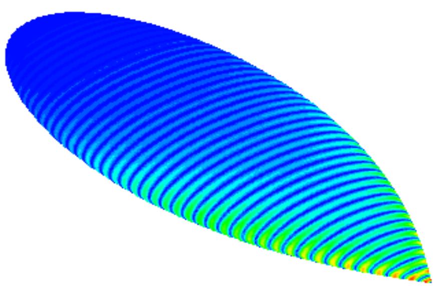
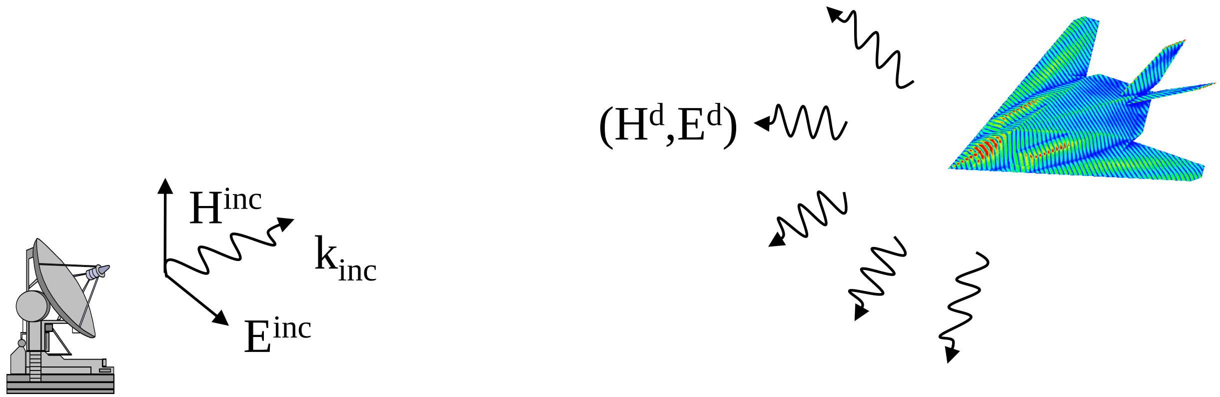
The High Performance Solution of Sparse Linear Systems and its application to large 3D Electromagnetic Problems



David GOUDIN
CEA/DAM/CESTA

What is the problem ?

The simulation of the electromagnetic behavior for a complex target in the frequency domain formulation

