total Code_Aster matrix

The description of the classification of the matrix (total and local) is made by the objects of `NUME_DDL`. `NUME_EQUA` contains total information. One quickly points out the role of the fields which one will display the value and one returns to [D4.06.07] for the description of reference.

- `.NUMÉRIQUE.REFN` is a table of names, containing the name of the grid, the discretized size and the solvor. Here `REFN` contains:

MY	DEPL_R	GMRES

- `.NUMÉRIQUE.NEQU` is the full number of equations N
- `.NUMÉRIQUE.DELG` is a table of size N, which contains a marker of the degrees of freedom of Lagrange with convention:

$$DELG(i)=\begin{cases}0 & : & ddl\ physique\\ -1 & : & Lagrange\,1\\ -2 & : & Lagrange\,2\end{cases}$$

- `.NUMÉRIQUE.DEEQ` is a whole table of size $2N$ The equation I_{eq} is described by a couple of entireties (I_{No},I_{cmp}) in positions (I_{eq}-1) × 2+1 and (I_{eq}-1) × 2+2 of the table:
  - I_{No}>0 are the number of the node of the grid support of the degree of freedom I_{eq}.
    - If I_{cmp}>0, the degree of freedom correspond to the component I_{cmp} subjacent size.
    - If I_{cmp}<0, then the equation is one of the two equations of dualisation of the blocking of the component I_{cmp} of this size on the node I_{No}>0.
  - I_{No}=0 indicates that the equation I_{eq} is the equation of dualisation of a linear relation. In this case, I_{cmp} is inevitably null.
- `.NUMÉRIQUE.PRNO` of size 1 + the number of loads dualized in the model is a collection. One reaches these objects by the pointer of names `.LILI`. The first object collection contains the degrees of freedom associated with the grid (by convention `.LILI (1) =&MAILLA`) and following objects late degrees of freedom created by dualisation. For each node, one finds
  - the number of equation of the first component associated with this node,
  - the component count increased by this node

- a specifying vector of coded entireties which are the components of the size present on this node (see D4.06.05 for the system of coding)

Notice :

*Attention, there is an additional level of indirection (vector .NUEQ) who allows to find the position of the values of the terms of the matrix in the table .VALE . **This level voluntarily here is omitted** because it does not intervene in the example considered. It is only necessary for the static under-structuring. Indeed, one then needs to ensure that the degrees of freedom of interface are in the matrix after the internal degrees of freedom to a under-field. That is necessary to calculate the complement of Schur without having explicitly a structure of matrix per blocks. The static under-structuring corresponds to the order MACR_ELEM_STAT . A macronutrient is equivalent to a under-field. One calculates inter alia his rigidity and it can be integrated like an element into a model.*

3.3 Classification of the equations of the local matrix Aster

One places oneself now if the matrix *Code_Aster* is assembled by two processors.

In our example, processor 0 has the meshs M7, M8 and M10 and processor 1 the meshs M9, M11 and M12. Each processor assembles the elementary contributions corresponding to the meshs which it has. The characteristics of these local matrices are defined in .NUML NUME_DDL. Tables .NUML.NULG and .NUML.NUGL are opposite one of the other. They contain the correspondence between local classification with each processor of the degrees of freedom and total classification. Thus, if i_l is the local number of a degree of freedom and i_g its total number, one a:

$$\mathrm{NULG}(i_l)=i_g, \mathrm{NUGL}(i_g)=i_l$$

If the degree of freedom i_g is not present on a processor, $\mathrm{NUGL}(i_g)=0$.

4 The linear system for modeling A

The limiting conditions are applied by elimination. One explains the principle of elimination in the paragraph 4.1, and one gives the values of the various descriptors of the matrix in the paragraph 4.2.

4.1 The discrete system

The discrete system is written $K\,U=F$ where the matrix of rigidity K is of size $N\times N$.

