Higher-Order Finite Element Code for Electromagnetic Simulation on HPC Environments

Luis E. Garcia-Castillo[1],
Daniel Garcia-Doñoro[2], Adrian Amor Martin[1]

[1]Departamento de Teoría de la Señal y Comunicaciones
Universidad Carlos III de Madrid, Spain
[luise,aamor]@tsc.uc3m.es
[2]Xidian University, Xi'an, China.
daniel@xidian.edu.cn

UC3M : a young University established in 1989

3 Campuses in Madrid Region:

- **G**etafe: 11km far from capital
- **L**eganés: 12km far from capital
- **C**olmenarejo: 45km far from capital

3 Schools (Bachelor programs)

- Social and Legal Sciences (G/C)
- Humanities, Communication and Library

  Sciences (G/C)
- Polytechnic School (L/C)

1 Center for Advanced Studies

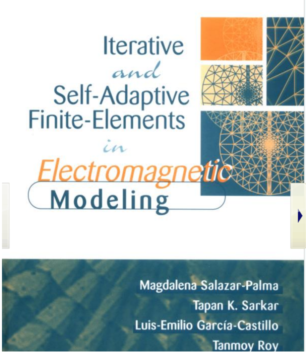
- For Master programs


GREMA – Radiofrequency, Electromagnetics, Microwaves and Antennas Group. Signal Theory and Communications Departme...
GREMA – Radiofrequen...
grema.webs.tsc.uc3m.es
jola
GREMA
RADIOFREQUENCY, ELECTROMAGNETICS, MICROWAVES & ANTENNAS
HOME
GROUP MEMBERS
RESEARCH LINES
PROJECTS
CONTACT
COMPUTATIONAL EM
In-house software based on FEM/MoM/PO/UTD & HPC cluster resources
READ MORE!
About GREMA

Outline

Antecedents

- More than 20 years of experience on numerical methods for EM (mainly FEM but also others). Contributions on:
  - Curl-conforming basis functions
  - Non-standard mesh truncation technique (FE-IIEE) for scattering and radiation problems
  - Adaptivity: *h* and *hp* strategies
  - Hybridization with MoM and high frequency techniques such as PO/PTD and GTD/UTD.
  - . . .
- Code writing from scratch mainly during Ph.D thesis of D. Garcia-Doñoro
- Parallel processing (MPI) and HPC in mind
- Inclusion of well-proven research techniques developed within the research group
- “Reasonable” friendly to be used by non-developers



FEM Formulation

Double-Curl Vector Wave PDE

- Formulation based on double curl vector wave equation (use of **E** or **H**).

$$\boldsymbol{\nabla} \times \left(f_r^{-1} \boldsymbol{\nabla} \times \mathbf{V} \right) - k_0^2 g_r \mathbf{V} = -jk_0 H_0 \mathbf{P} + \nabla \times f_r^{-1} \mathbf{Q}$$

Table: Formulation magnitudes and parameters

	$\mathbf{V}$	$\bar{\bar{f}}_r$	$\bar{\bar{g}}_r$	h	$\mathbf{P}$	$\mathbf{L}$	Γ_D	Γ_N
Form. **E**	$\mathbf{E}$	$\bar{\bar{\mu}}_r$	$\bar{\bar{\epsilon}}_r$	η	$\mathbf{J}$	$\mathbf{M}$	Γ_{PEC}	Γ_{PMC}
Form. **H**	$\mathbf{H}$	$\bar{\bar{\epsilon}}_r$	$\bar{\bar{\mu}}_r$	$\frac{1}{\eta}$	$\mathbf{M}$	$-\mathbf{J}$	Γ_{PMC}	Γ_{PEC}

FEM Formulation

Boundary Conditions and Excitations

- The boundary conditions considered are of Dirichlet, Neumann and Cauchy types:

$$\hat{\mathbf{n}} \times \mathbf{V} = \mathbf{\Psi}_{\mathrm{D}} \quad \text{over } \Gamma_{\mathrm{D}} \tag{1}$$

$$\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \nabla \times \mathbf{V} \right) = \mathbf{\Psi}_{\mathrm{N}} \quad \text{over } \Gamma_{\mathrm{N}} \tag{2}$$

$$\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \nabla \times \mathbf{V} \right) + \gamma\, \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{V} = \mathbf{\Psi}_{\mathrm{C}} \quad \text{over } \Gamma_{\mathrm{C}} \tag{3}$$

- Periodic Boundary Conditions on unit cell (infinite array approach)
- Analytic boundary conditions for waveports of common waveguides and also numerical waveport for arbitrary waveguides by means of 2D eigenvalue/eigenmode characterization.
- Lumped RLC (resistance, coils and capacitors) elements an ports
- Impressed electric and magnetic currents; plane waves.