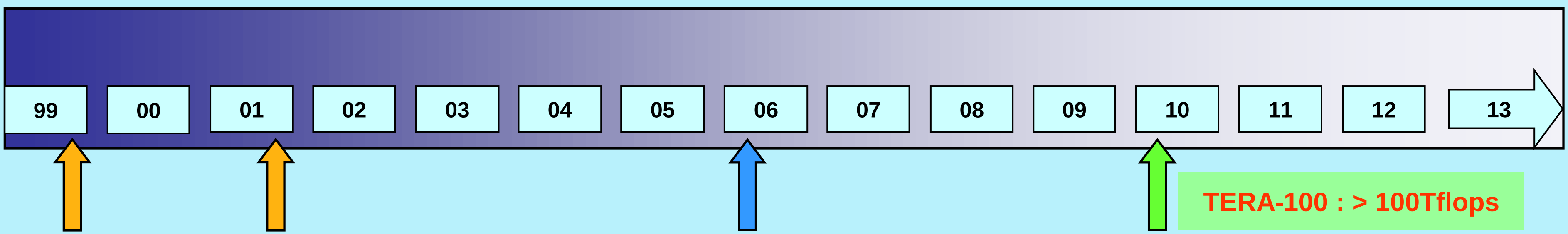


⇨ Numerical solution of Maxwell’s equations in the free space

- ☐ RCS (Radar Cross Section) computation
- ☐ Antenna computation

Our Supercomputers

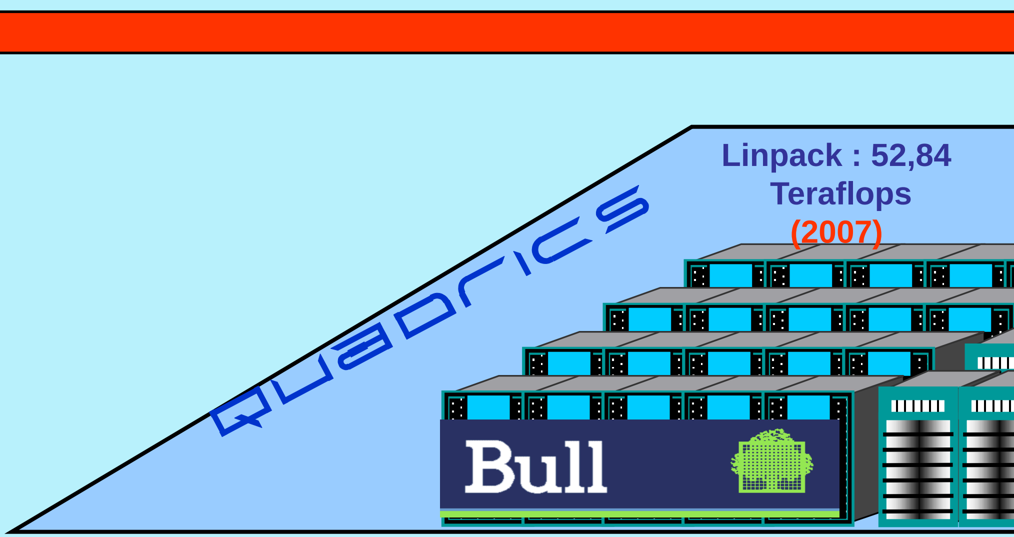
TERA



30 (50) Gflops

TERA-1 : 1 (5) Tflops

TERA-10 : > 10 Tfops



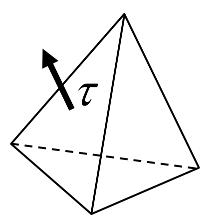
Main characteristics of TERA 10

SMP Nodes	544*16 proc.
Peak performance (Benchmark CEA)	> 60 Tflops (12,5 Tflops)
RAM	30 To
Disk space	1 Po - 56 nodes
Consumption	< 2000 kW

Partial Differential Equations Methods = PDE

- Volumic description
- the unknowns : fields in the computational volume : *E* and/or *H*
- formulations
  - ➔ *2nd order on E*
  - ➔ *2nd order on H*

➡ Discretization by volumic finite elements H(rot)



Boundary Integral Equations Methods = BIEM

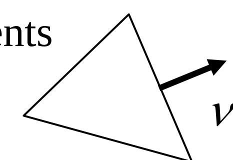
- surfacic description
- the unknowns : equivalent currents on the surface : *J* et *M*
- formulations
  - ➔ *EFIE*
  - ➔ *CFIE*
  - ➔ *EID (CEA originality)*

➡ Integral representation :

$$E^d = i\omega\mu L_\omega J - \frac{1}{i\omega\varepsilon} grad\, L_\omega div_\Gamma J - ro$$

$$H^d = i\omega\varepsilon L_\omega M - \frac{1}{i\omega\mu} grad\, L_\omega div_\Gamma M$$

➡ Discretization by surfacic finite elements H(div)



➡ **IN BOTH CASE it leads to solve a linear system *A.x = b with A dense or sparse***

□ **Fully BIEM**

□ **Meshes at the surface and on interfaces between homogeneous isotropic medium**

- number of DOF (Degrees of Freedom) reasonable
- leads to a full matrix

$$\left(\begin{array}{c|c} A^s_{22} & A^s_{23} \\ \hline A^s_{23} & A^s_{33} \end{array}\right) \left(\begin{array}{c} M \\ \hline J \end{array}\right)$$

⇨ Parallelization : *very efficient*

□ **Solver**

In house parallel Cholesky-Crout solver.

