Interior Penalty Discontinuous Galerkin Time Domain Methods

Stylianos Dosopoulos and Jin-Fa Lee

ElectroScience Laboratory, ECE Department, The Ohio State University

April 27, 2010

Overview of Maxwell Equations, Forms, and Energy Densities (Dual Pairings)

Forms

0-form (pt)	ϕ
1-form (length)	E, H
2-form (area)	D, B, J, M
3-form (volume)	ρ

$$\nabla \times \mathbf{E} = -j\omega\mathbf{B} \qquad \mathbf{E},\mathbf{H}: \text{ fields } \in \mathbf{H}(curl;\Omega)$$

$$\nabla \times \mathbf{H} = \mathbf{J} + j\omega\mathbf{D} \qquad \mathbf{J},\mathbf{D},\mathbf{B}: \text{ flux densities } \in \mathbf{H}(div;\Omega)$$

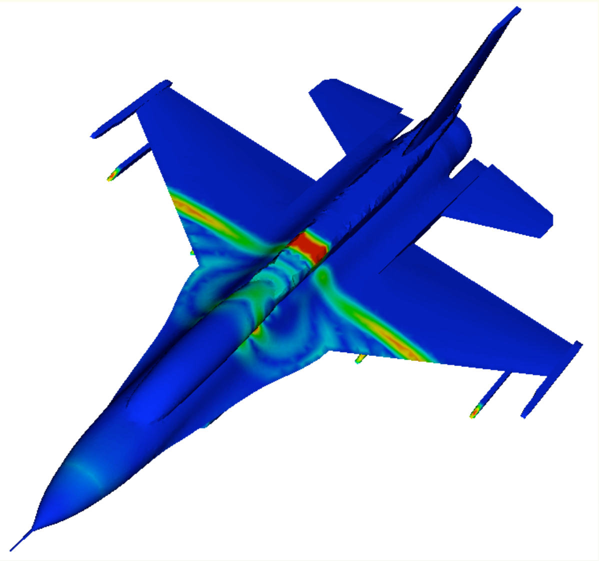
$$\nabla \cdot \mathbf{B} = 0 \qquad \mathbf{u} \in \mathbf{H}(curl;\Omega) \leftrightarrow \mathbf{u} \text{ and } \nabla \times \mathbf{u} \in \mathbf{L}^2(\Omega)$$

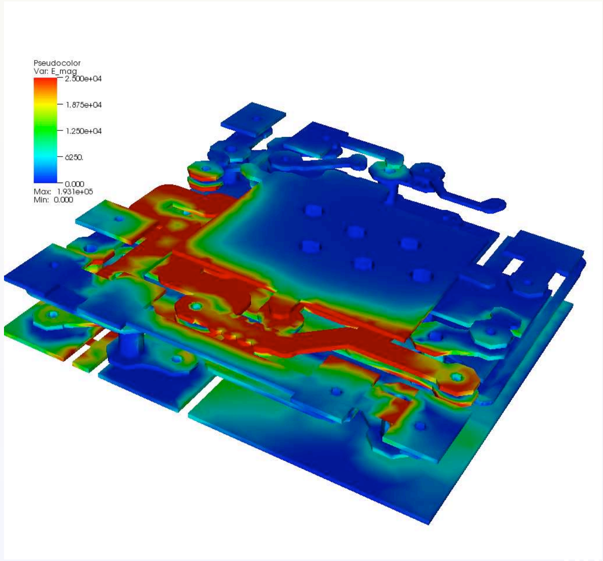
$$\nabla \cdot \mathbf{D} = \rho \qquad \mathbf{u} \in \mathbf{H}(div;\Omega) \leftrightarrow \mathbf{u} \in \mathbf{L}^2(\Omega) \text{ and } \nabla \cdot \mathbf{u} \in L^2(\Omega)$$

$$\left.\begin{matrix}\text{Energy Densities:} \\ \rho\phi,\ E\bullet D,\ H\bullet B,\ E\bullet J,\ H\bullet M\end{matrix}\right\} \xrightarrow{\text{dual pairing}} \text{p form pairs with 3-p form}$$

Outline

- Motivation
- Formulation
- Numerical Experiments
  - Central vs Upwind Flux
  - Conformal PML in DGTD
  - Late time instability in PML
  - Lumped Element Modeling in DGTD
  - Multi-Scale Simulations
- Conclusions

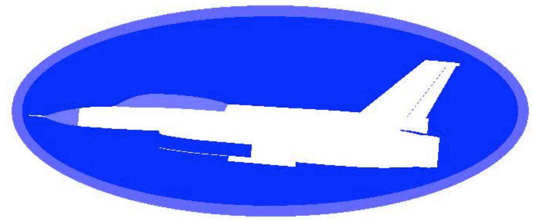




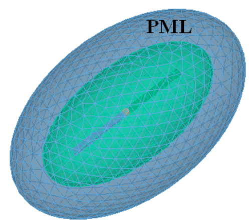
Features of DG Methods

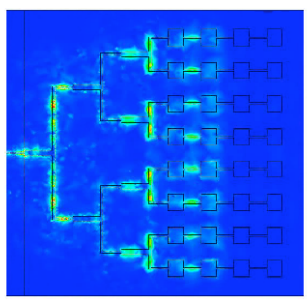
- General Principles of DG Methods
  - Partition the computational domain into polyhedra
  - In each polyhedron the field is represented as a linear combination of a local set of basis
  - Interelement continuity at polyhedra interfaces is weakly enforced
- DG Pros
  - Explicit time marching schemes in time-domain
  - Non-conformal meshes
  - Easier *hp*− refinement
  - High parallel efficiency
- DG Cons
  - High number of degrees of freedom.