- Use of **H**(curl) spaces:

$$\mathbf{H}(\text{curl})_0 = \{\mathbf{W} \in \mathbf{H}(\text{curl}),\ \hat{\mathbf{n}} \times \mathbf{W} = 0 \ \text{ on } \ \Gamma_{\text{D}}\} \tag{4}$$

$$\mathbf{H}(\text{curl}) = \{\mathbf{W} \in L^2,\ \boldsymbol{\nabla} \times \mathbf{W} \in L^2\} \tag{5}$$

- and Galerkin method

Find $\mathbf{V} \in \mathbf{H}(\text{curl})$ such that $c(\mathbf{F}, \mathbf{V}) = l(\mathbf{F}), \quad \forall\, \mathbf{F} \in \mathbf{H}(\text{curl})_0$

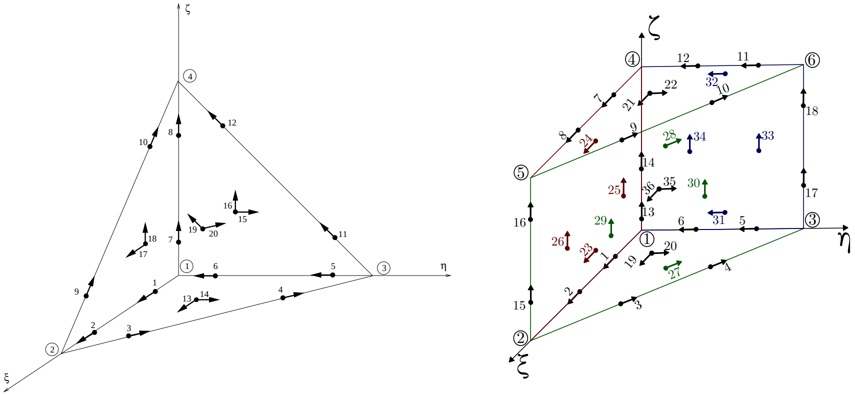
$$c(\mathbf{F}, \mathbf{V}) = \int_\Omega (\boldsymbol{\nabla} \times \mathbf{F}) \cdot \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) d\Omega - k_0^2 \int_\Omega (\mathbf{F} \cdot \bar{\bar{g}}_r \mathbf{V})\, d\Omega + \gamma \int_{\Gamma_{\text{C}}} (\hat{\mathbf{n}} \times \mathbf{F}) \cdot (\hat{\mathbf{n}} \times \mathbf{V})\, d\Gamma_{\text{C}}$$

$$l(\mathbf{F}) = -jk_0 h_0 \int_\Omega \mathbf{F} \cdot \mathbf{P}\, d\Omega \ - \int_{\Gamma_{\text{N}}} \mathbf{F} \cdot \boldsymbol{\Psi}_{\text{N}}\, d\Gamma_{\text{N}} - \int_{\Gamma_{\text{C}}} \mathbf{F} \cdot \boldsymbol{\Psi}_{\text{C}}\, d\Gamma_{\text{C}} - \int_\Omega \mathbf{F} \cdot \boldsymbol{\nabla} \times \left(\bar{\bar{f}}_r^{\,-1} \mathbf{L} \right) d\Omega$$

FEM Discretization

Higher-Order Curl-Conforming Element

- Own family of higher order isoparametric curl-conforming finite elements (tetrahedron, prism, hexahedron —under test—)
  Rigorous implementation of Nedelec's mixed order elements

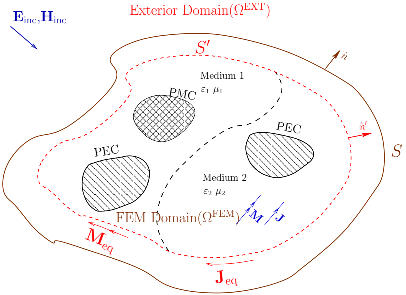


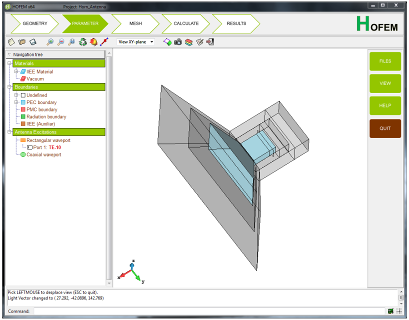
Example: 2nd order versions of tetra and prism

Mesh Truncation with FE-IIEE

Problem Set-Up

- Open region problems (optionally) by means of FE-IIEE (Finite Element - Iterative Integral Equation Evaluation)
  $\Rightarrow$ Asymptotically exact absorbing boundary condition





- Local B.C. for FEM (sparse matrices)

$$\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{-1} \boldsymbol{\nabla} \times \mathbf{V} \right) + \gamma \, \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{V} = \boldsymbol{\Psi}_{\mathrm{INC}} + \boldsymbol{\Psi}_{\mathrm{SCAT}} \text{ over } S$$

- Iterative estimation of $\boldsymbol{\Psi}_{\mathrm{INC}}$ by exterior Equivalence Principle on S'