The matrix is :

- symmetric
- complex
- non hermitian

For some applications, the matrices are non symmetric so we use

- The ScaLapack LU solver

Some results with the 3D BIEM code

Complete geometry and plane symmetries

- Parallel Direct solver for dense systems, the matrix is complex and symmetric

N=486 636, size 1895 Gbytes, LDLt factorization

Number of Proc	CPU time (s)	% / peak power	Tflops
1024	39 642	60 %	3.87

Unvarious geometry under rotation

- Parallel Direct solver for dense systems, the matrix is complex and non symmetric (ScaLapack)

N=75 000, size 90 Gbytes, LU factorization

Number of Proc	CPU time (s)	% / peak power
128	1809	76 %
256	968	70%
512	507	68%

N=285 621, size 1306 Gbytes, LU factorization

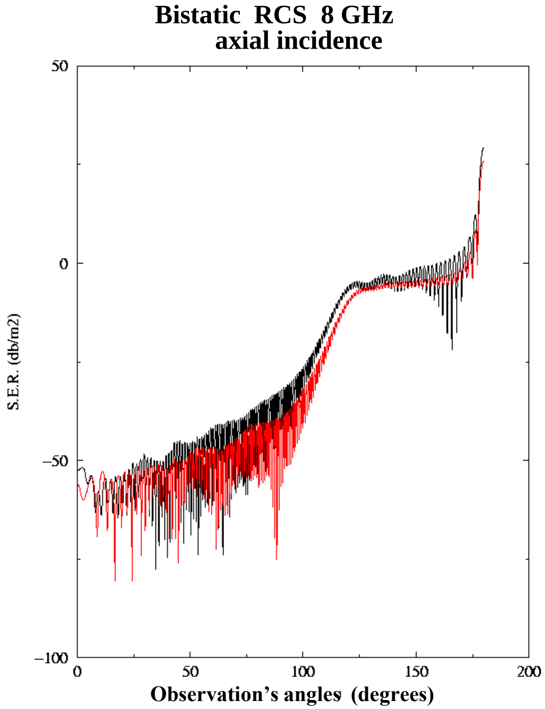
Number of proc	CPU time (s)	% / peak power	Tflops
1600	8315	73 %	7.47

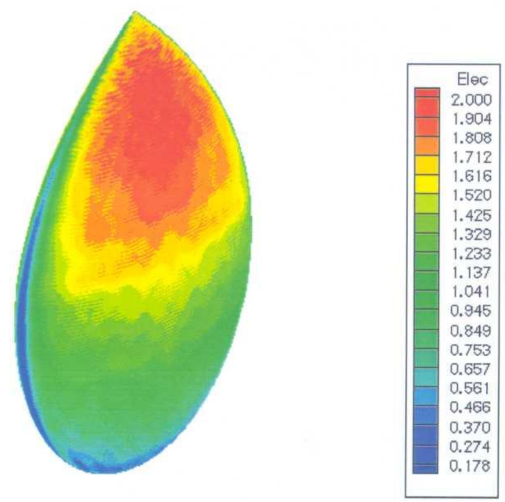
A test case for the 3D BIEM code

- **NASA Almond** *8 Ghz*

  233 027 unknowns with 2 symmetries ⇒ 932 108 unknowns

  - matrix size : 434 GBytes
  - 4 factorizations & 8 back/forward solves to compute the currents
  - global CPU time on 1536 processors : **36 083 seconds**

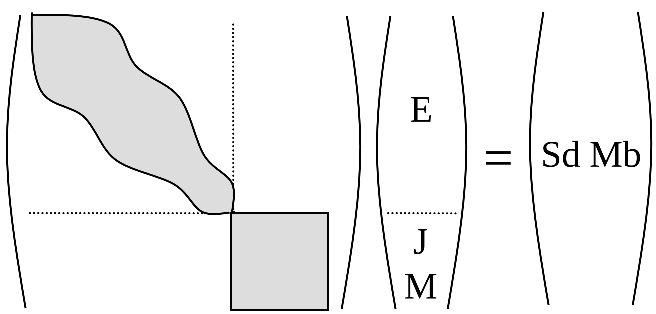




BIEM

- Only homogeneous and isotropic media
- Numerical instability for very thin layers
- Size of the dense linear system ↗ according to the number of layers

