Parallel multigrid solver for time harmonic Maxwell equations

M. Chanaud\textsuperscript{1}  L. Giraud\textsuperscript{1}  D. Goudin\textsuperscript{2}  JJ. Pesqué\textsuperscript{2}  J. Roman\textsuperscript{1}

\textsuperscript{1}HiePACS project, INRIA Bordeaux Sud-Ouest, Talence, France
joint laboratory INRIA-CERFACS on HPC

\textsuperscript{2}CEA/DAM/CESTA, Le Barp, France

Sparse Days 2011,
Sep. 6-7 2011, Toulouse, France
Aim: compute large 3D stealthy objects Radar Cross Section (RCS).

We consider a concentric layer decomposition of a global domain $\Omega$

- Innermost boundary ($\Sigma_0$) represent the studied object
- $\Omega_n$ outer boundary ($\Sigma_n$) represent the infinite domain

Must solve time harmonic Maxwell equations over all these domains

$$\nabla \times \nabla \times \vec{E} - k^2 \vec{E} = 0$$

This talk will focus on volume subdomain solutions.
Global domain resolution scheme

- Volume subdomains discretized by Nédélec’s 3D first order finite elements → sparse symmetric complex linear system.
- Radiation condition applies on the outer surface $\Sigma_n$. Despres Integral Equations (EID) formulation → dense complex linear system.

Each volume subdomain is coupled to its neighbors by a Robin boundary condition (transmission).

Global problem solved by iterative block Gauss-Seidel method:
- Direct method in volume subdomains
- Coupling boundary condition
- Iterative + Fast Multipole Method on outer surface subdomain ($S_n$)
Volume subdomains

Direct methods ($LL^T$, $LDL^T$ factorization) are used on volume subdomains problems. Performed in actual software suite by PaStiX parallel direct solver.

http://pastix.gforge.inria.fr/

Advantages & disadvantages

+ Solution accuracy
+ Predictable behavior (static mapping)
+ Multiple right-hand sides
+ Triangular solution complexity ($O(n^{4/3})$)
  - Huge memory requirements
  - Factorization complexity ($O(n^2)$)

Classical problems require tens of millions of unknowns per volume subdomain.
Full-Multigrid

Compute on multiple level of meshes.
Direct method on coarse mesh (small initial problem).
Solution interpolated on fine mesh and solved iteratively (smoother).

Advantages & disadvantages

+ Low memory consumption (small factorized matrix)
+ Linear multigrid complexity ($O(n)$)
+ Automatic uniform refinement
- Less accurate than direct method
Nédélec’s base element

Unknowns are carried by edges rather than nodes.
Values are computed according to the following equation:

\[ \alpha_i = \int_{\Gamma_i} \vec{E}(\xi_1, \ldots, \xi_n) \cdot \vec{\tau}_i \, dl. \]

Where:
- \( \Gamma_i \) – the considered edge,
- \( \vec{\tau}_i \) – \( \Gamma_i \) unitary vector,
- \( dl \) – displacement along \( \Gamma_i \).
Some iterative methods reduce High Frequency (HF) error modes
Remaining error composed of Low Frequency (LF) error modes
LF on fine grid becomes HF on coarse grid

Figure: Error over a 10x10 regular mesh (before and after smoothing)
Multigrid V-cycle and full-multigrid

- Coarse grid factorization performed **once**
- Multiple coarse grid solution
- Residual restricted from fine to coarse mesh
- Solution/error prolongated from coarse to fine mesh
Mesh refinement and coarsening

Unstructured mesh refinement:
- uniform: easy coarsening
- adaptive: hard coarsening or storage needed

Two uniform refinements:
- edge refinement: split each edge
  - good aspect ratio,
  - large number of elements (12x elements),
  - coupling interface modified.
- cell refinement: one new vertex per elt.
  - coupling interfaces preserved,
  - slow problem size increase (4x elements),
  - aspect ratio (flat elements).
Jacobi decomposition