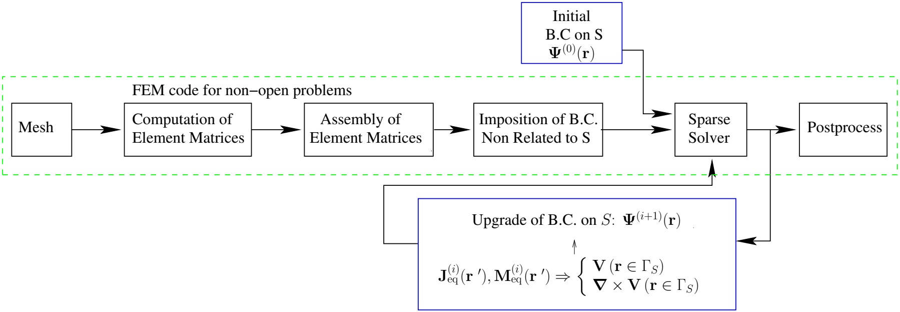
$$\mathbf{V}^{\text{FE-IIEE}} = \oiint_{S'} (\mathbf{L}_{\mathrm{eq}} \times \boldsymbol{\nabla} G) \cdot \mathbf{dS}' - jk_0 h_0 \oiint_{S'} \left(\mathbf{O}_{\mathrm{eq}} \left(G + \frac{1}{k_0^2} \boldsymbol{\nabla}\boldsymbol{\nabla} G \right) \right) \cdot \mathbf{dS}'$$

$$\boldsymbol{\nabla} \times \mathbf{V}^{\text{FE-IIEE}} = jk_0 h_0 \oiint_{S'} (\mathbf{O}_{\mathrm{eq}} \times \boldsymbol{\nabla} G) \, \mathbf{dS}' - \oiint_{S'} \left(\mathbf{L}_{\mathrm{eq}} \left(k_0^2 G + \boldsymbol{\nabla}\boldsymbol{\nabla} G \right) \right) \cdot \mathbf{dS}'$$

$$\boldsymbol{\Psi}_{\mathrm{SCAT}} = \hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{-1} \boldsymbol{\nabla} \times \mathbf{V}^{\text{FE-IIEE}} \right) + \gamma \, \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{V}^{\text{FE-IIEE}}$$

Mesh Truncation with FE-IIEE

Flow Chart



Computational Features

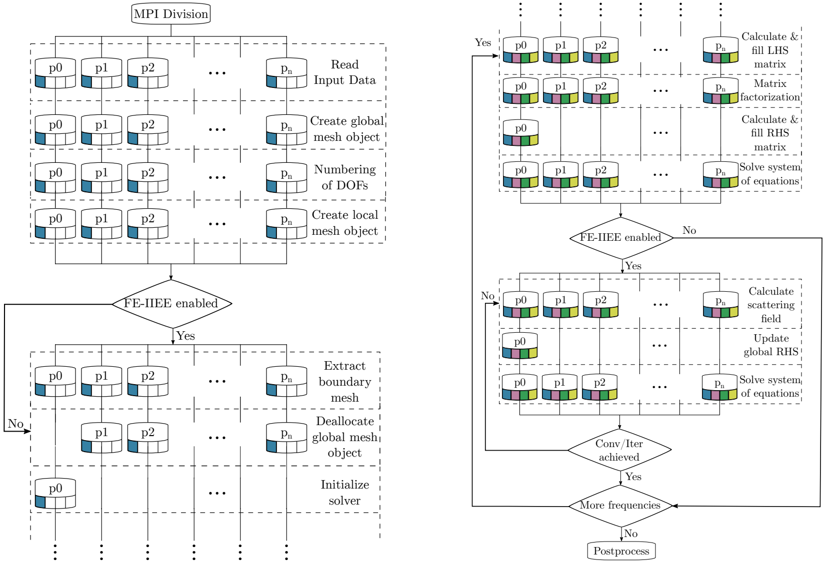
Computational Features

- Code written using modern Fortran constructions (F2003)
- Strong emphasis in code maintainability by use of OOP (Object Oriented Programming) paradigms
- Numerical *verification* by use of the Method of Manufactured Solutions. Numerical *validation* by tests with EM benchmark problems
- Hybrid MPI+OpenMP programming
- Direct solver interfaces (HSL, MUMPS, MKL Pardiso, . . .)
- Graphical User Interface (GUI) with HPCaaS Interface
- Linux & Windows versions

Towards HPC

- “Rethink” some of the OOP constructions (e.g., arrays of small derived types, . . .)
- Global mesh object $\rightarrow$ local mesh objects on each processor
- Specialized direct solver interfaces
- . . .
- . . .
- Problems of several tens of millions of unknowns on more than one thousand cores

Parallel Flow Chart of the Code



Features

- GUI based on a general purpose pre- and post-processor called GiD `http://gid.cimne.upc.es/`
- Creation (or importation) of the geometry model of the problem
- Mesh generation
- Assignation of material properties and boundary conditions
- Visualization of results
- Integration with *Posidona* (in-house HPCaaS)

