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

Luis E. Garcia-Castillo\textsuperscript{(1)}, Daniel Garcia-Doñoro\textsuperscript{(2)}, Adrian Amor Martin\textsuperscript{(1)}

\textsuperscript{(1)}Departamento de Teoría de la Señal y Comunicaciones
Universidad Carlos III de Madrid, Spain
[luise, aamor]@tsc.uc3m.es

\textsuperscript{(2)}Xidian University, Xi’an, China.
daniel@xidian.edu.cn
UC3M: a young University established in 1989

3 Campuses in Madrid Region:
• Getafe: 11km far from capital
• Leganés: 12km far from capital
• Colmenarejo: 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 Department.

GREMA
RADIOFREQUENCY, ELECTROMAGNETICS, MICROWAVES & ANTENNAS

COMPUTATIONAL EM
In-house software based on FEM/MoM/PO/UTD & HPC cluster resources

READ MORE!

About GREMA
GREMA has developed a general purpose electromagnetic parallel solver based on the Finite Element Method (FEM) called HOFEM (Higher Order FEM).

It makes use of its own higher-order isoparametric curl-conforming tetrahedra and prisms. Scattering and radiation open region problems use an arbitrarily accurate mesh truncation boundary condition retaining the original sparse structure of the matrices.

HOFEM provides a user-friendly graphical user interface with flexible and powerful pre- and post-processing capabilities and a remote job submission tool for HPC clusters.

Luis Emilio García Castillo
luise@tsc.uc3m.es
IN-HOUSE CODES

GREMA has a self-adaptive hp-FEM code that achieves exponential rates of convergence for arbitrary problems.

Codes implementing the hybridization on a fully coupled sense of FEM, MoM, PO/PTD, and UTD are also used inside the group with successful results for antenna reflectors, antenna placement, on-board antennas or RCS prediction of multi-scale objects.

HPC CLUSTER

The group has a 32 nodes HPC cluster equipped with the latest technology and software that is able to run any kind of simulation that GREMA needs.

The cluster provides more than 2 TFlops of computation capacity with 1 TB RAM and 16 TB HDD. It is equipped with virtualization technologies that improve the computational resources flexibility.

REMOTE SIMULATION

The group has implemented an intuitive tool to run simulation codes on HPC clusters providing an easy way to use these complex computational systems. This tool may be customized for any code under request.

COMPUTING SERVICES

GREMA offers access to its HPC systems for private companies and research oriented institutions in which they can perform their simulations using their own software. The service may also include access to GREMA simulation software resulting in a full (hardware & software) solution.
Outline

1. Antecedents
2. Parallel Higher-Order FEM Code
3. Applications & Performance
4. Work in Progress and Future Work
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

- Inclusion of well-proven research techniques developed within the research group

- “Reasonable” friendly to be used by non-developers
Formulation based on double curl vector wave equation (use of $E$ or $H$).

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

Table: Formulation magnitudes and parameters

<table>
<thead>
<tr>
<th></th>
<th>$V$</th>
<th>$\bar{f}_r$</th>
<th>$\bar{g}_r$</th>
<th>$h$</th>
<th>$P$</th>
<th>$L$</th>
<th>$\Gamma_D$</th>
<th>$\Gamma_N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form. E</td>
<td>$E$</td>
<td>$\bar{\mu}_r$</td>
<td>$\bar{\epsilon}_r$</td>
<td>$\eta$</td>
<td>$J$</td>
<td>$M$</td>
<td>$\Gamma_{PEC}$</td>
<td>$\Gamma_{PMC}$</td>
</tr>
<tr>
<td>Form. H</td>
<td>$H$</td>
<td>$\bar{\epsilon}_r$</td>
<td>$\bar{\mu}_r$</td>
<td>$\frac{1}{\eta}$</td>
<td>$M$</td>
<td>$-J$</td>
<td>$\Gamma_{PMC}$</td>
<td>$\Gamma_{PEC}$</td>
</tr>
</tbody>
</table>
The boundary conditions considered are of Dirichlet, Neumann and Cauchy types:

\[ \hat{n} \times V = \psi_D \quad \text{over } \Gamma_D \]  
\[ \hat{n} \times \left( f_r^{-1} \nabla \times V \right) = \psi_N \quad \text{over } \Gamma_N \]  
\[ \hat{n} \times \left( f_r^{-1} \nabla \times V \right) + \gamma \hat{n} \times \hat{n} \times V = \psi_C \quad \text{over } \Gamma_C \]  

- 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 $H(\text{curl})$ spaces:

\[
H(\text{curl})_0 = \{ W \in H(\text{curl}), \hat{n} \times W = 0 \text{ on } \Gamma_D \} 
\]

\[
H(\text{curl}) = \{ W \in L^2, \nabla \times W \in L^2 \} 
\]

and Galerkin method

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

\[
c(F, V) = \int_{\Omega} (\nabla \times F) \cdot \left( f_r^{-1} \nabla \times V \right) d\Omega - k_0^2 \int_{\Omega} (F \cdot \tilde{g}_r V) d\Omega + \\
\gamma \int_{\Gamma_C} (\hat{n} \times F) \cdot (\hat{n} \times V) d\Gamma_C
\]

\[
l(F) = -jk_0 h_0 \int_{\Omega} F \cdot P d\Omega - \int_{\Gamma_N} F \cdot \Psi_N d\Gamma_N - \int_{\Gamma_C} F \cdot \Psi_C d\Gamma_C \\
- \int_{\Omega} F \cdot \nabla \times \left( f_r^{-1} L \right) d\Omega
\]
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)
  - Asymptotically exact absorbing boundary condition
Mesh Truncation with FE-IIIEE
Algorithm