Consider the following componentwise decomposition of $A$:

\[ A = D + L + U. \]

Where:
- $D$ – diagonal matrix
- $L$ – lower triangular
- $U$ – upper triangular

The Jacobi iteration is defined by

\[ x^{k+1} = x^k + D^{-1}(b - Ax^k) \]

Damped Jacobi iteration

\[ x^{k+1} = (1 - \omega)x^k + \omega(x^k + D^{-1}(b - Ax^k)) \]
\[ x^{k+1} = x^k + \omega(D^{-1}(b - Ax^k)) \]
Matrix-free Jacobi

From the damped Jacobi equation

\[ x^{k+1} = x^k + \omega (D^{-1} (b - Ax^k)), \]

one Jacobi iteration consists "only" in a matrix-vector product.

Given the assembly equality

\[ A = \sum_{\text{elements}} A_e, \]

matrix-vector multiplication can be performed without global matrix assembly

⇒ Low memory consumption
⇒ Only vectors are stored
⇒ Every element matrix computed multiple times ("cheap")
Parallel aspects

- First approximation computed by a parallel direct method
- Unknowns renumbered by nested dissection algorithm and halo
  approximate minimum degree (PT-Scotch software)
  http://scotch.gforge.inria.fr/
- Static mapping computed by PaStiX

→ Use this data distribution for the iterative solver.

+ No need to distribute data across processors before each direct solve
+ Load balance preserved by uniform refinement
- Edge distribution ≠ cell distribution
Multigrid solver – Software design

- Elementary matrices computed “on the fly”
- EM code acts as a service provider (direct solver driven)
- Multigrid library in C
- EM code developed in Fortran

→ Need a coupling code
Multigrid solver – Software design

- EM code
  - Global geo.
  - Local geo. (naive)
  - Local geo. (PaStiX)

- Coupling code
  - FE
  - EF

- Multigrid solver
  - CSC
  - Scotch–Fax–Blend
  - Assembly
  - Factorization
  - Multigrid

- Geo. refinement
- RHS
- Element matrix

- Refinement
- RHS
- Element matrix
Validation – Variable material properties

Sphere test case

- Constant mesh and multigrid cycles (FMG 10 V(2,2))
- Increase k by modifying coating material properties (green material in the graph above)
### Validation – Variable material properties

**Table:** DDL per wavelength, sphere test case

<table>
<thead>
<tr>
<th>freq.</th>
<th>lev.0</th>
<th>lev.1</th>
<th>lev.2</th>
<th>$\epsilon$</th>
<th>$\mu$</th>
<th>n</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>low</td>
<td>12</td>
<td>17</td>
<td>23</td>
<td>2</td>
<td>1</td>
<td>1.41</td>
<td>4.44</td>
</tr>
<tr>
<td>low</td>
<td>6</td>
<td>9</td>
<td>11</td>
<td>4</td>
<td>2</td>
<td>2.82</td>
<td>8.89</td>
</tr>
<tr>
<td>low</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>4</td>
<td>5.64</td>
<td>17.77</td>
</tr>
<tr>
<td>low</td>
<td>12</td>
<td>17</td>
<td>23</td>
<td>1+1i</td>
<td>1+1i</td>
<td>1.41</td>
<td>4.44</td>
</tr>
<tr>
<td>low</td>
<td>6</td>
<td>9</td>
<td>11</td>
<td>2+2i</td>
<td>2+2i</td>
<td>2.82</td>
<td>8.89</td>
</tr>
<tr>
<td>low</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>4+4i</td>
<td>4+4i</td>
<td>5.64</td>
<td>17.77</td>
</tr>
</tbody>
</table>

| unks.      | 14 625 | 57 909 | 231 045 |
| edges      | 11mm   | 8mm    | 6mm     |
Validation – Variable material properties

Multigrid solver:

Figure: MG convergence for lossless material (left) and absorbing material (right)
**Validation – Variable material properties**