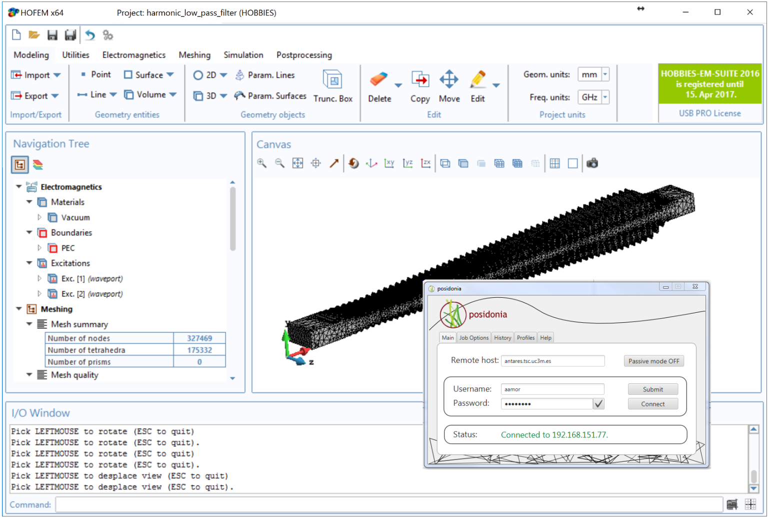
Easing the use of HPC platforms

- Remote job-submission to HPC infraestructures
- Designed with security, user-friendliness, collaborative-computing and mobility, in mind.
- Management of all the communication with the remote computer system (file transfers, . . .)
- Interaction with its batch system (job scheduler).
- History repository of simulations
- Notification when job submitted is completed
- Transparent downloading of the results to visualize them locally.
- Posidonia also available as stand-alone desktop/Android/Web solution (also for general use with other simulator and/or applications)

A. Amor-Martin, I. Martinez-Fernandez, L. E. Garcia-Castillo. “Posidonia: A Tool for HPC and Remote Scientific Simulations”. *IEEE Antennas and Propagation Magazine*, 6:166–177, Dec. 2015.

GUI with HPCaaS Interface

Screenshot



Getting Experience with MUMPS
From "MUMPS do it all" to . . .

Matrix Format

- Elemental
- Assembled (centralized on process 0)
- Assembled (distributed among processes)
- Asking to MUMPS for Schur complements and "playing" with them (outside MUMPS)
- . . .

RHS and solution

- Dense RHS
- Sparse RHS (large number of RHS vectors)
- Centralized solution
- Distributed solution? (waiting for distributed RHS feature. . .)

MUMPS Interface

Present Stage of Development (version 5.0.2)

- MUMPS initialization
- Call to ParMETIS (or PT-Scotch) to partition matrix among processors
  - Other alternatives for partitioning have been considered due to memory problems (commented in following slides)
- Computation of FEM matrix coefficients associated to each local process
- Input of matrix coefficients to MUMPS in distributed assembled format.
- Call to MUMPS for matrix factorization
- Computation of FEM RHS coefficients on process 0 (in blocks of 10-20 vectors) in sparse format
- Call to MUMPS for system solve
- (FE-IIEE enabled) Iteratively update of RHS and system solve until error criterion is satisfied
- MUMPS finalization

∗ Frequent use of *out-of-core* (OOC) capabilities of MUMPS

Some Issues with Memory

Memory Allocation inside MUMPS

Memory Issue

- A peak memory use during analysis phase has been detected (distributed assembled)
- Found out to be due to memory allocation inside MUMPS routines related to maximum MAXS among processors

Listing 1: file zana_aux_par.F

```
1589   SUBROUTINE ZMUMPS_BUILD_LOC_GRAPH
...
1647    MAXS = ord\%LAST(I)-ord\%FIRST(I)+1
...
1653      ALLOCATE(SIPES(max(1,MAXS), NPROCS))
...
1864 END SUBROUTINE ZMUMPS_BUILD_LOC_GRAPH
```

Some Issues with Memory (cont.)

Memory Allocation inside MUMPS

Memory Issue

- Example: 45.000.000 dof problem using 192 processes and 4 bytes per integer:
  MAXS bandwidth is 45.000.000
  $\Rightarrow$ 34.56 GB memory per process

Workaround

- Matrix partition based on rows instead of elements of the mesh
  - Slightly worse LU fill-in (size of cofactors) than with partition based on elements
- Change ordering of dof as input to MUMPS? (to be done)

It may be the case we are doing something completely wrong

Some Issues with Memory

Large Number of RHS vectors

Analysis of a given problem under a large number of excitations. Examples:

- Monostatic *radar cross section* (RCS) prediction
- Large arrays of antennas

Present MUMPS Interface

Treatement of RHS solution & vectors in blocks (typically 10-20 vectors at a time)