One partitionne the whole of the indices of the degrees of freedom $S=[1:N]$, in two subsets:

- D for the indices of the degrees of freedom fixed by the limiting condition of Dirichlet (embedding). Here, $D=\{1,2,5,6,11,12\}$, which corresponds to fix DX, DY on the nodes $N1, N3, N6$ and $U_0=0$.
- I for the indices of the nonconstrained, unknown degrees of freedom: $I=S\setminus D$

One applies this partition with the system which is written:

$$\begin{pmatrix} K_{II} & K_{ID} \\ K_{DI} & K_{DD} \end{pmatrix} \begin{pmatrix} U_I \\ U_D \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \text{ with } U_D=U_0 .$$

One replaces this system by:

$$\begin{pmatrix} K_{II} & 0_{ID} \\ 0_{DI} & Id_{DD} \end{pmatrix} \begin{pmatrix} U_I \\ U_D \end{pmatrix} = \begin{pmatrix} f-K_{ID}U_0 \\ U_0 \end{pmatrix} .$$

What corresponds to the following updates to make on the matrix and the second member:

$$\begin{array}{ccc} F_I & \leftarrow & f - K_{ID} U_0 \\ F_D & \leftarrow & U_0 \\ K_{DD} & \leftarrow & Id_{DD} \\ K_{ID} & \leftarrow & 0_{ID} \\ K_{DI} & \leftarrow & 0_{DI} \end{array} .$$

4.2 Descriptors of the matrix

Total matrix

There is no degree of freedom of Lagrange: the table `.NUMÉRIQUE.DELG` is filled of 0. The table `.NUMÉRIQUE.DEEQ` is of size $2\mathrm{N}=36$. Here is its contents (1[era] line) and its interpretation (2[eme] line):

1	1	1	2	2	1	2	2	...	9	1	9	2
N1	DX	N1	DY	N2	DX	N2	DY	...	N9	DX	N9	DY

The collection `.PRNO` only one object has: the only degrees of freedom are the physical degrees of freedom associated with the nodes of the grid. Here is its contents node by node:

```
 1    2    6    0    0    0    0    # N1 node
 3    2    6    0    0    0    0    # node N2
 5    2    6    0    0    0    0    # N3 node
 7    2    6    0    0    0    0    # N4 node
 9    2    6    0    0    0    0    # N5 node
11    2    6    0    0    0    0    # N6 node
13    2    6    0    0    0    0    # N7 node
15    2    6    0    0    0    0    # N8 node
17    2    6    0    0    0    0    # N9 node
```

Thus the first equation posed on the node N3 grid carries the number of equation 5. There are 2 equations posed on this node. To code the components of the size `DEPL_R` one uses a vector of size 5. This vector has like values $(6\ 0\ 0\ 0\ 0)$.

Only the first term is nonnull: it is worth $6=2^1+2^2$.

That which means that the selected components of the size are the two first in the order of the catalogue, i.e. components `DX` and `DY`.

4.2.1 Distributed matrix

One defers then the values of `.NUML` for processors 0 and 1.

	P0		P1	
.NUML.NEQU	10		16	
.NUML.NULG	il	ig	il	ig
	1	15	1	3
	2	16	2	4
	3	3	3	9
	4	4	4	10
	5	7	5	17
	6	8	6	18
	7	17	7	15
	8	18	8	16
	9	13	9	1
	10	14	10	2
			11	11
			12	12
			13	5
			14	6
			15	13
			16	14

The opposite correspondence is given by the table .NUML.NUGL :

	P0		P1	
.NUML.NUGL	ig	il	ig	il
	1	0	1	9
	2	0	2	10
	3	3	3	1
	4	4	4	2
	5	0	5	13
	6	0	6	14
	7	5	7	0
	8	6	8	0
	9	0	9	3
	10	0	10	4
	11	0	11	11
	12	0	12	12
	13	9	13	15
	14	10	14	16
	15	1	15	7
	16	2	16	8
	17	7	17	5
	18	8	18	6

Thanks to .DEEQ one can find which nodes are supports of the degrees of freedom on each processor:

	P0			P1		
.NUML.NEQU	10			16		
	il	*ig*	Noeud	*il*	*ig*	Noeud
.NUML.NULG	1	15	8	1	3	2
	2	16	8	2	4	2
	3	3	2	3	9	5
	4	4	2	4	10	5
	5	7	4	5	17	9
	6	8	4	6	18	9
	7	17	9	7	15	8
	8	18	9	8	16	8
	9	13	7	9	1	1
	10	14	7	10	2	1
				11	11	6
				12	12	6
				13	5	3
				14	6	3
				15	13	7
				16	14	7

It is seen that the components on the same node are numbered in way contiguous.