Multigrid preconditioned GMRES:

![Graph showing GMRES convergence for lossless material (left) and absorbing material (right)](image)

**Figure:** GMRES convergence for lossless material (left) and absorbing material (right)
Validation – Variable frequency

Haltere test case

- Constant mesh and multigrid cycles (FMG 10 V(2,2))
- Increase k by modifying the incident wave frequency
## Validation – Variable frequency

**Table:** DDL per wavelength, haltere test case (fixed \(n=1.41\)).

<table>
<thead>
<tr>
<th>freq.</th>
<th>lev.0</th>
<th>lev.1</th>
<th>lev.2</th>
<th>lev.3</th>
<th>lev.4</th>
<th>lev.5</th>
<th>n</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 GHz</td>
<td>20</td>
<td>29</td>
<td>38</td>
<td>49</td>
<td>62</td>
<td>78</td>
<td>1.41</td>
<td>29.6</td>
</tr>
<tr>
<td>2 GHz</td>
<td>10</td>
<td>14</td>
<td>19</td>
<td>24</td>
<td>31</td>
<td>39</td>
<td>1.41</td>
<td>59.2</td>
</tr>
<tr>
<td>3 GHz</td>
<td>7</td>
<td>10</td>
<td>13</td>
<td>16</td>
<td>21</td>
<td>26</td>
<td>1.41</td>
<td>88.8</td>
</tr>
<tr>
<td>5 GHz</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>16</td>
<td>1.41</td>
<td>148.1</td>
</tr>
</tbody>
</table>

| unks.  | 1.2M  | 5.2M  | 21.1M | 84.6M | 338.5M | 1.3B  |
| edges  | 10mm  | 7mm   | 5mm   | 4mm   | 3mm   | 2mm   |
Validation – Variable frequency

1 level multigrid (relative residual):

Haltère – $n=1.41$ ($\varepsilon = \mu = 1 + i$)
FMG 10V(2,2) 1 niveau (5.2M inconnues)

[Graph showing residual versus multigrid step for different frequencies (1GHz, 2GHz, 3GHz, 5GHz).]
2 level multigrid (relative residual):

Haltère – n=1.41 (ε= μ= 1 + i)  
FMG 10V(2,2) 2 niveaux (21.1M inconnues)
Validation – Strong scaling

- Haltere test case
- Fixed coarse mesh
- Variables number of cores
- Run on TERA100 nodes (4 Intel Xeon processors, 32 cores/node)
- Use only 2 cores per node (memory constraints)
**Validation – Strong scaling**

**Table:** fine 84.6M – coarse 1.2 M – FMG(2,2) 3 levels

<table>
<thead>
<tr>
<th>Cores</th>
<th>Init.</th>
<th>CSC</th>
<th>Assemb.</th>
<th>Facto.</th>
<th>Solve</th>
<th>10V</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>123.98</td>
<td>0.48</td>
<td>0.42</td>
<td>79.62</td>
<td>0.55</td>
<td>2285.58</td>
</tr>
<tr>
<td>32</td>
<td>118.67</td>
<td>0.31</td>
<td>0.23</td>
<td>82.72</td>
<td>0.30</td>
<td>1369.26</td>
</tr>
<tr>
<td>64</td>
<td>103.10</td>
<td>0.15</td>
<td>0.12</td>
<td>68.18</td>
<td>0.19</td>
<td>710.57</td>
</tr>
<tr>
<td>128</td>
<td>102.69</td>
<td>0.08</td>
<td>0.06</td>
<td>60.94</td>
<td>0.11</td>
<td>399.23</td>
</tr>
<tr>
<td>256</td>
<td>130.32</td>
<td>0.03</td>
<td>0.03</td>
<td>75.27</td>
<td>0.21</td>
<td>194.42</td>
</tr>
<tr>
<td>512</td>
<td>170.97</td>
<td>0.01</td>
<td>0.01</td>
<td>78.81</td>
<td>0.06</td>
<td>100.17</td>
</tr>
</tbody>
</table>