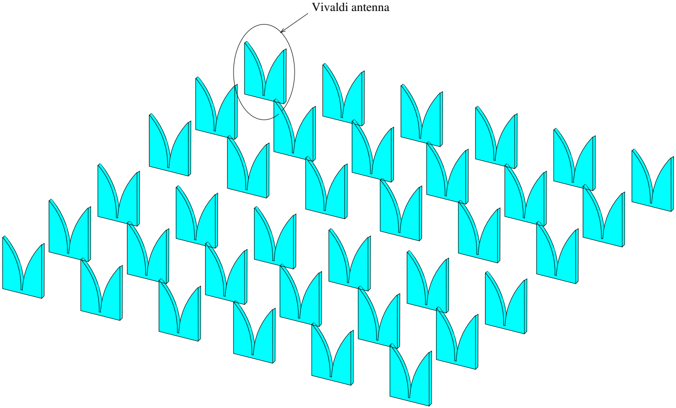
- Use of sparse format for RHS
- Use of centralized solution vectors
  - The reason behind the treatment of RHS & solution vectors in blocks is to limit the memory needed to storage solution vectors

Distributed Solution Planned for Near Future

- Update of centralized RHS by FE-IIEE $\Rightarrow$ use of centralized solution is "natural" (easy in terms of code maintenance)
- Wish list: distributed RHS
  ¿is distributed RHS feature planned for near future versions of MUMPS?

Specialized MUMPS Interfaces

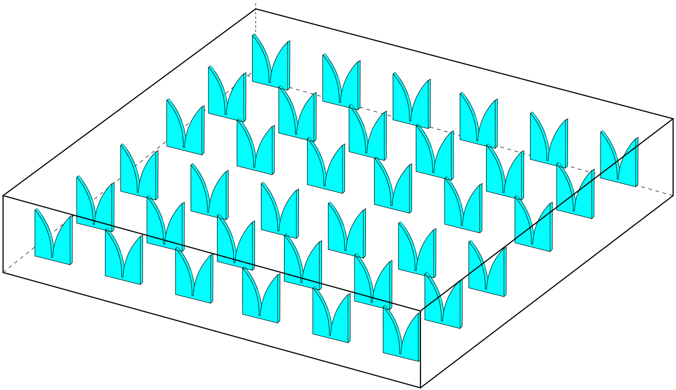
Repetitive Solver



Antenna Array

Specialized MUMPS Interfaces (cont.)

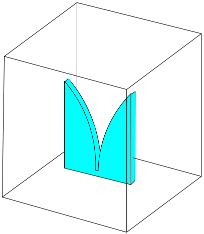
Repetitive Solver



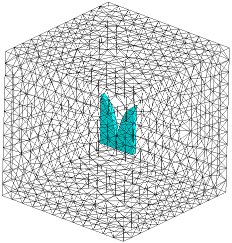
Antenna Array

Specialized MUMPS Interfaces (cont.)

Repetitive Solver



(a) Unit Cell



(b) Unit Cell Mesh

Specialized MUMPS Interfaces (cont.)

Repetitive Solver

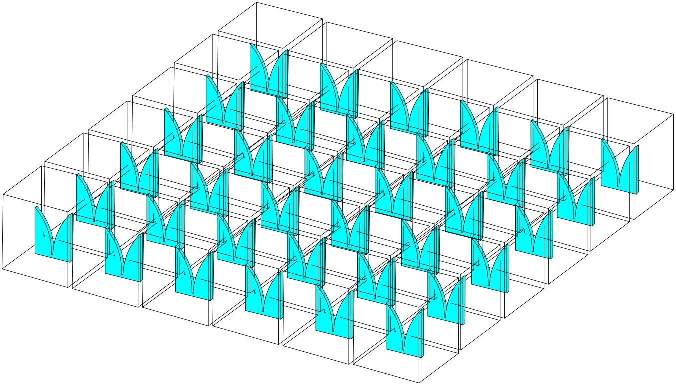


Figure: Virtual Mesh of Antenna Array

Specialized MUMPS Interfaces (cont.)

Repetitive Solver

Algorithm

1. Computation of Schur complement of unit-cell
2. Assembled of Schur complements of all “virtual” cells $\Rightarrow$ Interface problem
3. Addition of boundary conditions to interface problem
4. Solve the interface problem
5. Solve interior unit cell problems
    - Identical matrices with different right hand sides

Specialized MUMPS Interfaces (cont.)

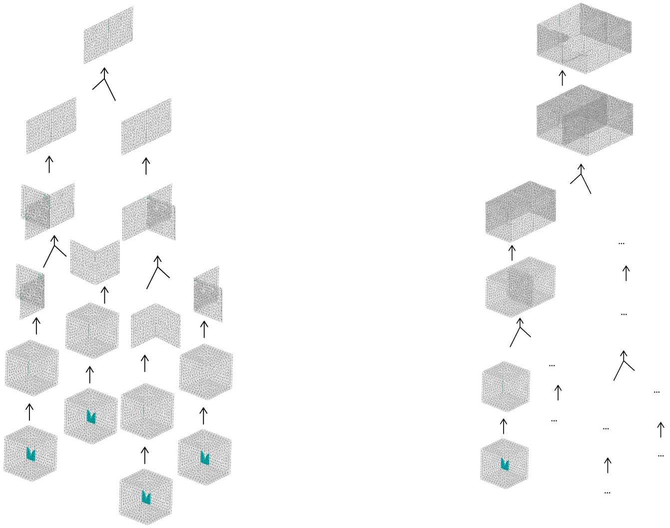
Repetitive Solver

Features and Remarks

- Advantages: saving in time and memory
- Under certain circumstances (number of cells equal to power of 2 and no B.C.) all leaves of a certain level of the tree are identical
  - Further saving in time
  - Large saving in memory
- Boundary conditions (B.C.) alter this one branch tree behavior.
  $\Rightarrow$ B.C. may be left up to the root of the tree
- Or “algebraic symmetry” can be explored

Specialized MUMPS Interfaces (cont.)