Notice :

When the matrix is distributed, the equations of blocking kinematics are multiplied by the number of processors used for calculation. The constraint $U=U_0$ is replaced on p processors by $pU=pU_0$. In the routine `asmchc`, one puts of the 1 on the diagonal of the matrix and this operation is carried out on all the processors. These terms accumulate in the total matrix.

5 The linear system for modeling B

The limiting conditions are dualisées. The method of (double) dualisation is described in documentation [R3.03.01]. One specifies the discrete system in the paragraph 5.1 and the value of the descriptors of the matrix in the paragraph 5.2.

5.1 The discrete system

The condition limits on Γ_D is imposed like a constraint on displacements, by multipliers of Lagrange. One notes:

- U "physical" degrees of freedom;
- Λ multipliers of Lagrange.

One solves an increased linear system, form:

$$\begin{pmatrix} K_{UU} & C_{U\Lambda}^T \\ C_{\Lambda U} & D_{\Lambda\Lambda} \end{pmatrix} \begin{pmatrix} U \\ \Lambda \end{pmatrix} = \begin{pmatrix} f \\ u_0 \end{pmatrix}$$

where $C_{\Lambda U}$ is the matrix of the constraints.

5.2 Descriptors of the assembled matrix

5.2.1 Total matrix

The number of equations of the problem is $N = .\mathrm{NUME.NEQU} = 30$.

Calculation of the full number of degrees of freedom :
There are 9 nodes and each node carries the components DX and DY size DEPL_R, that is to say on the whole 18 equations relating to 18 "physical" degrees of freedom.
With these "physical" degrees of freedom, one adds degrees of freedom of the type "Lagrange" which make it possible to take into account by dualisation the limiting condition of embedding. There are 3 nodes embedded in the problem. On each embedded node, one blocks two components DX and DY. For each blocked component, one uses 2 multipliers of Lagrange. On the whole, one thus defines for this problem $3\times2\times2=12$ degrees of freedom of the Lagrange type.

Values of the NUME_EQUA:
In .PRNO, the first object describes the nodes of the grid, which carry degrees of physical freedom (as in the modelisation A):

```
3    2   6   0   0   0   0   # node 1
7    2   6   0   0   0   0   # node 2
11   2   6   0   0   0   0   # node 3
15   2   6   0   0   0   0   # node 4
17   2   6   0   0   0   0   # node 5
21   2   6   0   0   0   0   # node 6
25   2   6   0   0   0   0   # node 7
27   2   6   0   0   0   0   # node 8
29   2   6   0   0   0   0   # node 9
```

The second describes embedding:

```
1    1   134217728   0   0   0   0
5    1   134217728   0   0   0   0
2    1   134217728   0   0   0   0
6    1   134217728   0   0   0   0
9    1   134217728   0   0   0   0
13   1   134217728   0   0   0   0
10   1   134217728   0   0   0   0
14   1   134217728   0   0   0   0
19   1   134217728   0   0   0   0
23   1   134217728   0   0   0   0
20   1   134217728   0   0   0   0
24   1   134217728   0   0   0   0
```

Each node carries only one component which is determined by the entirety $134217728=2^{27}$, it is thus the component LAGR in question (multiplier of Lagrange).
.DEEQ give for each equation the node support of the degree of freedom and the component of the size. One displays his values in the following table. When the degree of freedom is a physical degree of freedom, (DX, DY) indicate which component of the size it discretizes. For the degrees of freedom of the Lagrange type, (Lag DX,Lag DY) indicate which component is blocked by the multiplier of Lagrange.

