MAT 1750

A Fast Multipole Algorithm for the Helmholtz Equation in 3D

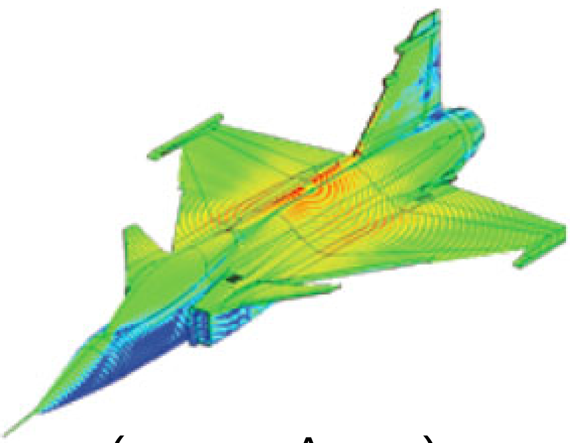
Shashwat Sharma

Dec 04, 2018

Background

- Boundary integral equation methods:
  - ✔ Dimensionality reduction
  - ✔ Linear system of equations
  - ✗ Large problems --> dense matrices
    - ✗ Storage
    - ✗ Factorization
- Acceleration methods needed

(source: FEKO)



(source: Ansys)

Formulation

- Helmholtz equation in 3D:

$$\nabla^2 \vec{A}(\vec{r}) + k^2 \vec{A}(\vec{r}) = \vec{f}(\vec{r})$$

- In the context of electromagnetics:

$$\nabla^2 \vec{A}(\vec{r}) + k^2 \vec{A}(\vec{r}) = -\vec{J}(\vec{r})\mu$$

Formulation

- Helmholtz equation in 3D:

$$\nabla^2 \vec{A}\left(\vec{r}\right) + k^2 \vec{A}\left(\vec{r}\right) = \vec{f}\left(\vec{r}\right)$$

- In the context of electromagnetics:

$$\nabla^2 \vec{A}\left(\vec{r}\right) + k^2 \vec{A}\left(\vec{r}\right) = -\vec{J}\left(\vec{r}\right)$$

- We want to solve for the associated electric field:

$$\vec{E}\left(\vec{r}\right) = -j\omega\mu \int_{\mathcal{S}} dS' \, \overline{\overline{\mathbf{G}}}\left(\vec{r}, \vec{r}'\right) \cdot \vec{J}\left(\vec{r}'\right)$$

Formulation

- Helmholtz equation in 3D:

$$\nabla^2 \vec{A}\left(\vec{r}\right) + k^2 \vec{A}\left(\vec{r}\right) = \vec{f}\left(\vec{r}\right)$$

- In the context of electromagnetics:

$$\nabla^2 \vec{A}\left(\vec{r}\right) + k^2 \vec{A}\left(\vec{r}\right) = -\vec{J}\left(\vec{r}\right)$$

- We want to solve for the associated electric field:

$$\vec{E}\left(\vec{r}\right) = -j\omega\mu \int_{\mathcal{S}} dS' \,\overline{\overline{\mathbf{G}}}\left(\vec{r}, \vec{r}'\right) \cdot \vec{J}\left(\vec{r}'\right)$$

- The dyadic Green's function can be simplified:

$$\vec{E}\left(\vec{r}\right) = -j\omega\mu \int_{\mathcal{S}} dS' \left(1 + \frac{\nabla\nabla\cdot}{k^2}\right) G\left(\vec{r}, \vec{r}'\right) \vec{J}\left(\vec{r}'\right)$$

Acceleration

- 3D scalar Green's function:

$$G\left(\vec{r}, \vec{r}'\right) = \frac{\mathrm{e}^{-jk|\vec{r}-\vec{r}'|}}{|\vec{r}-\vec{r}'|}$$

Acceleration

- 3D scalar Green's function:

$$G\left(\vec{r},\vec{r}'\right) = \frac{\mathrm{e}^{-jk|\vec{r}-\vec{r}'|}}{|\vec{r}-\vec{r}'|}$$

- For simple, perfect electric conductors:

$$\underbrace{-j\omega\mu\int_{\mathcal{S}} dS'\left(1+\frac{\nabla\nabla\cdot}{k^2}\right)G\left(\vec{r},\vec{r}'\right)}_{\mathbf{A}}\underbrace{\vec{J}\left(\vec{r}'\right)}_{\mathbf{x}} = \underbrace{\vec{E}_{\mathrm{inc}}\left(\vec{r}\right)}_{\mathbf{b}}$$