Repetitive Solver



Hybrid Direct & Iterative Solver

- Multifrontal algorithm on only a few levels
- Iterative solution from the last level of multifrontal algorithm
- It can be understood as the direct solver acting as preconditioner of the iterative solver.
- Natural approach to some DDM strategies

Lack of Availability of LU Cofactors

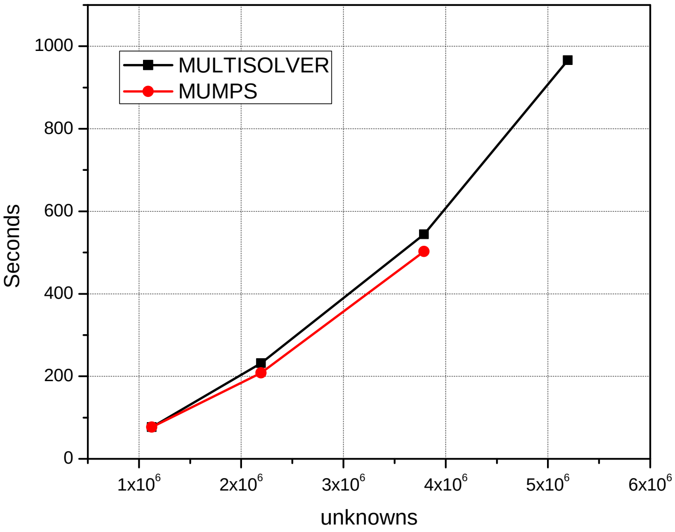
- Calls to multiple (typicall sequential) MUMPS instances for Schur complements
- Assembling Schur complements
- Finalizing MUMPS instances
- Solve interface problem
- Create new MUMPS instances to solve the interior problems

MUMPS Instances for Interior Problems

- Idea inspired by work leaded by Prof. Paszynski:
  - Reproduction (or restore) of interior matrix equation and interior right hand side
  - Call to multiple (typically sequential) MUMPS instances to factorize/solve the interior problems.
  - Use of Dirichlet conditions for interface unknowns
  - Preliminary tests shows that the approach is worthy in memory (expected) but competitive in time

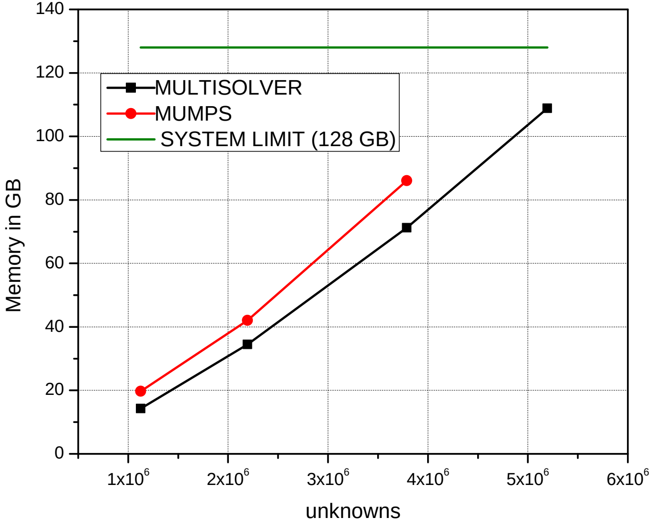
Regular MUMPS & MULTISOLVER

Time and Memory Comparison



Regular MUMPS & MULTISOLVER (cont.)

Time and Memory Comparison



Cluster of Xidian University

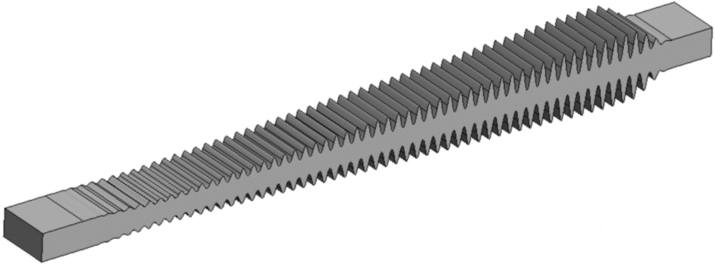
- 140 compute nodes
  - Two twelve-core Intel Xeon 2690 V2 2.2 GHz CPUs
  - 64 GB of RAM
  - 1.8 TB of hard disk
- 56 Gbps InfiniBand network.

