Thermomechanical and hydraulic industrial simulations using MUMPS at EDF

MUMPS User group meeting
15 april 2010
O.Boiteau, C.Denis, F.Zaoui

MUltifrontal Massively Parallel sparse direct Solver
1a. EDF Group: a European Electricity Utility with strong R&D involvement

EDF Energy
- Capacity: 101 GW (63 GW nuclear)
- Customers: 28 Million
- Networks: 1,340,000 km
- Gas: 3 Gm³

EnBW
- Sales: €64.3 billion
- Global customer base: 38.1 million
- Employees worldwide: 161,000
- Installed capacity: 128.2 GW
- R&D: €1 million/day

EDF
- Capacity: 101 GW (63 GW nuclear)
- Customers: 28 Million
- Networks: 1,340,000 km
- Gas: 3 Gm³

Edison
1b. Operation, Maintenance & Optimization of complex systems at EDF

Permanent objectives

• guarantee safety,
• improve performances/costs,
• maintain assets.

Changing operating conditions

• face unexpected events, ageing issues, maintenance,
• improve performance through new technologies, new operating modes and system-wide optimization,
• adapt to evolving set of rules (safety, environment, regulatory).

In-house technical backing

• expertise: strong Engineering and R&D Divisions,
• physical testing and simulation are key tools from the outset.

Software Quality Plan (requirements of the Nuclear Structures Safety Authority)
1c. Workflow of EDF physical simulation codes

- Mesh generation, material, loadings…
- Data distribution
- Discretization, time, space
- Linear solver
- Post-processing, visualization
- CFD code
- Thermomechanical code
- Thermics/Neutronics for nuclear
- Electromagnetism
- SYRTHES
- Cocagne
- Code_Carmel
1d. Use of MUMPS in two EDF physical simulation codes

**Code_Aster:** A finite element code for analysis of structures and thermomechanics studies and researches ([www.code-aster.org](http://www.code-aster.org)).

**TELEMAC system:** a group of numerical modeling softwares for free surface water, sedimentology, waves, water quality, underground flows ([www.telemacsystem.com](http://www.telemacsystem.com)) …
2a. The bottleneck is the linear system step

- Direct methods versus iterative ones

\[ \begin{align*}
L D L^T v &= f \\
L w &= f \\
D v &= w \\
L^T u &= v
\end{align*} \]

\[ Ku = f \]

Aster keyword METHODE

- LDLT
- GCPC
- MUMPS
- PETSC
- MULT_FRONT
- FETI
2b. **Code_Aster & MUMPS (1/3)**

A story of sparse linear system!

- (Usually) General real symmetric < 5 millions dof;

- Flexibility/Efficiency, Distributed parallelism, Out-of-core capability

---

**Assembly of the FE matrix**

- Centralized
- Distributed

**Analysis**

- Factorization
- Solve

**Factorization**

- $K^{-1}$

**Solve**

- $u$

**Data F77**

- Aster

**Data F90**

- MUMPS

**RAM**

**Disk**

---

**Assembly of the FE matrix**

- CHAM_ELEM
- MATR_ELEM/VECT_ELEM
- NUME_DDL/CHAM_NO/MATR_ASSE

**Factorization**

- $K^{-1}$

**Solve**

- $u$

**IC**

- $K^{-1}$

**OOC**

- $K^{-1}$

---

- IRN(_loc), JCN(_loc), A(_loc) et RHS
- $f$, $f$, $K$, $K$ et $f$
2b. Code_Aster & MUMPS (2/3)
A story of sparse linear system!

✓ (Often) **Bad conditionning** \((10^8)\) and indefinite matrix (mixte FE, Lagrange multipliers, X-FEM...).

- **X-FEM on a pipe**: enriching the sane mesh with special FE to simulate multi-cracking.

- Detection of **singular matrix** (lacks/excess of boundary conditions, eigenvalue problem, null space analysis...).

- (Sometimes) Unsymmetric, SDP, complex arithmetic, reuse of the analysis phase for several solves.

Numerical robustness, Pivoting/scaling strategies, Error analysis and iterative refinement.

Zero-pivot detection option

Full range of possibilities
Mixte-precision strategies:

- Direct solver in non-linear analysis with Newton-like algorithm,
- Krylov solver (linear or not): coarse/cheap/robust preconditioner.