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

\[ \hat{n} \times \left( \frac{1}{f_r} \nabla \times V \right) + \gamma \hat{n} \times \hat{n} \times V = \psi_{\text{INC}} + \psi_{\text{SCAT}} \text{ over } S \]

- Iterative estimation of \( \psi_{\text{INC}} \) by exterior Equivalence Principle on \( S' \)

\[ \mathbf{V}^{\text{FE-IIIEE}} = \iint_{S'} (\mathbf{L}_{eq} \times \nabla G) \cdot dS' - jk_0 h_0 \iint_{S'} \left( \mathbf{O}_{eq} \left( G + \frac{1}{k_0^2} \nabla \nabla G \right) \right) \cdot dS' \]

\[ \nabla \times \mathbf{V}^{\text{FE-IIIEE}} = jk_0 h_0 \iint_{S'} (\mathbf{O}_{eq} \times \nabla G) dS' - \iint_{S'} \left( \mathbf{L}_{eq} \left( k_0^2 G + \nabla \nabla G \right) \right) \cdot dS' \]

\[ \psi_{\text{SCAT}} = \hat{n} \times \left( \frac{1}{f_r} \nabla \times \mathbf{V}^{\text{FE-IIIEE}} \right) + \gamma \hat{n} \times \hat{n} \times \mathbf{V}^{\text{FE-IIIEE}} \]
Mesh Truncation with FE-IIEE

Flow Chart

Mesh Truncation with FE-IIEE

Mesh

FEM code for non-open problems

Computation of Element Matrices

Assembly of Element Matrices

Imposition of B.C. Non Related to S

Sparse Solver

Postprocess

Initial B.C. on S

$\Psi^{(0)}(r)$

Upgrade of B.C. on S: $\Psi^{(i+1)}(r)$

$J^{(i)}_{eq}(r'), M^{(i)}_{eq}(r') \Rightarrow \begin{cases} V(r \in \Gamma_S) \\ \nabla \times V(r \in \Gamma_S) \end{cases}$
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
“Rethink” some of the OOP constructions (e.g., arrays of small derived types, . . .)

Global mesh object → 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

MPI Division

Yes

No

p0 p1 p2 pn...

Read Input Data

Create global mesh object

Numbering of DOFs

Create local mesh object

FE-IIEE enabled

Yes

No

... Calculate & fill LHS matrix

... Matrix factorization

... Calculate & fill RHS matrix

... Solve system of equations

... Calculate scattering field

... Update global RHS

... Solve system of equations

Conv/Iter achieved

Yes

No

More frequencies

Yes

No

Postprocess
Graphical User Interface

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 infrastructures
- 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)

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 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 \texttt{MAXS} among processors

Listing 1: file \texttt{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
  - \( \Rightarrow 34.56 \text{ GB memory per process} \)

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 & Solution Vectors

**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**
Treatment 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
Large Number of RHS & Solution Vectors

- 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
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
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.
  ⇒ 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
Specialized MUMPS Interfaces
Dealing with the Lack of Availability of LU Cofactors

Lack of Availability of LU Cofactors

- Calls to multiple (typically 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

GREMA-UC3M
Higher Order Finite Element Code...v1.2
Regular MUMPS & MULTISOLVER (cont.)

Time and Memory Comparison

<table>
<thead>
<tr>
<th>Unkowns (x10^6)</th>
<th>MUMPS</th>
<th>MULTISOLVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>64</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>128</td>
</tr>
</tbody>
</table>

Memory in GB

SYSTEM LIMIT (128 GB)
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

Waveguide Problem (cont.)
Low Pass Filter with Higher-Order Mode Suppression

![Graph showing |S_{11}| (dB) vs Frequency (GHz)]
Waveguide Problem (cont.)
Low Pass Filter with Higher-Order Mode Suppression
Scattering Problem
Bistatic RCS of Car

- Bistatic RCS at 1.5 GHz
- Tyres modeled 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)
Incident plane wave arriving from behind
Incident plane wave arriving from the front
Incident plane wave arriving from the front
32 element array + feeding network

- [2 – 3] GHz
- 3.4 M tetrahedrons
- 23 M unknowns
Radiation Problem (cont.)

LTE 4G Base Station Antenna

- Wall time: 38 min per freq. point
- 48 compute nodes, 1152 cores
- Out-of-Core using 1.14 TB RAM
3D representation of directivity of the array at 2.6 GHz
64 element array + feeding network

- [2 – 3] GHz
- Length: 1.6 m
- 6.9 M tetrahedrons
- 45.1 M unknowns
Wall time: 5.5 h per freq. point

- 48 compute nodes, 1152 cores
- Out-of-Core using 1.9 TB RAM
3D representation of directivity of the array at 2.6 GHz
Parallel Scalability
Factorization and FE-IIIEE 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

GREMA-UC3M
Higher Order Finite Element Code . . . v1.2
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!!!