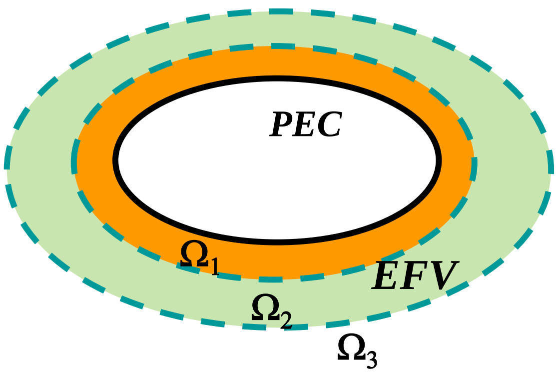
A solution : a strong coupling PDE - BIEM



⇨ It needs to use a Schur complement : elimination of the unknown E

⇨ **BUT need of a great number of computations (solutions) of the sparse linear system**

The other solution: a weak coupling PDE - BIEM



Based on :

* a domain decomposition method (DDM) partitioned into concentric sub-domains
* a Robin-type transmission condition on the interfaces
* on the outer boundary, the radiation condition is taken into account using a new IE formulation called EID
* we solve this problem with inner/outer iteration

① Gauss-Seidel for the global problem

Inside each sub-domain

- ② iterative solver (// conjugate gradient)
- the PaStiX direct solver (EMILIO library)

for the free space problem

- ② a // Fast Multipole Method
- a direct ScaLapack solver

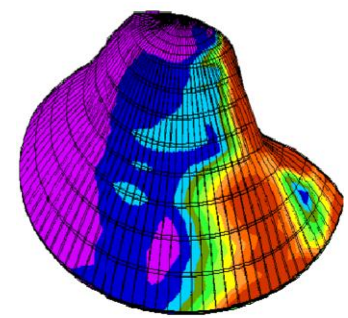
Focus on EMILIO = industrial version of PaStiX solver

EMILIO in few words

- EMILIO was developed in collaboration with INRIA's team

- EMILIO uses efficient parallel implementation of direct methods to solve sparse linear systems, thanks to the high performance direct solver PaStiX and the Scotch package (both developed by INRIA's team)

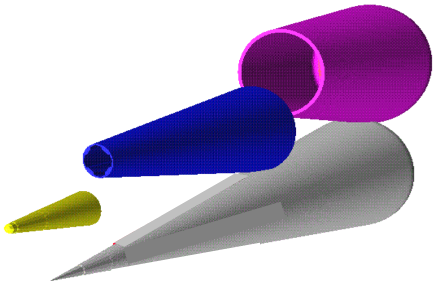
- Organized into two parts :
  - a sequential pre-processing phase using a global strategy,
  - a parallel phase.



- In our 3D code, we use an old version of PaStiX
  - 1-D block column distribution
  - Full MPI implementation
  - Static scheduling

How we use PaStiX (Direct Solver) in EMILIO

Sequential Computation (1 time per problem)