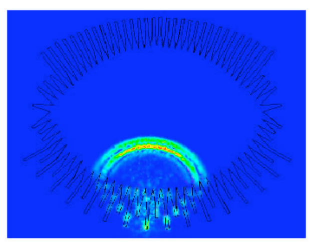
- General Principal DG Methods
  - Partition the computational domain into polyhedra
  - In each polyhedron the field is represented as a linear combination of a local set of basis
  - Interelement continuity at polyhedra interfaces is weakly enforced
- DG Pros
  - Explicit time marching schemes in time-domain
  - Non-conformal meshes
  - Easier *hp*− refinement
  - High parallel efficiency
- DG Cons
  - High number of degrees of freedom.



The Proposed Approach



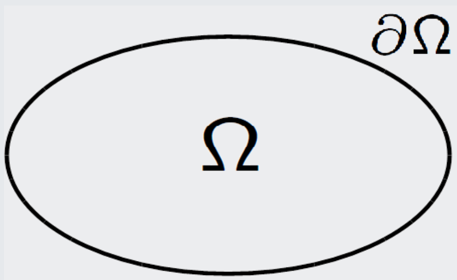




- The derived DGTD method is explicit and conditionally stable.
- A matrix-free memory efficient implementation is applied. There is no assembly and storage of global matrices. All updates are performed at element level.
- Interior Penalty (IP) derivation provides multiple formulations
- A Conformal PML is applied to reduce the buffer space
- A Local Time-Stepping is strategy is applied to increase computational efficiency especially for multi-scale applications

BVP Statement

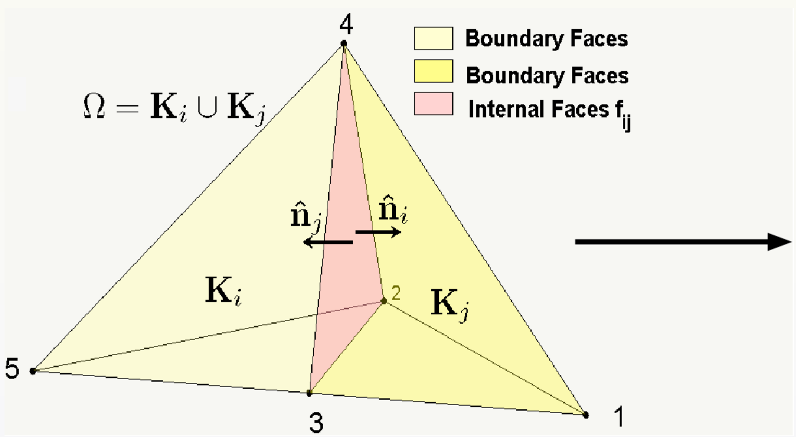
Original BVP(No free sources $\sigma = 0$)



$$\nabla \times \mathbf{E} = \frac{-\mu \partial \mathbf{H}}{\partial t} \quad in\, \Omega$$

$$\nabla \times \mathbf{H} = \frac{\epsilon \partial \mathbf{E}}{\partial t} \quad in\, \Omega$$

$$\mathbf{n}_{ext} \times \mathbf{E} = 0 \quad on\, \partial\Omega$$



Decomposed BVP

$$\nabla \times \mathbf{E}_i = \frac{-\mu_i \partial \mathbf{H}_i}{\partial t} \quad in\, K_i$$

$$\nabla \times \mathbf{H}_i = \frac{\epsilon_i \partial \mathbf{E}_i}{\partial t} \quad in\, K_i$$

$$\nabla \times \mathbf{E}_j = \frac{-\mu_j \partial \mathbf{H}_j}{\partial t} \quad in\, K_j$$

$$\nabla \times \mathbf{H}_j = \frac{\epsilon_j \partial \mathbf{E}_j}{\partial t} \quad in\, K_j$$

$$\hat{\mathbf{n}}_i \times \mathbf{E}_i = -\hat{\mathbf{n}}_j \times \mathbf{E}_j \quad on\, \Gamma$$

$$\hat{\mathbf{n}}_i \times \mathbf{H}_i = -\hat{\mathbf{n}}_j \times \mathbf{H}_i \quad on\, \Gamma$$

$$\mathbf{n}_{ext} \times \mathbf{E}_{i,j} = 0 \quad on\, \partial\Omega$$

Notation: $\gamma_\tau(\mathbf{u}_i) = \hat{\mathbf{n}}_i \times \mathbf{u}_i|_{\partial\Omega_i}$