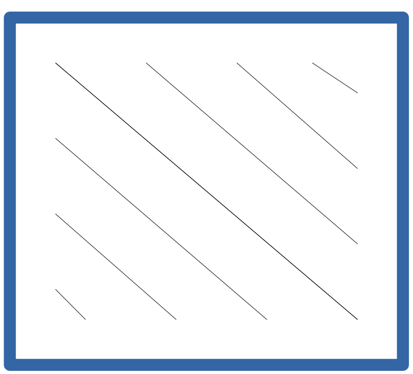
Acceleration

- 3D scalar Green's function:

$$G\left(\vec{r},\vec{r}\,'\right) = \frac{\mathrm{e}^{-jk\left|\vec{r}-\vec{r}\,'\right|}}{\left|\vec{r}-\vec{r}\,'\right|}$$

- For simple, perfect electric conductors:

$$\underbrace{-j\omega\mu\int_{S} dS'\left(1+\frac{\nabla\nabla\cdot}{k^2}\right) G\left(\vec{r},\vec{r}\,'\right)}_{\mathbf{A}}\underbrace{\vec{J}\left(\vec{r}\,'\right)}_{\mathbf{x}} = \underbrace{\vec{E}_{\mathrm{inc}}\left(\vec{r}\right)}_{\mathbf{b}}$$



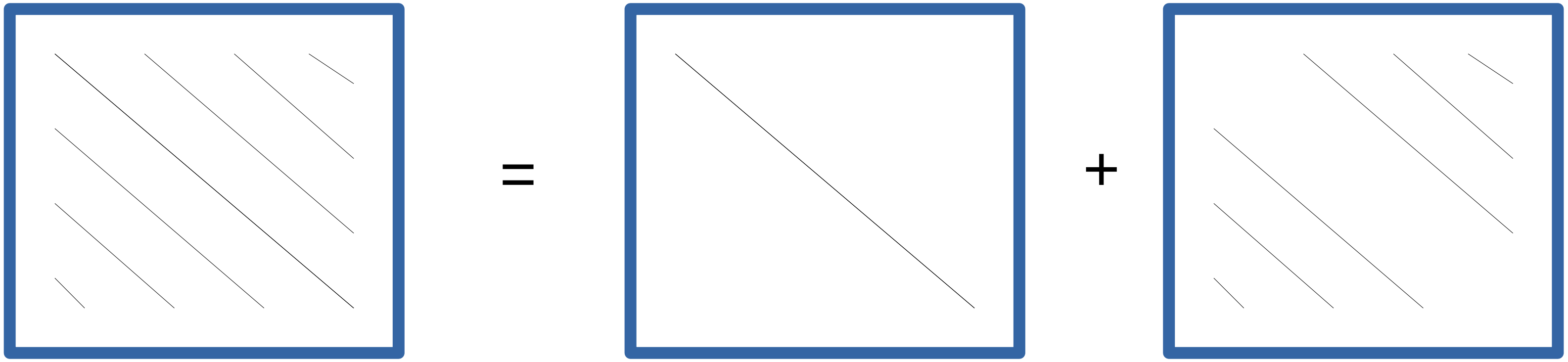
Acceleration

- 3D scalar Green's function:

$$G\left(\vec{r},\vec{r}'\right)=\frac{\mathrm{e}^{-jk|\vec{r}-\vec{r}'|}}{|\vec{r}-\vec{r}'|}$$

- For simple, perfect electric conductors:

$$\underbrace{-j\omega\mu\int_S dS'\left(1+\frac{\nabla\nabla\cdot}{k^2}\right)G\left(\vec{r},\vec{r}'\right)}_{\mathbf{A}}\underbrace{\vec{J}\left(\vec{r}'\right)}_{\mathbf{x}}=\underbrace{\vec{E}_{\mathrm{inc}}\left(\vec{r}\right)}_{\mathbf{b}}$$



Fast Multipole Algorithm (FMA)

- 3D scalar Green's function:

$$G\left(\vec{r},\vec{r}'\right) = \frac{\mathrm{e}^{-jk\left|\vec{r}-\vec{r}'\right|}}{\left|\vec{r}-\vec{r}'\right|}$$