Various kinds of linear systems:

- One-shot resolution,
- (Often) **Multiples right-hand sides** (Newton with periodic reactualization of the tangent matrix...),
- (Sometimes) Concurrent resolutions (Schur complement-like solves in contact-friction problems...).
2c. Feedbacks of the software integration/use

- More than 100 Aster test-cases (seq. and //) using MUMPS, dozens of MUMPS parameters available to the Aster’User.

- **Steady software worknings in the Aster/MUMPS’ links:** bug tracking, optimization, upgrade, user training…

  Often questioning/debugging about exterior librairies induce improvement in the caller code (data workflow…)

<table>
<thead>
<tr>
<th>Year</th>
<th>2006</th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td># Works about Code_Aster/MUMPS</td>
<td>28</td>
<td>36</td>
<td>56</td>
<td>103</td>
</tr>
</tbody>
</table>

- Daily use throught Code_Aster at EDF R&D/Engineering
2d. Some results (1/3)

\( N=0.8 \text{ M} \)
\( \text{nnz}=4.5 \text{ M} / K^{\text{fact}}=372\text{M} \)

(77%) Th. speed-up 4/16/32 procs:
2.4/ 3.6/ 3.9

Real speed-up 4/16/32 procs:
2.1/2.0 3.6/3.6 3.9/3.9

Good Speed-ups

\( N=0.8 \text{ M} \)
\( \text{nnz}=71 \text{ M} / K^{\text{fact}}=1900\text{M} \)

(95%) Th. speed-up 4/16/32 procs:
3.5/ 9.1/ 12.5

Real speed-up 4/16/32 procs:
2.4/2.7 4.3/ 6.7 7.7/10.3

OpenMP needed
2d. Some results (2/3)

- **RAM memory consumption**
  - **Unbalance RAM** memory consumption between cores
    - x4 on MUMPS OOC
    - x6 on Aster memory

Significant improvements
2e. Conclusions regarding *Code_Aster*/MUMPS

- **Daily use** through *Code_Aster* at EDF R&D/Engineering (‘best-in-class’ tool)
  
  Much more important than performances, we particularly appreciate the MUMPS Software Quality and the reactivity/friendliness of its team.

- **Partnership** through the ANR SOSLTICE

- **Wish for future MUMPS functionalities/Letter to Santa Claus**
  - Hybrid parallelism (MPI/Threads),
  - OOC capability (analyse step, integer, automatic),
  - Reuse of the factorized Matrix between two runs (restart mode),
  - Parallel Incomplete factorization…

- **Test and benchmark of others strategies/libraries**: DD, multigrid, PastiX, HIPS/MaPhys…
3. TELEMAC : an Integrated Modelling System

Free Surface Hydrodynamics
TELEMAC2D – TELEMAC3D

Water Quality
(coupled) TELEMAC

Groundwater Flows
ESTEL2D – ESTEL3D

Sedimentology
SISYPHE – TELEMAC3D

Waves
ARTEMIS – TOMAWAC

Smoothed Particle Hydrodynamics
SPARTACUS
3. First tests of MUMPS in TELEMAC (context)

Telemac has a common library of parallel iterative solvers developed at EDF + 1 direct sequential solver (YSMP) recently included.

7 // iterative solvers :
- developed and maintained at EDF
- very good performances in most cases
- but fail to converge with ARTEMIS!

→ YSMP :
- works with ARTEMIS
- limitation on the problem size
- robustness not so good
- no parallelism

MUMPS in comparison?

MUMPS in replacement?
3. First tests of MUMPS in TELEMAC (description)

SEQUENTIAL TESTS (PC Linux Workstation) :
- **ARTEMIS** (MUMPS vs YSMP) : *Mild Slope equation (FEM)*
- **TELEMAC2D** (MUMPS vs Iterative solvers) : *Shallow Water (2D FEM)*

PARALLEL TESTS (HP supercomputer) :
- **ARTEMIS** (MUMPS)
- **TELEMAC3D** (MUMPS vs Iterative solvers) : *Navier-Stokes (3D FEM)*
3.a Sequential Test: ARTEMIS

MUMPS in L.D.L$^t$ mode with 2 systems to solve, 3-6 iterations

MUMPS is about 50% faster than YSMP (N ~ 100,000)

There is no more problem of robustness

As expected, MUMPS easily outperforms YSMP

Example:
N = 338,930
NNZ(upper) = 2,532,299

Ordering  NNZ(L+U-I)