Notation: $\pi_\tau(\mathbf{u}_i) = \hat{\mathbf{n}}_i \times (\mathbf{u}_i \times \hat{\mathbf{n}}_i)|_{\partial\Omega_i}$

The Galerkin Statement

Residuals

$$\mathbf{R}^{(1)}_{K_i} = \nabla \times \mathbf{E}_i + \frac{\mu \partial \mathbf{H}_i}{\partial t} \quad \in \mathbf{H}(\mathrm{div}, K_i)\ (\mathbf{B}^{err}_i)$$

$$\mathbf{R}^{(2)}_{K_i} = \nabla \times \mathbf{H}_i - \frac{\epsilon \partial \mathbf{E}_i}{\partial t} \quad \in \mathbf{H}(\mathrm{div}, K_i)\ (\mathbf{D}^{err}_i)$$

$$\mathbf{R}^{(3)}_{K_j} = \nabla \times \mathbf{E}_j + \frac{\mu \partial \mathbf{H}_j}{\partial t} \quad \in \mathbf{H}(\mathrm{div}, K_j)\ (\mathbf{B}^{err}_j)$$

$$\mathbf{R}^{(4)}_{K_j} = \nabla \times \mathbf{H}_j - \frac{\epsilon \partial \mathbf{E}_j}{\partial t} \quad \in \mathbf{H}(\mathrm{div}, K_j)\ (\mathbf{D}^{err}_j)$$

$$\mathbf{R}^{(5)}_{\Gamma} = \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j \quad \in \mathbf{H}^{-1/2}(\mathrm{div}_\tau, \Gamma)\ (\mathbf{m}^{err}_s)$$

$$\mathbf{R}^{(6)}_{\Gamma} = \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j \quad \in \mathbf{H}^{-1/2}(\mathrm{div}_\tau, \Gamma)\ (\mathbf{j}^{err}_s)$$

Residual	Physical Meaning	Duality Testing	Energy Term
$\mathbf{R}^{(1)}_{K_i}$	time changing $\mathbf{B}^{err}_i$	$\mathbf{H}(\mathrm{curl}, K_j)(\mathbf{H})$	$\mathbf{H} \cdot \mathbf{B}$
$\mathbf{R}^{(2)}_{K_i}$	time changing $\mathbf{D}^{err}_i$	$\mathbf{H}(\mathrm{curl}, K_j)(\mathbf{E})$	$\mathbf{E} \cdot \mathbf{D}$
$\mathbf{R}^{(3)}_{K_j}$	time changing $\mathbf{B}^{err}_i$	$\mathbf{H}(\mathrm{curl}, K_j)(\mathbf{H})$	$\mathbf{H} \cdot \mathbf{B}$
$\mathbf{R}^{(4)}_{K_j}$	time changing $\mathbf{D}^{err}_i$	$\mathbf{H}(\mathrm{curl}, K_j)(\mathbf{E})$	$\mathbf{E} \cdot \mathbf{D}$
$\mathbf{R}^{(5)}_{\Gamma}$	$\mathbf{m}^{err}_i$	$\mathbf{H}^{-1/2}(\mathrm{curl}_\tau, K_j)(\pi_\tau(\mathbf{E}))$	$\mathbf{M} \cdot \mathbf{H}$
$\mathbf{R}^{(6)}_{\Gamma}$	$\mathbf{j}^{err}_i$	$\mathbf{H}^{-1/2}(\mathrm{curl}_\tau, K_j)(\pi_\tau(\mathbf{H}))$	$\mathbf{J} \cdot \mathbf{E}$

Weighted Residuals with Interior Penalty

- A summation of weighted residuals can be written as

$$
\begin{aligned}
&(\mathbf{w}_i, \mathbf{R}^{(1)}_{K_i})_{K_i} - (\mathbf{v}_i, \mathbf{R}^{(2)}_{K_i})_{K_i} \\
&+ (\mathbf{w}_j, \mathbf{R}^{(3)}_{K_j})_{K_j} - (\mathbf{v}_j, \mathbf{R}^{(4)}_{K_j})_{K_j} \\
&+ c\,\langle \pi_\tau(\mathbf{v}_i) + \pi_\tau(\mathbf{v}_j), \mathbf{R}^{(6)}_{\Gamma} \rangle_\Gamma + d\,\langle \pi_\tau(\mathbf{w}_i) + \pi_\tau(\mathbf{w}_j), \mathbf{R}^{(5)}_{\Gamma} \rangle_\Gamma \\
&+ e\,\langle \gamma_\tau(\mathbf{v}_i) + \gamma_\tau(\mathbf{v}_j), \mathbf{R}^{(5)}_{\Gamma} \rangle_\Gamma + f\,\langle \gamma_\tau(\mathbf{w}_i) + \gamma_\tau(\mathbf{w}_j), \mathbf{R}^{(6)}_{\Gamma})_\Gamma = 0
\end{aligned}
$$

- $V_h^k = \{\mathbf{v} \in [\mathbf{L}^2(\Omega)]^3 : \mathbf{v}|_K \in [\mathbf{P}^k(K)]^3, \quad \forall K \in \mathcal{T}_h\}$

Remarks:

- The choice of c, d, e and f will define a corresponding numerical flux
- The choice of numerical flux can drastically affect the convergence rate and the numerical dispersion and dissipation of the final formulation.

