PERF013 – Performances of a modal calculation in parallel

Summary:

The objective of this CAS-test is to measure the performances in parallel mode of a standard modal calculation. For the moment, only the stages of construction of the matrices and, especially, of resolution of the linear systems associated are treated in parallel. It is this last stage which dimensions the costs in time and RAM memory of modal calculation. It is paralleled here *via* linear solvor MUMPS [U2.08.06]. The savings of time and memory are interesting and comparable to the results presented by the other codes general practitioners in mechanics of the structures.

To improve these performances, it will also be necessary to parallel the modal solvor including who controls the resolutions of systems linear.

Modal calculation used takes again modeling C of the CAS-test *perf003*: square plate with a grid in elements of hulls, linear finite elements, search for 10 clean modes *via* the operator CALC_MODES and method of Sorensen.

1 Problem of reference

See the CAS-test perf003c.

2 Reference solution

See the CAS-test perf003c.

3 Modeling A

3.1 Characteristics of modeling A

Number of processor: 1

One uses modeling C of the CAS-test perf0013: square plate with a grid in elements of hulls, linear finite elements, search for 10 clean modes *via* the operator CALC_MODES and method of Sorensen.

Characteristics of the grid: 167,281 NODE, 1,632 SEG2 and 166,464 QUAD4.
Many degrees of freedom : 1,013,490.

Dimensioned linear solvor, one uses external product MUMPS while seeking to privilege consumption in time (with the detriment *a priori* those in memory). One thus uses MUMPS in mode In-Core (GESTION_MEMOIRE= ' IN_CORE').

An important point is that it is necessary here to impose a basic renumerotor on MUMPS (RENUM=' QAMD' for example) rather than to let to him choose a renumerotor more sophisticated (RENUM=' AUTO' by default). Because if not the phase of analysis of MUMPS, which is purely sequential, becomes exaggeratedly large and parallelism does not bring any more profit in term of spent time, only in term of memory (cf notices Doc. [U2.08.03] associated with the keyword RENUM).

3.2 Results

Size	Reference
FREQ n°12	993.5

4 Modeling B

4.1 Characteristics of modeling B

Identical to modeling A but calculation is carried out here on 4 processors.

4.2 Results

Size	Reference
FREQ n°12	993.5

5 Modeling C

5.1 Characteristics of modeling C

Identical to modeling A but with a uniformly refined grid *via* the software LOBSTER (one SEG2 is crossed into 2 and one QUAD4 in 4). Calculation on 1 processor.

Characteristics of the grid: 667,489 NODE, 3,264 SEG2 and 665,856 QUAD4.
Many degrees of freedom : 4,024,530.

5.2 Results

Size	Reference
FREQ n°12	993.5

6 Modeling D

6.1 Characteristics of modeling D

Identical to modeling C but calculation is carried out here on 4 processors.

6.2 Results

Size	Reference
FREQ n°12	993.5

7 Modeling E

7.1 Characteristics of modeling E

Identical to modeling C but calculation is carried out here on 16 processors.

7.2 Results

Size	Reference
FREQ n°12	993.5

8 Modeling F

8.1 Characteristics of modeling F

Identical to modeling C but calculation is carried out here while seeking to privilege consumption in RAM memory (with the detriment *a priori* those in time). One thus uses MUMPS in Out-Of-Core mode (GESTION_MEMOIRE='OUT_OF_CORE').

Calculation is carried out on 1 processor.

8.2 Results

Size	Reference
FREQ n°12	993.5

9 Summary of the results

Machine	Modeling (no. of procs)	RAM memory (out of Mo)				Time ELAPSED (only CALC_MODES) (in S)				
		Allocated	Used by…			Linear Solvor MUMPS				Total operator
Linux 64 bits "aster4" V10,01			MUMPS	Aster (JEVEUX)	VmPeak	Analysis	Factorization	Solve	Total Solvor	Total operator
Idem	With (1)	2500	1811	1920	3910	10	182	31	223	254
Idem	B (4)	1500	676	1411	2219	14	63	17	94	124
Idem	C (1)	11000	8780	7590	16227	43	1469	147	1658	1786
Idem	D (4)	8000	3700	7590	10209	51	520	76	648	784
Idem	E (16)	8000	1270	7590	8887	59	367	140	456	576
Idem	F (1)	3000	1524	2998	6676	230	894	502	1625	1779

On the aspects purely linear solvor/parallelism, one notes within sight of these results:

- Into sequential, with the parameter setting by default ("priority time") consumption memory of calculation is dictated by that of MUMPS. It is not true any more when one parallels would be this only on 4 processors or if one activates the options of the linear solvor limiting consumption memories ("priority RAM memory", modeling F). Moreover these last always do not burden the total performances in time elapsed, certain losses (descent-increase of the stage of "solve" slowed down by the OOC) being sometimes compensated by spectacular profits on factorization (like here between C and F *via* change of renumerotor).
- Theoretical speed-UPS of parallelization on 4 procs of modeling A are of 2.7 in time and 4.0 in RAM. With modeling B one finds speed-UPS manpower, respectively, 2.0 and 2.7. Theoretical speed-UPS of parallelization on 4/16 procs of modeling C are of 3.0/6.4 in time and 4.0/16.0 in RAM. With modelings D and E, one finds speed-UPS manpower, respectively, 2.3/3.1 and 2.4/6.9. These accelerations are correct and conform to those observed with the other codes general practitioners in mechanics of the structures.

Therefore, the fact of parameterizing the aspects linear solvor finely and/or of paralleling a standard modal calculation can allow significant profits (up to 80%) in time elapsed and RAM memory.

However, these profits tend to reach a maximum starting from ten processors. To improve these performances, it will be necessary to extend the "parallel perimeter" of the code and thus to also parallel the modal solvor including (who controls the resolutions of systems linear).