Waveguide Problem

Low Pass Filter with Higher-Order Mode Suppression

- [10 – 16] GHz
- Length: 218 mm
- 324.5 K tetrahedrons

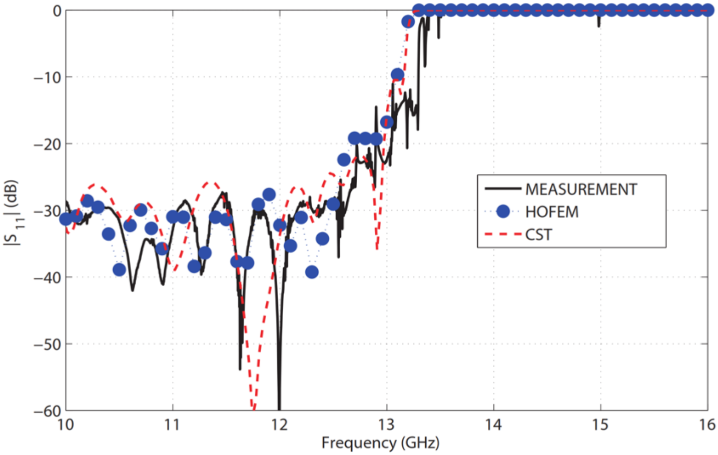
- 2.2 M unknowns
- Wall time: 7.3 min per freq. point



I. Arregui et al., “High-power low-pass harmonic filters with higher-order TE_{n0} and non-TE_{n0} mode suppression: design method and multipactor characterization”, IEEE Trans. Microw. Theory Techn., Dec. 2013.

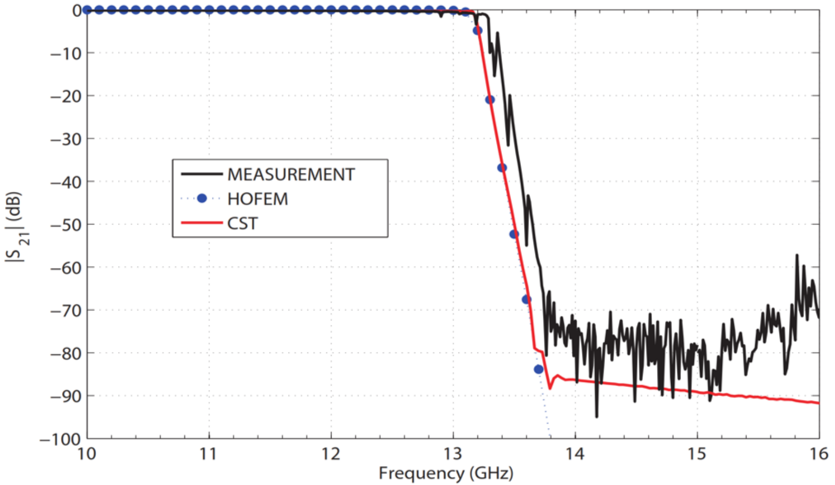
Waveguide Problem (cont.)

Low Pass Filter with Higher-Order Mode Suppression



Waveguide Problem (cont.)

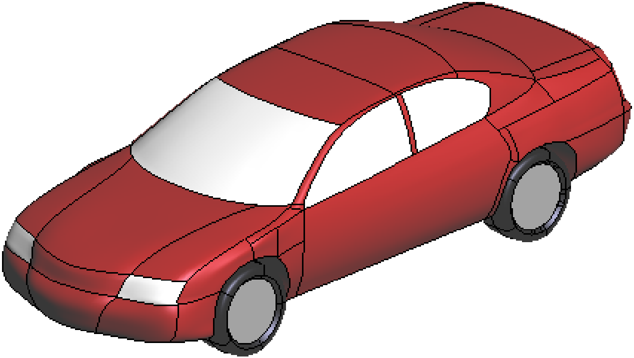
Low Pass Filter with Higher-Order Mode Suppression



Scattering Problem

Bistatic RCS of Car

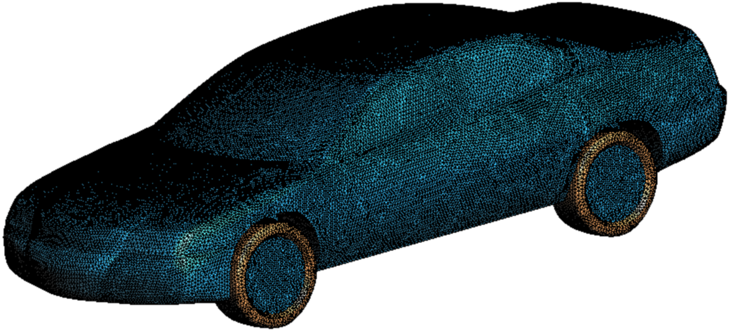
- Bistatic RCS at 1.5 GHz
- Tyres modeles as dielectric ($\varepsilon_r = 40$)
- Several incident planes waves from different directions



Scattering Problem (cont.)