---

**Graph: Halte – FMG(2,2) 3 levels – 84.6M (coarse : 1.2M)**

- **Time (s)**
- **Cores**
- **Prepare**
- **Facto.**
- **1 vcycle**

**Legend:**
- Red: Prepare
- Green: Facto.
- Blue: 1 vcycle

---

M. Chanaud, L. Giraud, D. Goudin, JJ. Pesqué, J. Roman

Parallel multigrid solver, SD11
### Validation – Strong scaling

**Table:** fine 84.6M – coarse 5.2 M – FMG(2,2) 2 levels

<table>
<thead>
<tr>
<th>Cores</th>
<th>Init.</th>
<th>CSC</th>
<th>Assemb.</th>
<th>Facto.</th>
<th>Solve</th>
<th>10V</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>229.34</td>
<td>0.36</td>
<td>0.24</td>
<td>103.04</td>
<td>0.26</td>
<td>354.23</td>
</tr>
<tr>
<td>256</td>
<td>286.71</td>
<td>0.10</td>
<td>0.12</td>
<td>95.40</td>
<td>0.20</td>
<td>174.91</td>
</tr>
<tr>
<td>512</td>
<td>369.69</td>
<td>0.05</td>
<td>0.06</td>
<td>98.56</td>
<td>0.19</td>
<td>85.89</td>
</tr>
<tr>
<td>1024</td>
<td>480.22</td>
<td>0.02</td>
<td>0.03</td>
<td>98.96</td>
<td>0.26</td>
<td>65.88</td>
</tr>
<tr>
<td>2048*</td>
<td>1331.51</td>
<td>0.01</td>
<td>0.02</td>
<td>107.95</td>
<td>0.27</td>
<td>47.29</td>
</tr>
</tbody>
</table>

(*) 4 cores per nodes instead of 2 (16384 reserved cores).
### Validation – Strong scaling

Table: fine 338.5M – coarse 21.1 M – FMG(2,2) 2 levels

<table>
<thead>
<tr>
<th>Cores</th>
<th>Init.</th>
<th>CSC</th>
<th>Assemb.</th>
<th>Facto.</th>
<th>Solve</th>
<th>10V</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>662.22</td>
<td>0.43</td>
<td>0.52</td>
<td>117.96</td>
<td>0.64</td>
<td>628.74</td>
</tr>
<tr>
<td>512</td>
<td>881.20</td>
<td>0.20</td>
<td>0.23</td>
<td>125.56</td>
<td>0.56</td>
<td>345.00</td>
</tr>
<tr>
<td>1024</td>
<td>1196.74</td>
<td>0.13</td>
<td>0.14</td>
<td>151.56</td>
<td>0.56</td>
<td>213.49</td>
</tr>
</tbody>
</table>
Validation – Strong scaling

Table: fine 1.3B – coarse 21.1 M – FMG(2,2) 3 levels

<table>
<thead>
<tr>
<th>Cores</th>
<th>Init.</th>
<th>CSC</th>
<th>Assemb.</th>
<th>Facto.</th>
<th>Solve</th>
<th>10V</th>
<th>(20V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024</td>
<td>1160.28</td>
<td>0.12</td>
<td>0.14</td>
<td>147.73</td>
<td>0.55</td>
<td>857.69</td>
<td>(1715.38)</td>
</tr>
</tbody>
</table>

Haltère 1 milliard d'inconnues (grossier : 21 millions) – FMG 3 niveaux, 20 itérations
Conclusion :

- Parallel multigrid solver fully functionnal and scalable
- Can be used as a solver for simple problems (absorbing materials)
- Can be used as a preconditioner for complex problems (lossless materials)

Perspectives :

- Lossless materials : apply a complex shift to the preconditioner
- Modify refinement to ensure tetrahedron aspect ratio
- Adaptive distribution depending on levels