IP-DGTD Formulation

Weak Problem Statement

Find $(\mathbf{H},\mathbf{E}) \in V_h^k \times V_h^k$ *such that*

$$\underbrace{\int_\Omega \mathbf{w}\cdot(\nabla\times\mathbf{E}+\frac{\mu\partial\mathbf{H}}{\partial t})\,d\Omega-\int_\Omega \mathbf{v}\cdot(\nabla\times\mathbf{H}-\frac{\epsilon\partial\mathbf{E}}{\partial t})\,d\Omega}_{\text{VolumeTerms}}+\underbrace{\int_{\mathcal{F}_h}\{\!\{\mathbf{v}\}\!\}\cdot[\![\mathbf{H}]\!]_\tau\,ds}_{SurfaceTerms}$$

$$\underbrace{-\int_{\mathcal{F}_h}\{\!\{\mathbf{w}\}\!\}\cdot[\![\mathbf{E}]\!]_\tau\,ds-e\int_{\mathcal{F}_h}[\![\mathbf{v}]\!]_\tau\cdot[\![\mathbf{E}]\!]_\tau\,ds-f\int_{\mathcal{F}_h}[\![\mathbf{w}]\!]_\tau\cdot[\![\mathbf{H}]\!]_\tau\,ds}_{SurfaceTerms}=0$$

Coefficients

- $\{\!\{\mathbf{u}\}\!\}=(\pi_\tau(\mathbf{u}_i)+\pi_\tau(\mathbf{u}_j))/2$ and $[\![\mathbf{u}]\!]_\tau=\gamma_\tau(\mathbf{u}_i)+\gamma_\tau(\mathbf{u}_j)$
- $c=-d=1/2$ and $e=f=0$ will give rise to a conservative formulation but suboptimal convergence(**central flux**)
- $c=-d=1/2$ and $e=\frac{\gamma}{Z_\Gamma}$ and $f=\frac{\gamma}{Y_\Gamma}$ with $Z_\Gamma=\frac{1}{2}(\sqrt{\frac{\mu_i}{\epsilon_i}}+\sqrt{\frac{\mu_j}{\epsilon_j}})$ and $Y_\Gamma=\frac{1}{2}(\sqrt{\frac{\epsilon_i}{\mu_i}}+\sqrt{\frac{\epsilon_j}{\mu_j}})$ will give rise to a lossy formulation but optimal convergence (**upwind flux**)
- Desretize in time using leap-frog scheme

$$\mathbf{M}_\epsilon\mathbf{e}_i^{n+1}=(\mathbf{M}_\epsilon+e\delta t\mathbf{P}_e)\mathbf{e}_i^n+\delta t(\mathbf{S}_e-\mathbf{F}_e^{ii})\mathbf{h}_i^{n+\frac{1}{2}}-\delta t\mathbf{F}_e^{ij}\mathbf{h}_j^{n+\frac{1}{2}}+e\delta t\mathbf{P}_e^{ij}\mathbf{e}_j^n$$

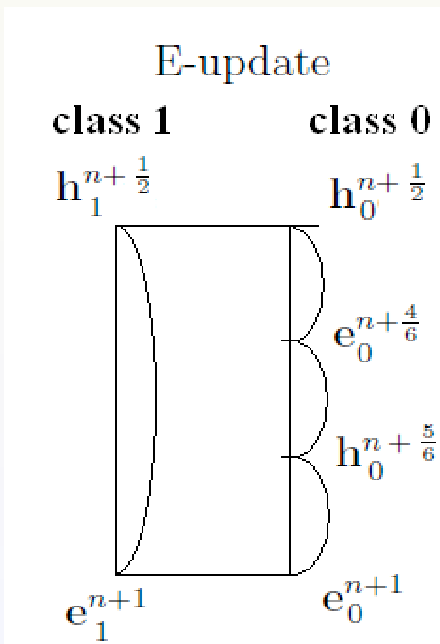
$$\mathbf{M}_\mu\mathbf{h}_i^{n+\frac{3}{2}}=(\mathbf{M}_\mu+f\delta t\mathbf{P}_h)\mathbf{h}_i^{n+\frac{1}{2}}+\delta t(-\mathbf{S}_h+\mathbf{F}_h^{ii})\mathbf{e}_i^{n+1}+\delta t\mathbf{F}_h^{ij}\mathbf{e}_j^{n+1}+f\delta t\mathbf{P}_h^{ij}\mathbf{h}_j^{n+\frac{1}{2}}$$