<table>
<thead>
<tr>
<th>Ordering</th>
<th>NNZ(L+U-I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pord</td>
<td>26M</td>
</tr>
<tr>
<td>Scotch</td>
<td>27M</td>
</tr>
<tr>
<td>Metis</td>
<td>27M</td>
</tr>
<tr>
<td>Amf</td>
<td>29M</td>
</tr>
<tr>
<td>Amd</td>
<td>31M</td>
</tr>
<tr>
<td>Qamd</td>
<td>31M</td>
</tr>
</tbody>
</table>

$T_{MUMPS} = 9s$
3. b Sequential Test: TELEMAC2D

Simulation of a dam break: Malpasset (1959)

Simulation of 1000 s with DT = 1 s

(L.D.L^T and systematic analysis for MUMPS) \( N = 153,253 \) NNZ = 1M

Global computation times (for the same precision on results ;)

Iterative: 19’33”
YSMP: 47’02”
MUMPS: 60’44”

Improvement:
No systematic analysis or suppress zeros?

/2 ?
3.c Parallel Test : ARTEMIS

Case Flamanville

(12 = 6x2 sparse linear systems to be solved with N= 169 465)

Experiments performed on HP supercomputer

MUMPS used in distributed mode (icntl(18)=3) double precision

METIS (sequential) used as reordering method

Remember : iterative methods do not converge !
3.c Parallel Test : ARTEMIS

ARTEMIS // using MUMPS //
(C. Denis)

MUMPS can now be used to deal with larger ARTEMIS problems!
3.0 Parallel Test: TELEMAC3D

Evolution of the salinity in the Berre Lagoon (South of France)

- Mediterranean
- Berre lagoon
- Caronte canal
- Boundary condition at Hydroelectric power plant

Vertical Mesh
One time step, 4 sparse linear systems need to be solved

sparse linear system $S_1$, $N=4\,098\,700$, Number of entries in factors $\sim 1,7\times 10^9$

sparse linear system $S_2$, $N=204\,935$, Number of entries in factors $\sim 8,5\times 10^6$

sparse linear system $S_3$, $N=4\,098\,700$, Number of entries in factors $\sim 1,7\times 10^9$

sparse linear system $S_4$, $N=4\,098\,700$, Number of entries in factors $\sim 1,7\times 10^9$

MUMPS// used in distributed mode (icntl(18)=3)

Scotch (sequential) used as reordering method

Experiments are performed on a HP Cluster on 32, 64 and 128 processors

Comparison are made with the iterative methods //

Iterative methods for this problem require few iterations to converge
### 3.d Parallel Test: TELEMAC3D

#### TELEMAC 3D// using MUMPS//

**C. Denis**

![Bar chart showing computational times for MUMPS](chart1.png)

- **Global computing times in s**
- **Computing times to solve S1**
- **Computing times to solve S2**

**Number of processors:**
- 32
- 64
- 128
- 256

#### TELEMAC 3D// using TELEMAC iterative methods

**C. Denis**

![Bar chart showing computational times for iterative methods](chart2.png)

- **Global computing times in s**
- **Computing times to solve S1**
- **Computing times to solve S2**

**Number of processors:**
- 32
- 64
- 128
- 256

---

**Ite. Method**

---

**MUMPS**
The solve phase is dominated by MUMPS algorithm.

Computing times to solve S1 with 256 procs
(C. Denis)
Precision on results are identical...

*The numerical scheme has to be conservative in terms of water mass*

<table>
<thead>
<tr>
<th>Loss of (water) mass</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>with MUMPS</td>
<td>-0.2698488E-05</td>
<td>0.1625352E-06</td>
<td>-0.1430511E-05</td>
<td>0.1625688E-06</td>
</tr>
<tr>
<td>with iterative methods</td>
<td>-0.2698488E-05</td>
<td>0.1625352E-06</td>
<td>-0.1430511E-05</td>
<td>0.1625688E-06</td>
</tr>
</tbody>
</table>
MUMPS and iterative methods are both useful depending of the sparse linear system to solve

Very useful when the sparse linear system to be solved is not well conditioned (ARTEMIS)

Not surprisingly, the conjugate gradient method gives best performances than MUMPS// when it needs a few number of iterations to converge for a well-posed problem (TELEMAC3D)

Future works:

• To improve the performance of ARTEMIS//
  • Optimisation of the matrix building by using MUMPS with complex numbers
  • solve in sequential the local sparse linear system with MUMPS and solve the interface problem
  • Implementation of MaPhys or HIPS in the TELEMAC system