Advanced Computation:
Computational Electromagnetics

Algebraic Elimination of Portions of FDFD Grids

1

Concept of Algebraic Elimination

The ordinary FDFD method converts Maxwell's equations into a matrix equation.

$$\mathbf{Af} = \mathbf{b}$$

Rows and columns are sorted to partition the matrix equation into points that are to be algebraically eliminated (subscript e) and retained (subscript r) in the matrix equation.

$$\begin{bmatrix} \mathbf{A}_{\mathrm{ee}} & \mathbf{A}_{\mathrm{er}} \\ \mathbf{A}_{\mathrm{re}} & \mathbf{A}_{\mathrm{rr}} \end{bmatrix} \begin{bmatrix} \mathbf{f}_{\mathrm{e}} \\ \mathbf{f}_{\mathrm{r}} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{\mathrm{e}} \\ \mathbf{b}_{\mathrm{r}} \end{bmatrix}$$

After algebraic elimination, the reduced matrix equation is

$$\mathbf{A}' = \mathbf{A}_{\mathrm{rr}} - \mathbf{A}_{\mathrm{re}}\,\mathbf{A}_{\mathrm{ee}}^{-1}\mathbf{A}_{\mathrm{er}}$$
$$\mathbf{b}' = \mathbf{b}_{\mathrm{r}} - \mathbf{A}_{\mathrm{re}}\,\mathbf{A}_{\mathrm{ee}}^{-1}\mathbf{b}_{\mathrm{e}}$$

$\Rightarrow$

$$\mathbf{f}_{\mathbf{r}} = \mathbf{A}'^{-1}\mathbf{b}'$$

EMPossible Slide 2

2

The Derivation

$$\begin{bmatrix} \mathbf{A}_{\text{ee}} & \mathbf{A}_{\text{er}} & \mathbf{b}_{\text{e}} \\ \mathbf{A}_{\text{re}} & \mathbf{A}_{\text{rr}} & \mathbf{b}_{\text{r}} \end{bmatrix}$$

Form augmented matrix

$$\begin{bmatrix} \mathbf{I} & \mathbf{A}_{\text{ee}}^{-1}\mathbf{A}_{\text{er}} & \mathbf{A}_{\text{ee}}^{-1}\mathbf{b}_{\text{e}} \\ \mathbf{A}_{\text{re}} & \mathbf{A}_{\text{rr}} & \mathbf{b}_{\text{r}} \end{bmatrix}$$

Predivide first row by $\mathbf{A}_{\text{ee}}$

$$\begin{bmatrix} \mathbf{A}_{\text{re}} & \mathbf{A}_{\text{re}}\mathbf{A}_{\text{ee}}^{-1}\mathbf{A}_{\text{er}} & \mathbf{A}_{\text{re}}\mathbf{A}_{\text{ee}}^{-1}\mathbf{b}_{\text{e}} \\ \mathbf{A}_{\text{re}} & \mathbf{A}_{\text{rr}} & \mathbf{b}_{\text{r}} \end{bmatrix}$$

Premultiply first row by $\mathbf{A}_{\text{re}}$

$$\begin{bmatrix} \mathbf{A}_{\text{re}} & \mathbf{A}_{\text{re}}\mathbf{A}_{\text{ee}}^{-1}\mathbf{A}_{\text{er}} & \mathbf{A}_{\text{re}}\mathbf{A}_{\text{ee}}^{-1}\mathbf{b}_{\text{e}} \\ \mathbf{0} & \mathbf{A}_{\text{rr}} - \mathbf{A}_{\text{re}}\mathbf{A}_{\text{ee}}^{-1}\mathbf{A}_{\text{er}} & \mathbf{b}_{\text{r}} - \mathbf{A}_{\text{re}}\mathbf{A}_{\text{ee}}^{-1}\mathbf{b}_{\text{e}} \end{bmatrix}$$

Subtract row 1 from row 2

$$\begin{bmatrix} \mathbf{A}_{\text{rr}} - \mathbf{A}_{\text{re}}\mathbf{A}_{\text{ee}}^{-1}\mathbf{A}_{\text{er}} & \mathbf{b}_{\text{r}} - \mathbf{A}_{\text{re}}\mathbf{A}_{\text{ee}}^{-1}\mathbf{b}_{\text{e}} \end{bmatrix}$$