Scotch

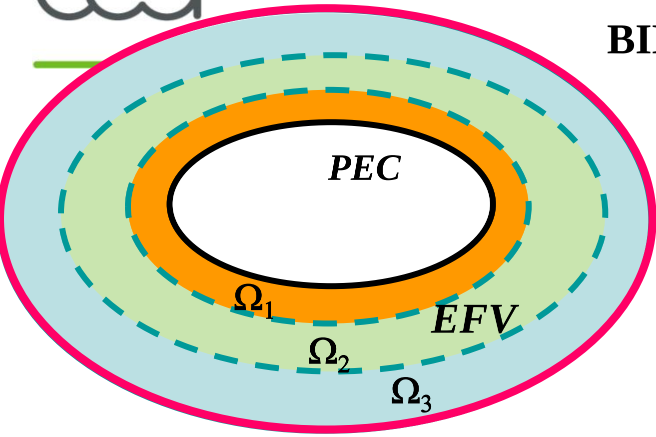
Reordering of the unknowns

PaStiX

Symbolic Factorization

Mesh of the object = graph of the unknowns

How we use PaStiX (Direct Solver) in EMILIO



BIEM : EID

Parallel Computation for the 1st iteration for each volumic subdomain

PaStiX → Odyssee routine → → ↓ ↓ → Odyssee routine → PaStiX → PaStiX

PaStiX

Block mapping of the matrix

Odyssee routine

Finite Element Mapping on the procs

Odyssee routine

Matrix assembly, Boundary Conditions, RHS Assembly

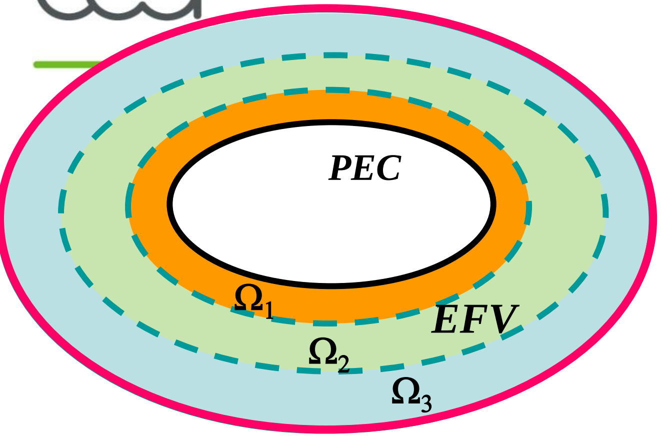