.NUME.DEEQ			
	ino	*icmp*	Commentaire
1	1	−2	Lag DY
2	1	−1	Lag DX
3	1	1	DX
4	1	2	DY
5	1	−2	Lag DY
6	1	−1	Lag DX
7	2	1	DX
8	2	2	DY
9	3	−2	Lag DY
10	3	−1	Lag DX
11	3	1	DX
12	3	2	DY
13	3	−2	Lag DY
14	3	−1	Lag DX
15	4	1	DX
16	4	2	DY
17	5	1	DX
18	5	2	DY
19	6	−2	Lag DY
20	6	−1	Lag DX
21	6	1	DX
22	6	2	DY
23	6	−2	Lag DY
24	6	−1	Lag DX
25	7	1	DX
26	7	2	DY
27	8	1	DX
28	8	2	DY
29	9	1	DX
30	9	2	DY

Note:

The two degrees of freedom of Lagrange which make it possible to block a component on a physical degree of freedom must be numbered in order to frame the physical degree of freedom: one must be before and the other afterwards. In Code_Aster, the algorithm of classification ensures this constraint but the degrees of freedom of Lagrange are not inevitably numbered with more close to the blocked component. It is the case on this example but it is not an obligation.

5.2.2 Distributed matrix

The partition is the same one as in Modelisation A. The degrees of freedom of Lagrange all are assigned to processor 0. One describes the classification of the matrix in the following table:

	P0			P1		
.NUML.NEQU	28			16		
		NULG	DELG		NULG	DELG
	il	ig		il	ig	
	1	27	0	1	7	0
	2	28	0	2	8	0
	3	7	0	3	17	0
	4	8	0	4	18	0
	5	15	0	5	29	0
	6	16	0	6	30	0
	7	29	0	7	27	0
	8	30	0	8	28	0
	9	25	0	9	3	0
	10	26	0	10	4	0
	11	3	0	11	21	0
	12	4	0	12	22	0
	13	1	−1	13	11	0
	14	5	−2	14	12	0
	15	11	0	15	25	0
	16	12	0	16	26	0
	17	9	−1			
	18	13	−2			
	19	21	0			
	20	22	0			
	21	19	−1			
	22	23	−2			
	23	2	−1			
	24	6	−2			
	25	10	−1			
	26	14	−2			
	27	20	−1			
	28	24	−2			

In this table, value -1 indicates that it is about Lagrange 1 and the value -2 which it is about Lagrange 2.

Notice :

.DEEQ also allows to know about which degrees of freedom are multipliers of Lagrange, but not if it is about Lagrange 1 or 2. Value -1 indicates that one blocks the component DX and the value -2 which one blocks the component DY . To rebuild all information, the two objects should be used (.DELG and .DEEQ).

6 Construction of the PETSc matrix

6.1 Data model PETSc

PETSc proposes a kind of data Chechmate. This kind of data supposes, in parallel, that the matrix is distributed on the processors available by group of contiguous lines.

The matrix thus should be redistributed *Code_Aster*. One builds a new total classification for PETSc. In this classification, a degree of freedom which does not belong that with a processor is affected with this processor and a degree of freedom divided by several processors belongs to the processor of lower row.

Notice :

PETSc stores the matrix in entirety (not symmetrical storage).
PETSc uses convention C of subscripting of the tables (of 0 in N-1 for a table of size NR).

6.2 Descriptor of matrix for modeling B

The correspondence between the local classification of the matrix and its total classification for PETSc is given by the table NUML.NLGP. One indicates in the table according to three classifications for modeling B: i_l is the local index, i_g the pure total index classification *Code_Aster* and i_{gp} the total index for PETSc.

	P0			P1		
.NUML.NEQU	28			16		
		NULG	NLGP		NULG	NLGP
	il	*ig*	*igp*	*il*	*ig*	*igp*
	1	27	1	1	7	3
	2	28	2	2	8	4
	3	7	3	3	17	29
	4	8	4	4	18	30
	5	15	5	5	29	7
	6	16	6	6	30	8
	7	29	7	7	27	1
	8	30	8	8	28	2
	9	25	9	9	3	11
	10	26	10	10	4	12
	11	3	11	11	21	19
	12	4	12	12	22	20
	13	1	13	13	11	15
	14	5	14	14	12	16
	15	11	15	15	25	9
	16	12	16	16	26	10
	17	9	17			
	18	13	18			
	19	21	19			
	20	22	20			
	21	19	21			
	22	23	22			
	23	2	23			
	24	6	24			
	25	10	25			
	26	14	26			
	27	20	27			
	28	24	28			