Eliminate first row and first column

$$\mathbf{A}' = \mathbf{A}_{\text{rr}} - \mathbf{A}_{\text{re}}\,\mathbf{A}_{\text{ee}}^{-1}\mathbf{A}_{\text{er}}$$
$$\mathbf{b}' = \mathbf{b}_{\text{r}} - \mathbf{A}_{\text{re}}\,\mathbf{A}_{\text{ee}}^{-1}\mathbf{b}_{\text{e}}$$

Extract new $\mathbf{A}'$ and $\mathbf{b}'$

MATLAB Code for Algebraic Elimination

Let `ind_e` be the linear indices of points to eliminate and `ind_r` be the linear indices of points to retain.

```
% EXTRACT PARTITIONS
Aee = A(ind_e,ind_e);
Aer = A(ind_e,ind_r);
Are = A(ind_r,ind_e);
Arr = A(ind_r,ind_r);
be  = b(ind_e);
br  = b(ind_r);

% CALCULATE NEW MATRICES
A2 = Arr - Are/Aee*Aer;
b2 = br - Are/Aee*be;
```

Implementation Notes

- Don't forget to modify the number of points on grid, axis vectors, etc.
- Consider eliminating PML regions.

Alternative Code for Algebraic Elimination

The elimination procedure can be done row-by-row. This may be advantageous in other programming languages.

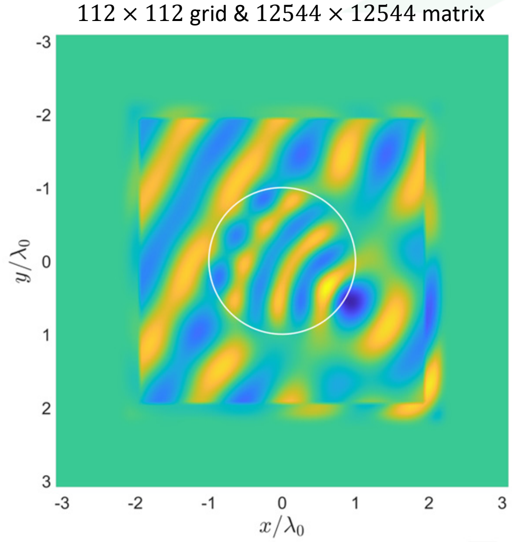
```
% ABSORPTION PROCEDURE
A2 = [ A b ];
for ind = 1 : length(ind_e)
    m = ind_e(ind);
    A2(m,:) = A2(m,:) / A2(m,m);
    A2 = A2 - A2(:,m)*A2(m,:);
    A2(m,:) = 0;
    A2(:,m) = 0;
end
A2(ind_e,:) = [];
A2(:,ind_e) = [];
M = length(A2(:,1));
b2 = A2(:,M+1);
A2 = A2(:,1:M);
```

Implementation Notes

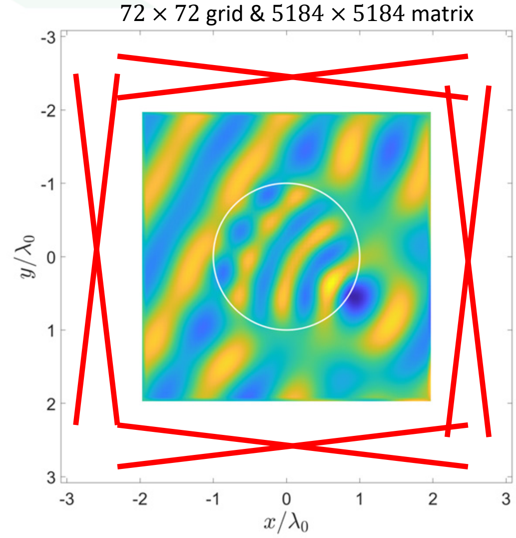
- Don't forget to modify the number of points on grid, axis vectors, etc.
- Consider eliminating PML regions.

Example – Eliminating PML Regions

112 × 112 grid & 12544 × 12544 matrix



72 × 72 grid & 5184 × 5184 matrix



83% reduction in matrix size