PaStiX

Factorization

PaStiX

Solution

How we use PaStiX (Direct Solver) in EMILIO



BIEM : EID

Parallel Computation for the iteration > 1
for each volumic subdomain

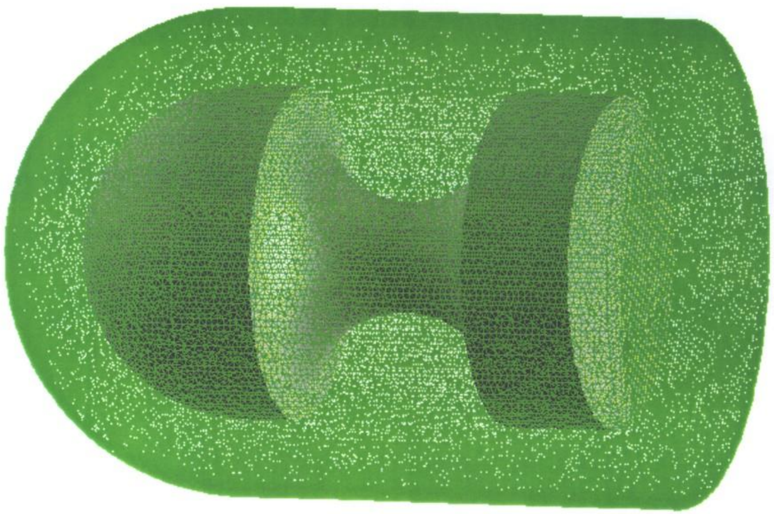
PaStiX ← PaStiX ← Odyssee routine ←

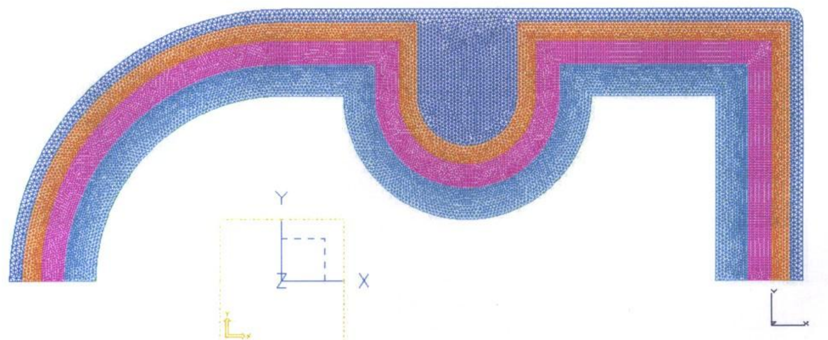
Solution

Factorization

Matrix assembly, Boundary Conditions, RHS Assembly

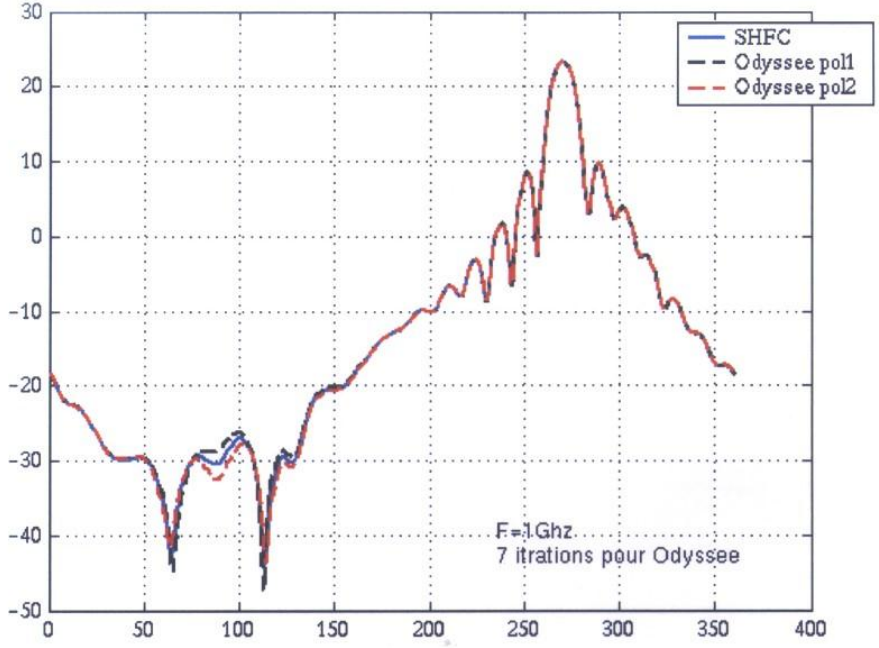
- **Altere :**
  * **5.63 millions** unknowns in 4 sub-domains
  * **120 000** edges for the outer boundary