- Can be decomposed:

$$\frac{\mathrm{e}^{-jk\left|\vec{r}-\vec{r}'\right|}}{\left|\vec{r}-\vec{r}'\right|} = \int_1 T\left(\vec{r}',\hat{n}_r\right) T_L\left(k,\hat{n}_r,\vec{r}_{ab}\right) R\left(\vec{r},\hat{n}_r\right) dS$$

Fast Multipole Algorithm (FMA)

- 3D scalar Green's function:

$$G\left(\vec{r},\vec{r}\,'\right) = \frac{\mathrm{e}^{-jk|\vec{r}-\vec{r}\,'|}}{|\vec{r}-\vec{r}\,'|}$$

- Can be decomposed:

$$\frac{\mathrm{e}^{-jk|\vec{r}-\vec{r}\,'|}}{|\vec{r}-\vec{r}\,'|} = \int_1 T\left(\vec{r}\,',\hat{n}_r\right) T_L\left(k,\hat{n}_r,\vec{r}_{ab}\right) R\left(\vec{r},\hat{n}_r\right) dS$$

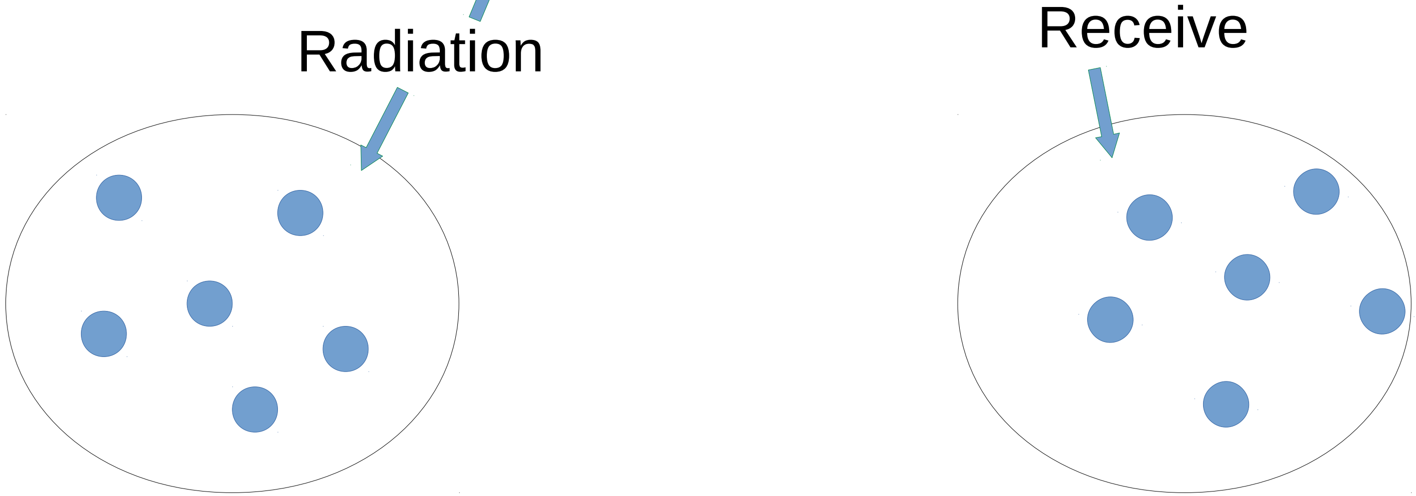
Fast Multipole Algorithm (FMA)

- 3D scalar Green's function:

$$\color{green}{G\left(\vec{r},\vec{r}\,'\right)} = \frac{\mathrm{e}^{-jk\left|\vec{r}-\vec{r}\,'\right|}}{\left|\vec{r}-\vec{r}\,'\right|}$$

- Can be decomposed:

$$\frac{\mathrm{e}^{-jk\left|\vec{r}-\vec{r}\,'\right|}}{\left|\vec{r}-\vec{r}\,'\right|} = \int_{1} T\left(\vec{r}\,',\hat{n}_r\right) T_L\left(k,\hat{n}_r,\vec{r}_{ab}\right) R\left(\vec{r},\hat{n}_r\right) dS$$