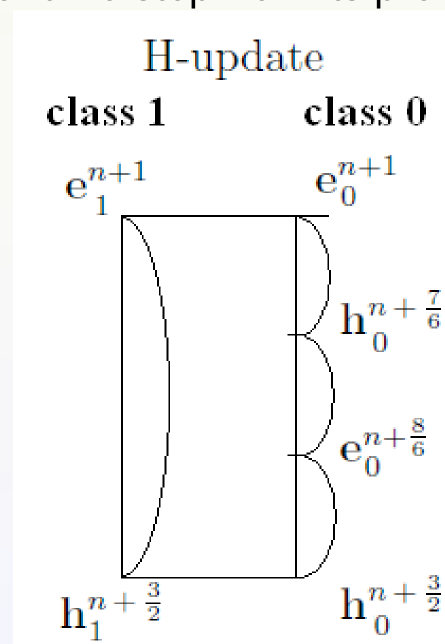
Stability Condition - Local Time-Stepping Update [1]

Stability Condition (S. Piperno)

$$\forall i, \forall k \;, c_i \delta t[2\alpha_i + \beta_{ik} max(\sqrt{\frac{\mu_i}{\mu_k}}, \sqrt{\frac{\epsilon_i}{\epsilon_k}})] < \frac{4V_i}{P_i}$$

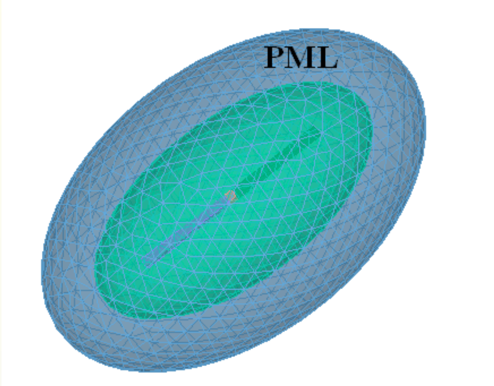
- The set of elements is partitioned into N classes. This partition is done before the time-marching simulation and is based on the stability condition
- For the i^{th} class $\delta t_i = (2m+1)^i \delta t_{min}$
- We choose $m = 1$ so each class has three times larger time step from its previous class

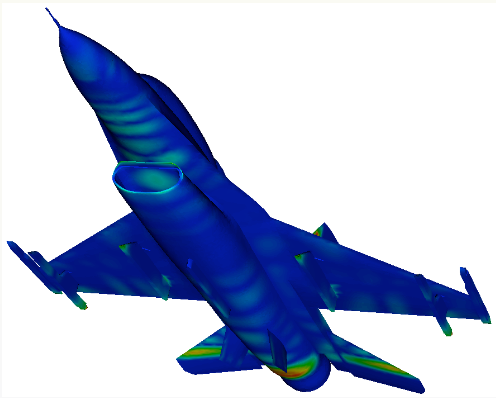


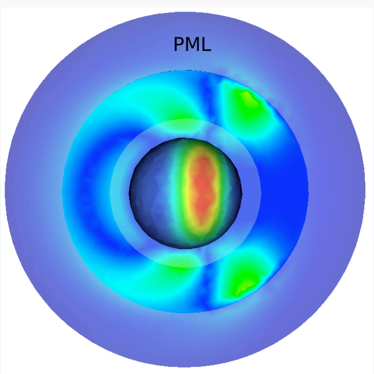


[1]: G. Cohen et.al."Dissipative terms and local time-stepping improvements in a spatial high order Discontinuous Galerkin scheme for the time-domain Maxwell's equations". J. Comput. Phys.,Vol. 227, 2008.

DGTD+PML







- The derived DGTD method is explicit and conditionally stable.
- A matrix-free memory efficient implementation is applied. There is no assembly and storage of global matrices. All updates are performed at element level.
- Interior Penalty (IP) derivation provides multiple formulations
- A Conformal PML is applied to reduce the buffer space
- A Local Time-Stepping strategy is applied to increase computational efficiency especially for multi-scale applications

Conformal Perfectly Matched Layer

- $-j\omega\mu\bar{\bar{s}}\cdot\mathbf{H} = \nabla\times\mathbf{E}$ and $j\omega\epsilon\bar{\bar{s}}\cdot\mathbf{E} = \nabla\times\mathbf{H}$ where $s_{11} = s_2 s_3/s_1$, $s_{22} = s_1 s_3/s_2$ and $s_{33} = s_1 s_2/s_3$
- $s_{1,2} = \kappa_{1,2}(\xi_3, r_{1,2}) + \sigma_{1,2}(\xi_3, r_{1,2})/j\omega\epsilon_o$, $s_3 = \kappa_3(\xi_3) + \sigma_3(\xi_3)/j\omega\epsilon_o$ and $r_{1,2}$ are the local radii of curvature for attenuation in the ξ_3 direction
- An anisotropic PML in time domain is enforced with the introduction of the auxiliary vector fields $\mathbf{P}_e$ and $\mathbf{P}_h$

Equations in PML [1]

- $-\nabla\times\mathbf{E}_i - \mu\bar{\bar{b}}\mathbf{H}_i - \mu\bar{\bar{c}}\mathbf{P_h}_i = \frac{\partial\mu\bar{\bar{a}}\mathbf{H}_i}{\partial t}$
- $\nabla\times\mathbf{H}_i - \epsilon\bar{\bar{b}}\mathbf{E}_i - \epsilon\bar{\bar{c}}\mathbf{P_e}_i = \frac{\partial\epsilon\bar{\bar{a}}\mathbf{E}_i}{\partial t}$
- $\bar{\bar{\kappa}}^{-1}\mathbf{H} - \bar{\bar{d}}\mathbf{P}_h = \frac{\partial\mathbf{P}_h}{\partial t}$
- $\bar{\bar{\kappa}}^{-1}\mathbf{E} - \bar{\bar{d}}\mathbf{P}_e = \frac{\partial\mathbf{P}_e}{\partial t}$