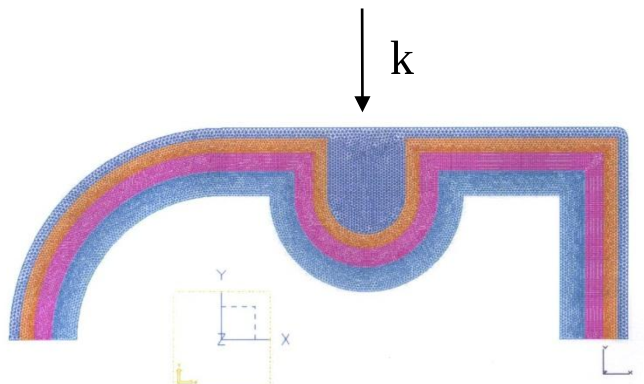




- Comparison 2D axi-symmetric code / Odyssée

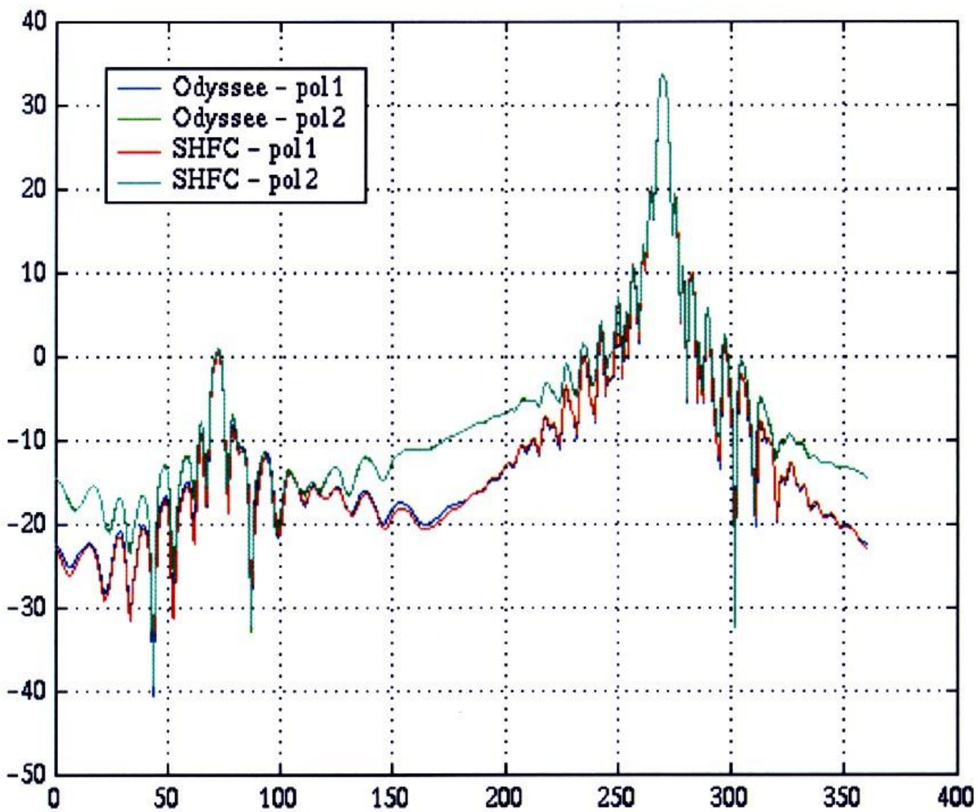
Bistatic RCS at 1GHz





Comparison 2D axi-symmetric / Odyssee

Bistatique RCS- θ=90 degrés

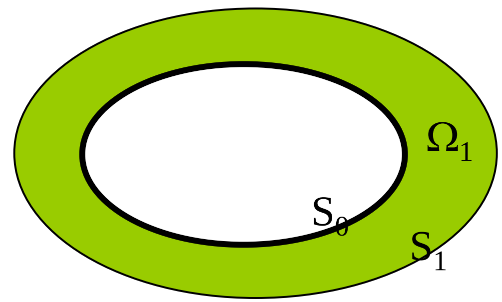


Observations (degrés)

- 7 iterations for the relaxed Gauss-Seidel

Our limit : 23 millions of unknowns

- A complex object :
  * **23 millions** of unknowns in **1 sub-domain**
  * **20 000** unknowns for the EID



Ω_1 : EMILIO – full MPI
(industrial version of PasTiX)

S_1 : EID + MLFMA

About 125 Tera operations needed for the factorization only

Number of procs	512 (32 nodes)
Number of MPI tasks	64 (2 per node)
Memory used per Node	94 Go
Time to Assembly the Matrix	870 s
Time of Factorization	8712 s
Time of Resolution	40 s
Time for the EID	30 s

How to bypass the gap of Memory consumption

<u>Problem :</u> Find a high performance linear solver able to solve very large sparse linear systems (hundreds of millions of unknowns)

<u>Existing software :</u>

- EMILIO (with solver PaStiX full MPI) already integrated in ODYSSEE
- PETSc (iterative solver) already integrated in ODYSSEE

<u>Limits :</u>

- Emilio is limited to 23 Millions of unknowns
- Classical Iterative solver are not well adapted to our problems

- Find New solvers
- Use the PTScotch software to avoid the ordering sequential step
- Test the MPI+Threads version of PaStiX (results on the next slide…)

The biggest problem (for the moment with TERA10…) solved by PaStiX MPI+Threads

3D Problem : 82 Millions of unknowns
(New PasStiX MPI + Threads)

OPC (number of operations needed for factorization) LDLt : 4.28 e+15
NNZ (number of non zeroes in the sparse matrix) A : 5.97 e+10

Number of cores (Tasks MPI / Threads)	Max Time Factorization per core	Max Memory per SMP Node
768 (48/16)	**27750 s**	**115 Go**

Other Softwares studied to bypass the GAP !