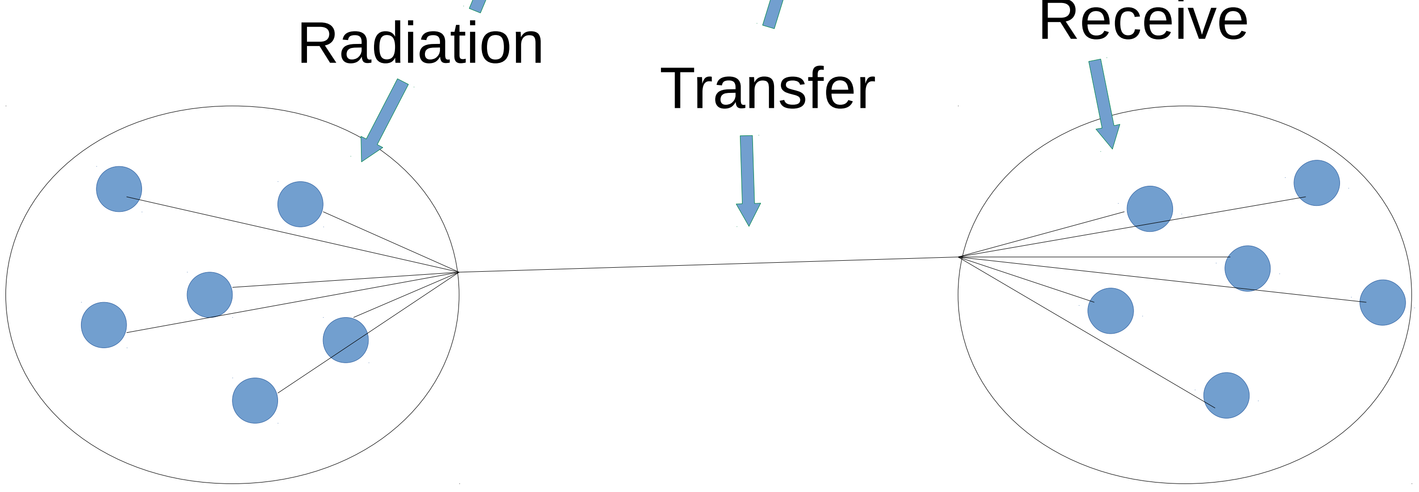
Fast Multipole Algorithm (FMA)

- 3D scalar Green's function:

$$G\left(\vec{r}, \vec{r}'\right) = \frac{\mathrm{e}^{-jk|\vec{r}-\vec{r}'|}}{|\vec{r}-\vec{r}'|}$$

- Can be decomposed:

$$\frac{\mathrm{e}^{-jk|\vec{r}-\vec{r}'|}}{|\vec{r}-\vec{r}'|} = \int_1 T\left(\vec{r}', \hat{n}_r\right) T_L\left(k, \hat{n}_r, \vec{r}_{ab}\right) R\left(\vec{r}, \hat{n}_r\right) dS$$



Fast Multipole Algorithm (FMA)

- 3D scalar Green's function:

$$\frac{\mathrm{e}^{-jk|\vec{r}-\vec{r}'|}}{|\vec{r}-\vec{r}'|} = \int_1 T\left(\vec{r}', \hat{n}_r\right) T_L\left(k, \hat{n}_r, \vec{r}_{ab}\right) R\left(\vec{r}, \hat{n}_r\right) dS$$

- With radiation, receive and transfer functions:

$$T\left(\vec{f}_n, \hat{n}_r\right) = \left[1 - \hat{n}_r \hat{n}_r \cdot\right] \int_{\vec{f}_n} \vec{f}_n\left(\vec{r}'\right) \mathrm{e}^{jk\hat{n}_r \cdot \vec{r}'} d\vec{r}'$$