Space and Time discetization

- $\mathbf{M}_a\frac{\partial\mathbf{e}_i}{\partial t} = -\mathbf{M}_b\mathbf{e}_i - \mathbf{M}_c\mathbf{e}_i + (\mathbf{S}_e - \mathbf{F}_e^{ii})\mathbf{h}_i - \mathbf{F}_e^{ij}\mathbf{h}_j\,|_{t=n+1}$
- $\mathbf{M}_a\frac{\partial\mathbf{h}_i}{\partial t} = -\mathbf{M}_b\mathbf{h}_i - \mathbf{M}_c\mathbf{h}_i - (\mathbf{S}_h + \mathbf{F}_h^{ii})\mathbf{e}_i + \mathbf{F}_h^{ij}\mathbf{e}_j\,|_{t=n+1/2}$
- $\mathbf{M}\frac{\partial\mathbf{p}_i^h}{\partial t} = \mathbf{M}_{k-1}\mathbf{h}_i - \mathbf{M}_d\mathbf{p}_i^h\,|_{t=n+1}$
- $\mathbf{M}\frac{\partial\mathbf{p}_i^e}{\partial t} = \mathbf{M}_{k-1}\mathbf{e}_i - \mathbf{M}_d\mathbf{p}_i^e\,|_{t=n+1/2}$

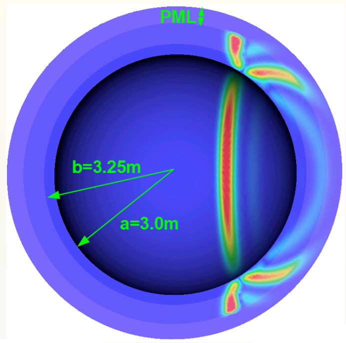
Conformal PML [2]

- $a_{11} = \frac{\kappa_2\kappa_3}{\kappa_1}$
- $b_{11} = \frac{1}{\kappa_1\epsilon_o}(\sigma_2\kappa_3 + \sigma_3\kappa_2 - a_{11}\kappa_3)$
- $c_{11} = \frac{\sigma_2\sigma_3}{\epsilon_o^2} - b_{11}\frac{\sigma_1}{\epsilon_o}$, $d_{11} = \frac{\sigma_1}{\kappa_1\epsilon_o}$
- $\mathbf{J} = \begin{pmatrix} u_{1x} & u_{1y} & u_{1z} \\ u_{2x} & u_{2y} & u_{2z} \\ u_{3x} & u_{3y} & u_{3z} \end{pmatrix}$
- $\bar{\bar{\Lambda}}_{xyz} = \mathbf{J}^t\bar{\bar{\Lambda}}_{u_1u_2u_3}\mathbf{J}$

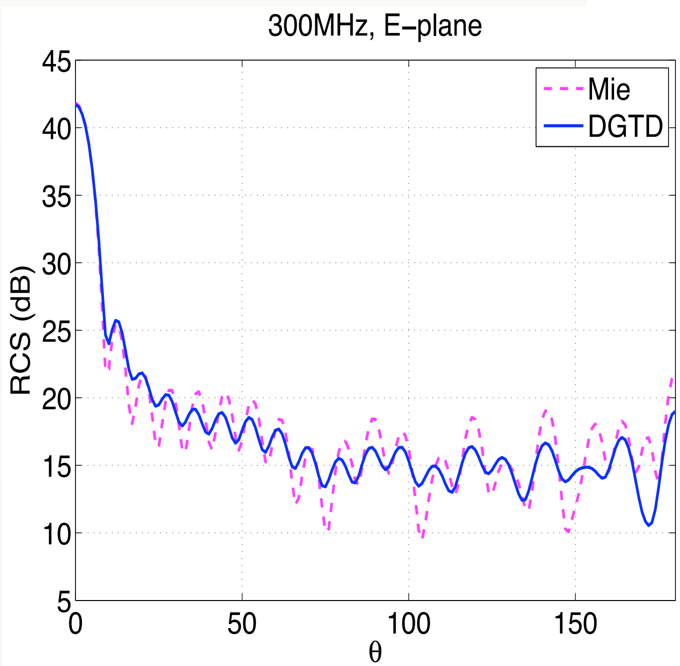
[1]: S. Gedney et.al."A Discontinuous Galerkin Finite Element Time Domain Method with PML"
[2]: F. Teixeira et.al."Analytical Derivation of a Conformal Perfectly Matched Absorber for Electromagnetic Waves"