Bistatic RCS of Car

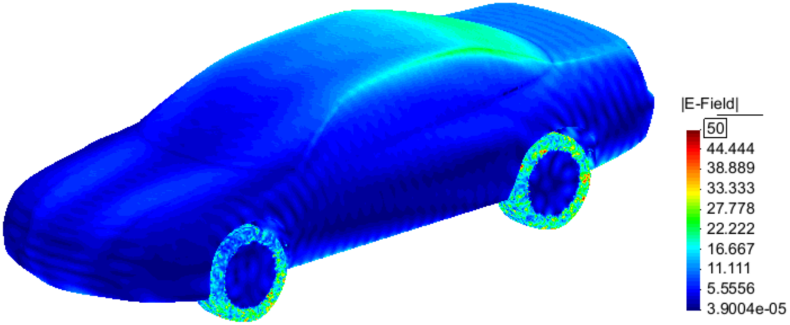
- 2.7 M tetrahedrons
- 17.3 M unknowns
- Wall time: 59 min per freq. point (46 compute nodes)



Scattering Problem (cont.)

Bistatic RCS of Car

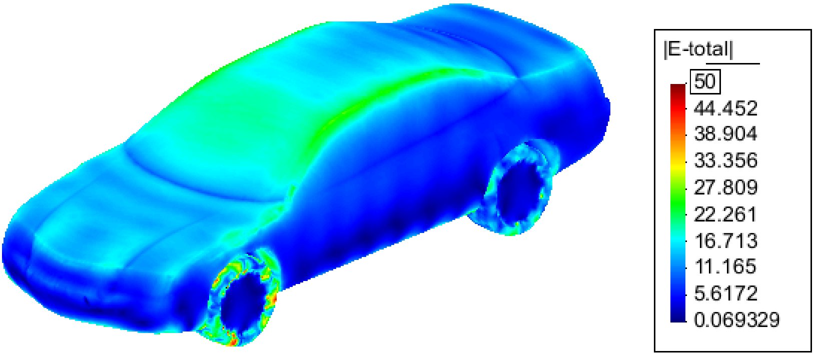
- ❑ Incident plane wave arriving from behind



Scattering Problem (cont.)

Bistatic RCS of Car

- Incident plane wave arriving from the front



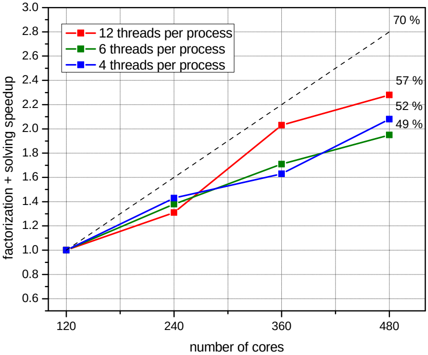
Scattering Problem (cont.)

Bistatic RCS of Car

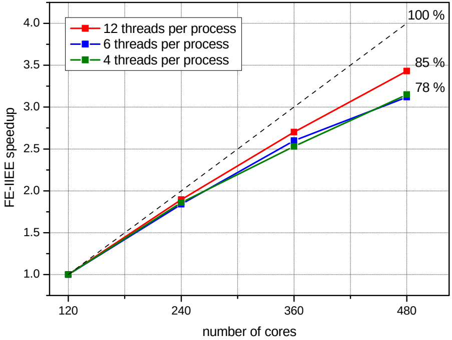
- ❑ Incident plane wave arriving from the front

Parallel Scalability

Factorization and FE-IIEE Stages



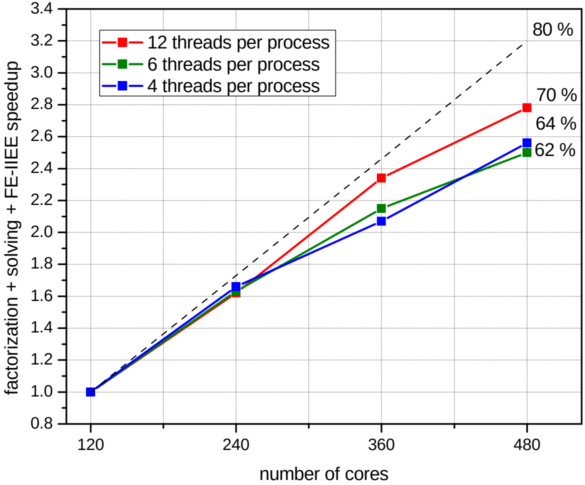
Speedup graph corresponding to the factorization phase.



Speedup graph corresponding to the mesh truncation phase

Benchmark: Bistatic RCS of Impala

Parallel Scalability
Whole Code



Speedup graph corresponding to the whole code

Work in Progress and Future Work

Work in Progress

- Hierarchical basis functions of variable order p
- h-adaptivity $\Rightarrow$ support for *hp* meshes

Future Work

- Conformal and non-conformal DDM
- Hybrid (direct + iterative) solver

Thanks for your attention!

Thanks to the MUMPS team!!!