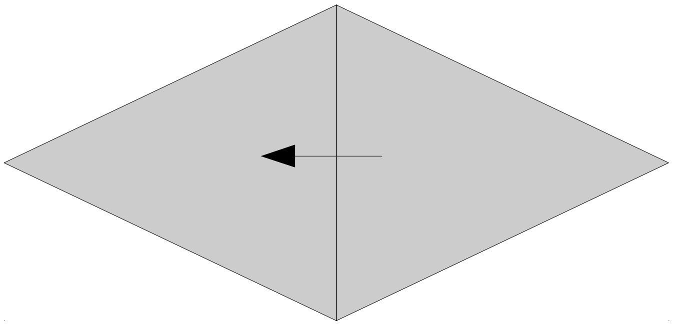
$$R\left(\vec{f}_m, \hat{n}_r\right) = j\omega\mu \int_{\vec{f}_m} \vec{f}_m\left(\vec{r}\right) \mathrm{e}^{-jk\hat{n}_r \cdot \vec{r}} d\vec{r}$$

$$T_L\left(k, \hat{n}_r, \vec{r}_{ab}\right) = \frac{k}{(4\pi)^2} \sum_{l=0}^{L} (-j)^{l+1} (2l+1)\, h_l^{(2)}\left(k\left|\vec{r}_{ab}\right|\right) P_l\left(\hat{n}_r \cdot \vec{r}_{ab}\right)$$

Fast Multipole Algorithm (FMA)

- Surface discretization:

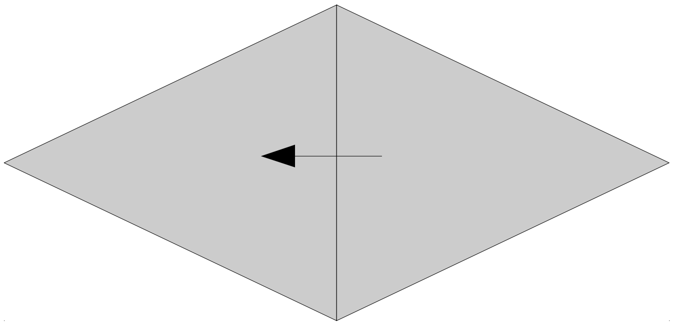
Raviart-Thomas
basis functions



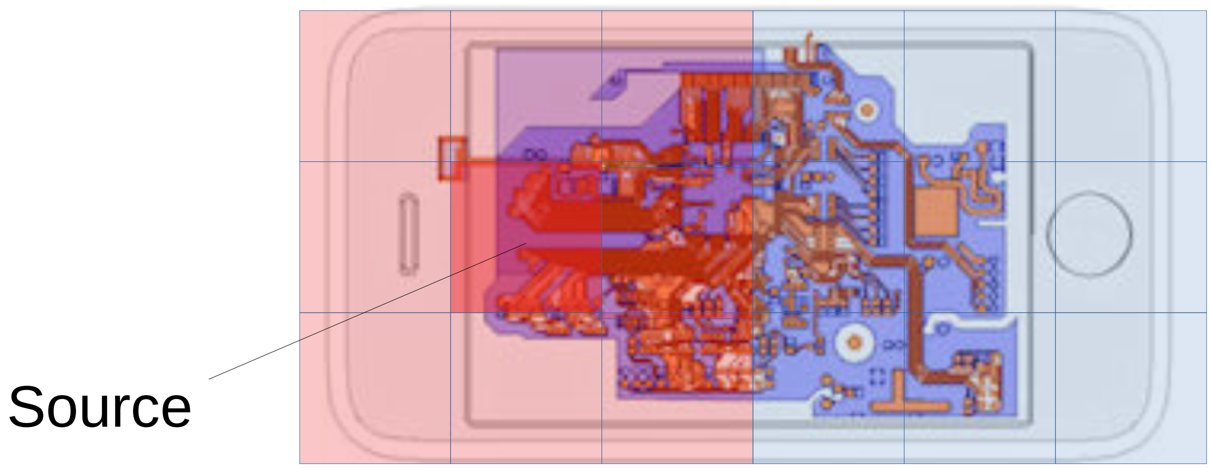
Fast Multipole Algorithm (FMA)

- Surface discretization:

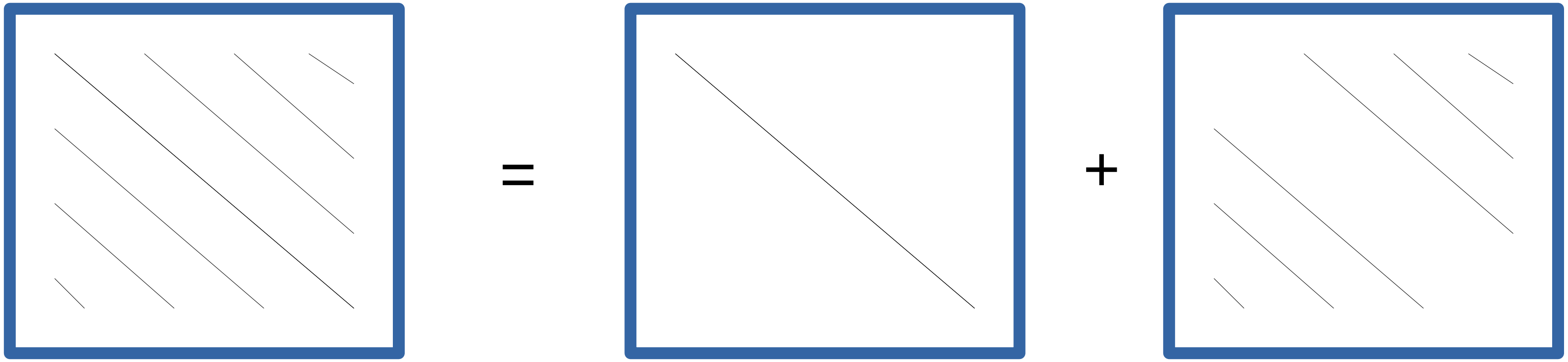
Raviart-Thomas
basis functions



- Grouping of basis functions:



Fast Multipole Algorithm (FMA)



Already have:

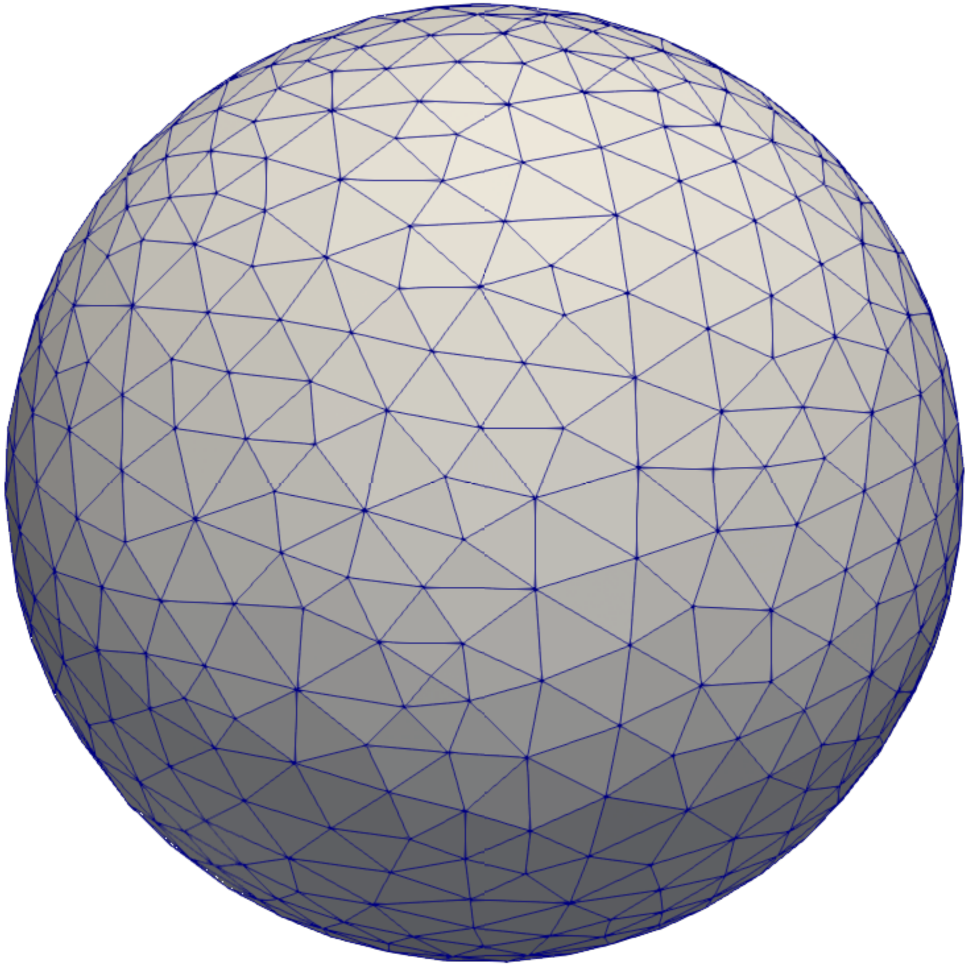
- Meshing tools
- Integration over basis functions
- Direct solver for dense matrices

