PERF013 – Performances of a modal computation in parallel

Summarized:

The purpose of this benchmark is to measure the performances in parallel mode of a standard modal computation. For time, only the stages of construction of the matrixes 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 computation. It is paralleled here *via* the linear solver 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 structural mechanics.

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

The computation modal used takes again the modelization C perf003 *benchmark*: square plate with a grid in shell elements, linear finite elements, search for 10 eigen modes *via* operator MODE_ITER_SIMULT and the method of Sorensen.

1 Problem reference

See the perf003c benchmark.

2 Reference solution

See the perf003c benchmark.

3 Modelization A

3.1 Characteristic of the modelization A

Number of processor: 1

One uses the modelization C perf0013 benchmark: square plate with a grid in shell elements, linear finite elements, search for 10 eigen modes *via* operator MODE_ITER_SIMULT and the method of Sorensen.

Characteristics of mesh: 167.281 NOEUD, 1.632 SEG2 and 166.464 QUAD4.
Many degrees of freedom : 1.013.490.

Dimensioned linear solver, one uses the external product MUMPS while seeking to privilege consumption in time (with the detriment *a priori* of 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 rather impose a basic renumerotor on MUMPS (RENUM=' QAMD' for example) than to let to him choose a renumerotor more sophisticated (RENUM=' AUTO' by default). Because if not the phase of analysis of MUMPS, which are purely sequential, becomes exaggeratedly large and parallelism does not bring any more gain in term of consumed time, only in term of memory (cf notices Doc. [U2.08.03] to the key word associated RENUM).

3.2 Results

Quantity	Reference
FREQ n°12	993.5

4 Modelization B

4.1 Characteristic of the modelization B

Identical to the modelization A but computation is carried out here on 4 processors.

4.2 Results

Quantity	Reference
FREQ n°12	993.5

5 Modelization C

5.1 Characteristic of the modelization C

Identical to the modelization A but with a mesh refined uniformly *via* the software HOMARD (a SEG2 is cut into 2 and one QUAD4 into 4). Computation on 1 processor.

Characteristics of mesh: 667.489 NOEUD, 3.264 SEG2 and 665.856 QUAD4.
Many degrees of freedom : 4.024.530.

5.2 Results

Quantity	Reference
FREQ n°12	993.5

6 Modelization D

6.1 Characteristic of the modelization D

Identical to the modelization C but computation is carried out here on 4 processors.

6.2 Results

Quantity	Reference
FREQ n°12	993.5

7 Modelization E

7.1 Characteristic of the modelization E

Identical to the modelization C but computation is carried out here on 16 processors.

7.2 Results

Quantity	Reference
FREQ n°12	993.5

8 Modelization F

8.1 Characteristic of the modelization F

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

The computation on 1 processor is carried out.

8.2 Results

Quantity	Reference
FREQ n°12	993.5

9 Summary of the results

Machine	Modelization (nbre of procs)	Memory RAM (out of Mo)				Time ELAPSED (only MODE_ITER_SIMULT) (in S)				
		Allocated	Used by…			Total				Linear solver MUMPS operator
Linux 64 bits "aster4" V10,01			MUMPS	Aster (JEVEUX)	VmPeak	Idem	Analyze	Total	Solve Factorization	Total Solver
operator	A (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 solver/parallelism, one notes within sight of these results:

- Into sequential, with the parameter setting by default ("priority time") the memory consumption of computation 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 solver limiting consumption memories ("priority memory RAM", modelization 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 gains on factorization (like here enters C and F *via* the change of renumerotor).
- Theoretical speed-UPS of parallelization on 4 procs of the modelization A are of 2.7 in time and 4.0 in RAM. With the modelization B one finds speed-UPS manpower, respectively, 2.0 and 2.7. Theoretical speed-UPS of parallelization on 4/16 procs of the modelization C are of 3.0/6.4 in time and 4.0/16.0 in RAM. With the modelizations 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 structural mechanics.

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

However, these gains tend to reach a maximum 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 solver including (who controls the resolutions of systems linear).