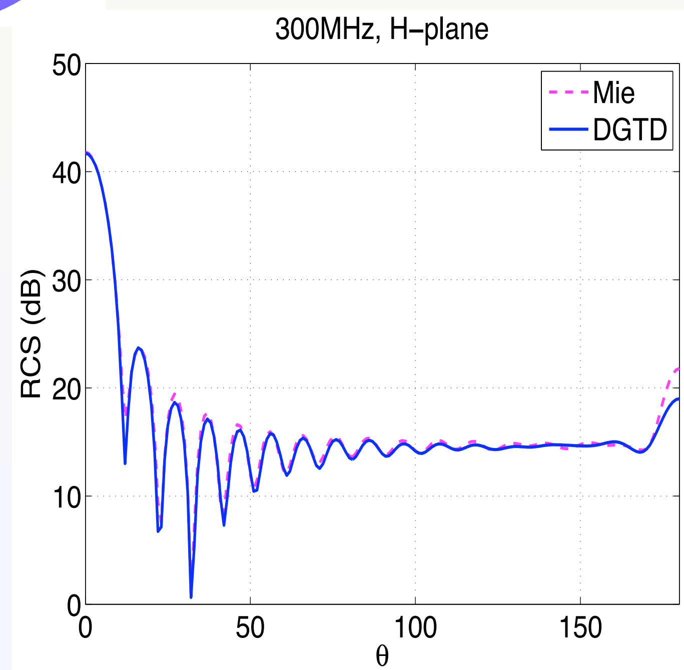
Coated Sphere Scattering

- Inner radius is $a = 3.0m$, outer radius is $b = 3.25m$ and the coating has $\epsilon_r = 2.0$
- $\mathbf{E}^{inc} = \mathbf{E}_0 e^{-[t-t_0-\hat{\mathbf{k}}\cdot(\mathbf{r}-\mathbf{r}_0)/c]^2/\tau^2}$,$\mathbf{E}_0 = (0,0,1)$, $\hat{\mathbf{k}} = (1,0,0)$



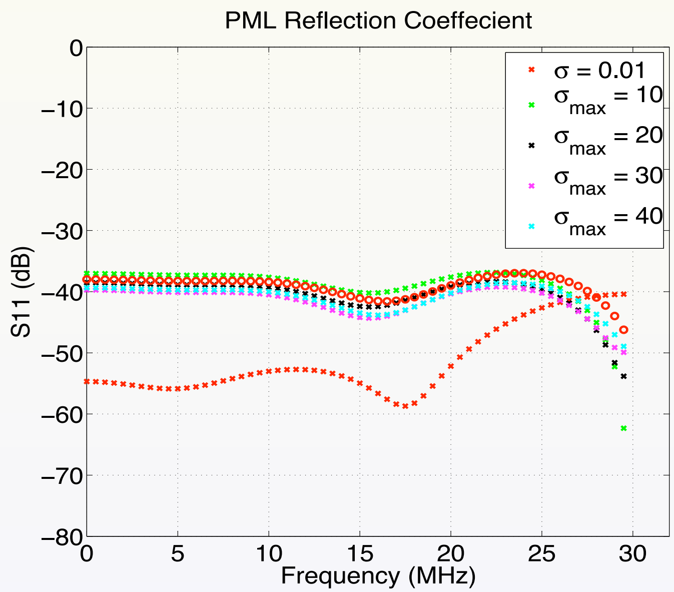
- $RCS_{error} = 9.03e-02$

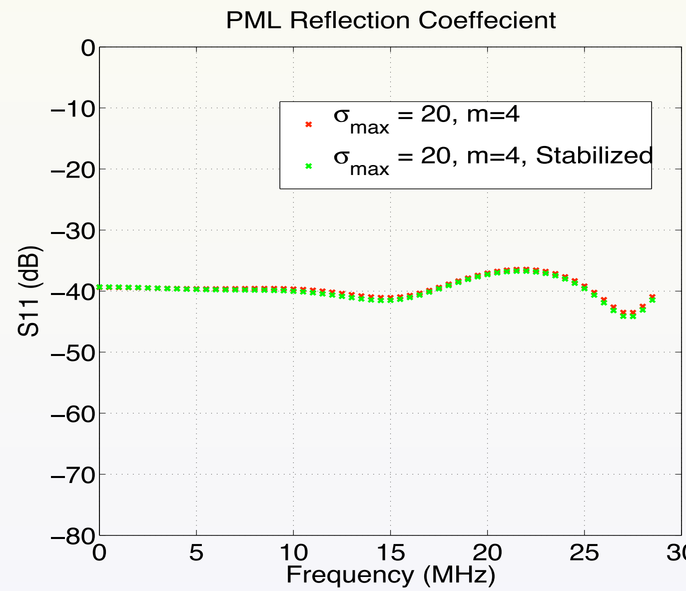




PML performance and Late Time Instability

- $\sigma(\xi) = \sigma_{max} \frac{\xi^m}{\delta^m}$ for attenuation in the ξ direction
- $m = 0$(constant σ) is stable and provides smallest reflection [1]
- σ profiling leads to late time instability [2](linear growth)
- Stabilization [2] removes instability without significantly altering the PML properties