This project:

- FMA cube subdivisions
- Transfer, radiation, receive functions
- Iterative solver

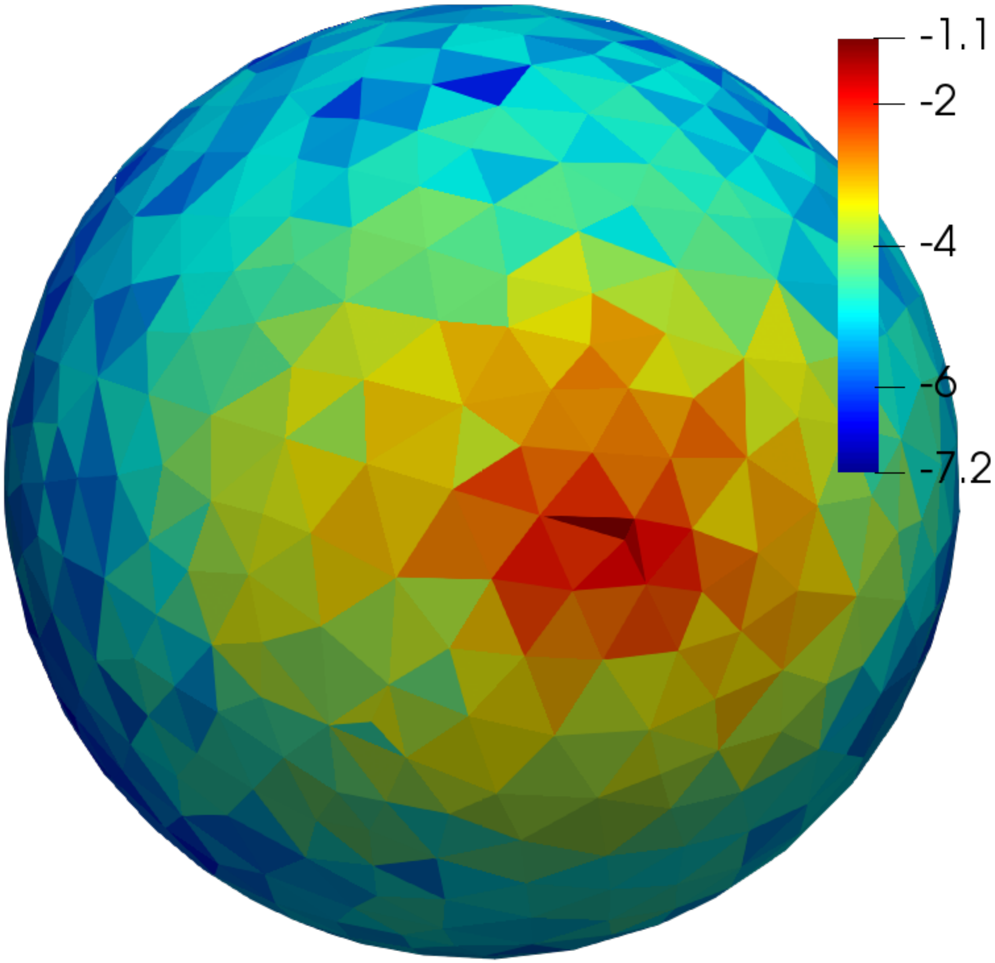
Test Case

3D sphere with 4120 basis functions



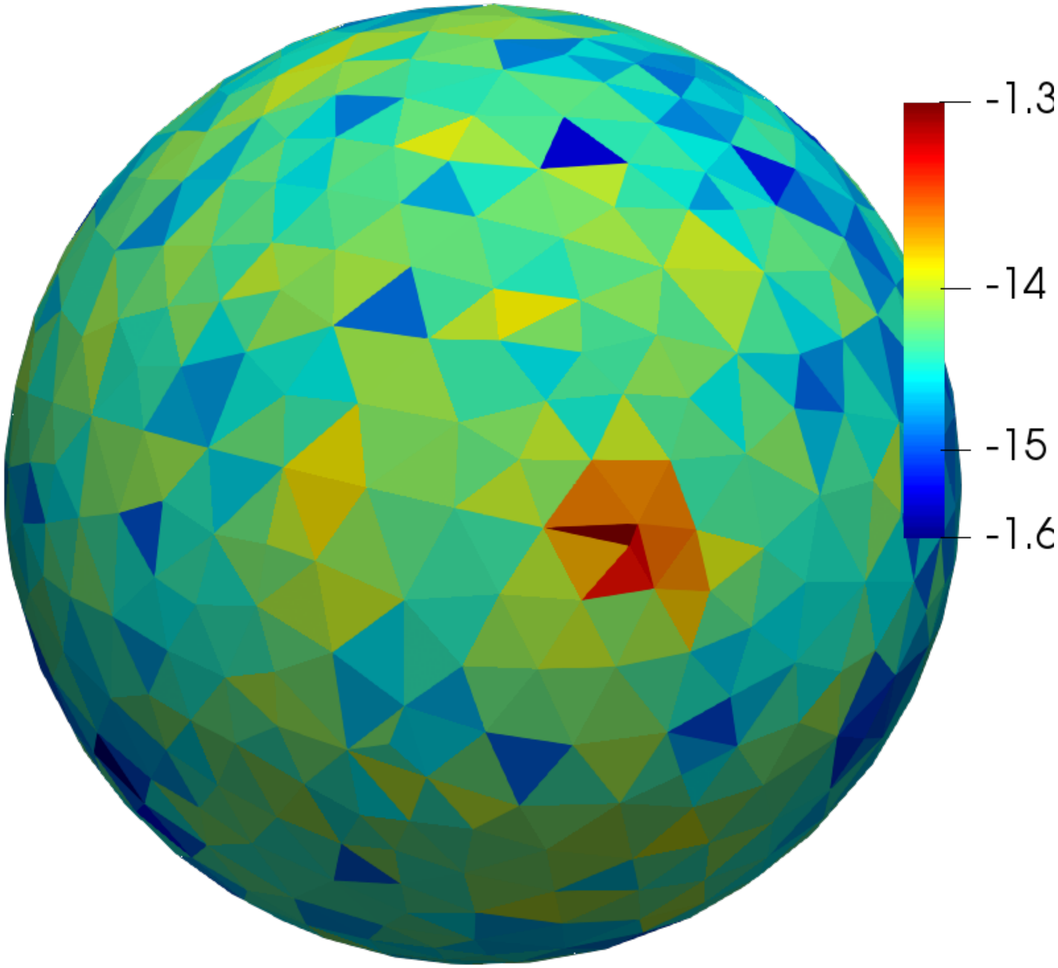
Test Case

Solution (current density) from direct solver



Test Case

Solution (current density) with FMA



Test Case

Iterative solver with FMA – Issues:

- Convergence

  ➔ Solution only converged to 30% residual norm error

  - Tried diagonal and ILU preconditioners

Test Case

Iterative solver with FMA – Issues:

- Convergence
  - ➔ Solution only converged to 30% residual norm error
    - Tried diagonal and ILU preconditioners
  - ➔ Well-known conditioning issues with the matrix

$$\underbrace{-j\omega\mu \int_S dS' \left(1 + \frac{\nabla\nabla\cdot}{k^2}\right) G\left(\vec{r}, \vec{r}'\right)}_{\mathbf{A}} \underbrace{\vec{J}\left(\vec{r}'\right)}_{\mathbf{x}} = \underbrace{\vec{E}_{\text{inc}}\left(\vec{r}\right)}_{\mathbf{b}}$$

Test Case

Iterative solver with FMA – Issues:

- Convergence
  - ➔ Solution only converged to 30% residual norm error
    - Tried diagonal and ILU preconditioners
  - ➔ Well-known conditioning issues with the matrix

$$\underbrace{-j\omega\mu \int_S dS' \left(1 + \frac{\nabla\nabla\cdot}{k^2}\right) G\left(\vec{r}, \vec{r}'\right)}_{\mathbf{A}} \underbrace{\vec{J}\left(\vec{r}'\right)}_{\mathbf{x}} = \underbrace{\vec{E}_{\text{inc}}\left(\vec{r}\right)}_{\mathbf{b}}$$

  - ➔ Formulation itself needs improvement
    - cond(A) ~ 10^14