Iterative and Multigrid Methods (Joint work with D. Lecas)

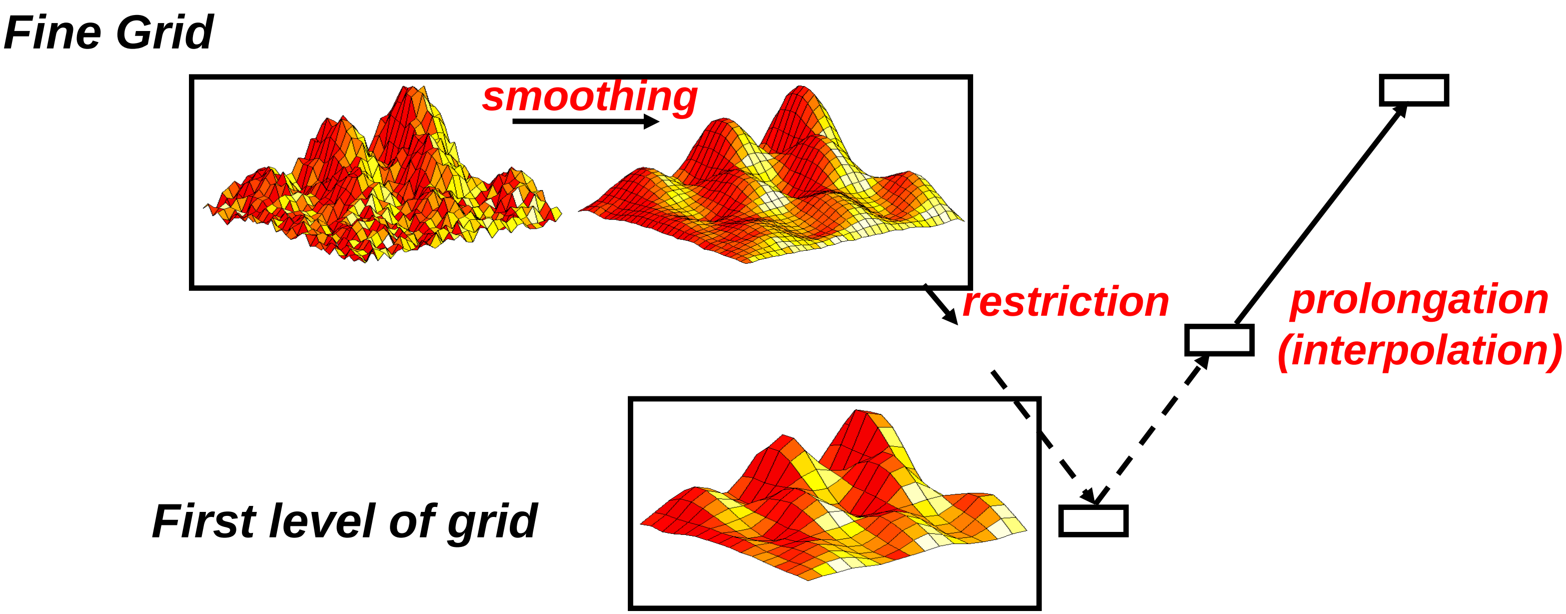
- ILUPACK
- Hypre
- PETSc
- HIPS : Hierarchical Iterative Parallel Solver, solver developped by the INRIA team Bacchus
- MaPHYS : developped by the INRIA team Hiepacs

Direct Methods

- SuperLU
- MUMPS

A parallel hybrid multigrid/direct method

- Compute solution **using direct solver on initial mesh**
- Prolongate solution on finer mesh
- Apply smoother and compute residual
- Restrict residual on coarse mesh and compute error e
- Apply correction using prolonged e then smooth
- Go to Step 2 until desired refinement level is reached



Conclusions and Future research

Conclusions for Parallel codes for electromagnetism

- We have developed a 3D code which is able to take into account all the goals we want to attain

- We successfully validated all the physics through comparison with :
  - other codes we have (full BIEM, 2D axis symmetric)
  - measurements

Future research for Odyssee

- We must finish to develop a complete hybrid, implementation of Odyssee with MPI + Threads + GPU ?

- Improve our methods and our softwares, test some iterative methods for the solution of each sub domain in the volume, work on a Fast Multipole Method adapted to hybrid architecture of new supercomputers

That’s all !

Thank you

Merci

for your attention

Congratulations to the Soccer’s US Team !
No congratulations at all for the French team…..