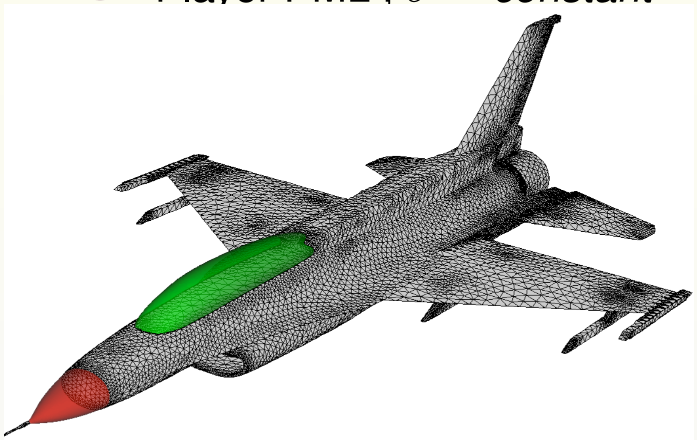


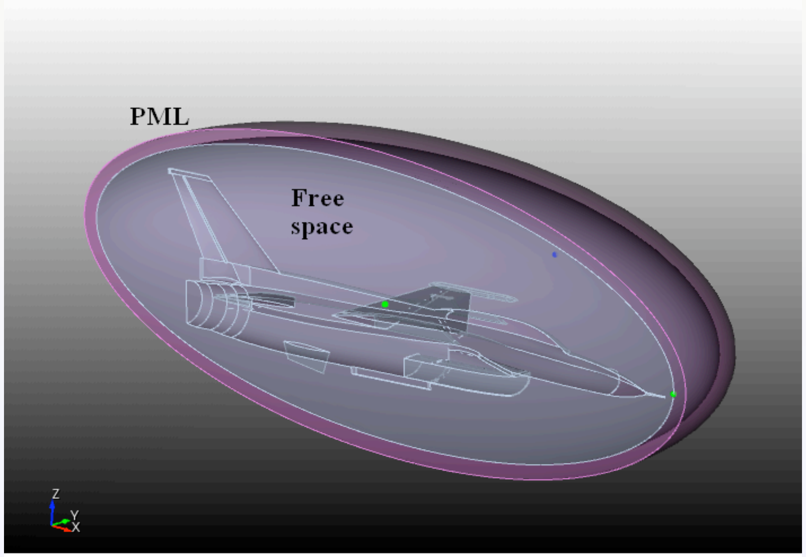
[1]: J. Niegemann et.al."Higher-order time-domain methods for the analysis of nano-photonic systems"

[2]: J. Heasthaven et.al."Long Time Behavior of the Perfectly Matched Layer Equations in Computational Electromagnetics"

F-16 Scattering with Conformal PML

- F-16 featuring fine detail($\delta t_{min} = 7.50e-13$, $\delta t_{max} = 5.33e-11$)
- Dielectric radome and glass canopy along with PEC body
- 4-layer PML , $\sigma = constant = 0.02$



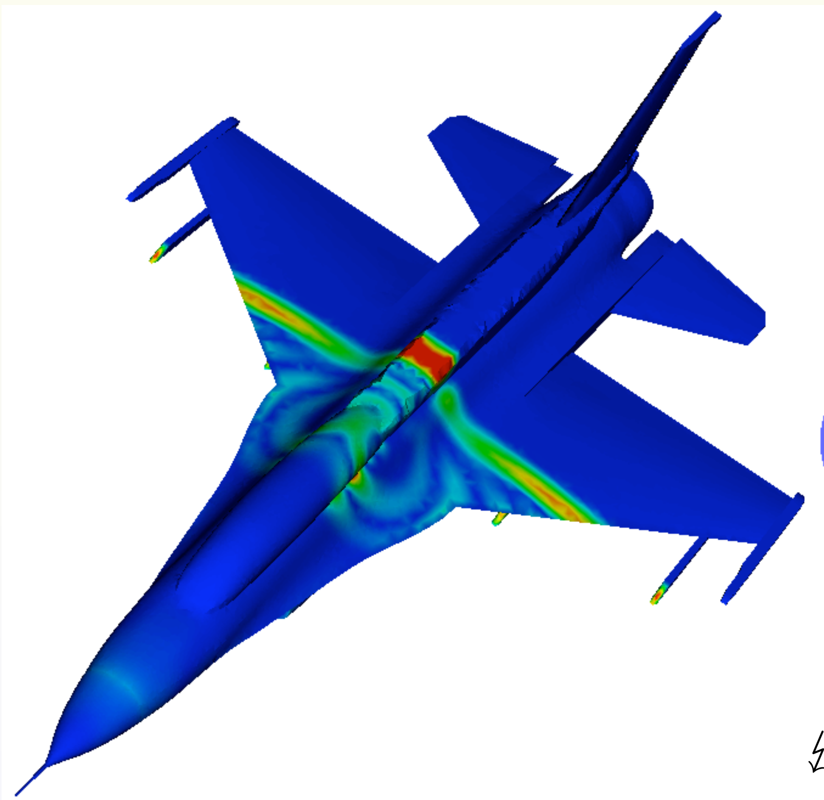


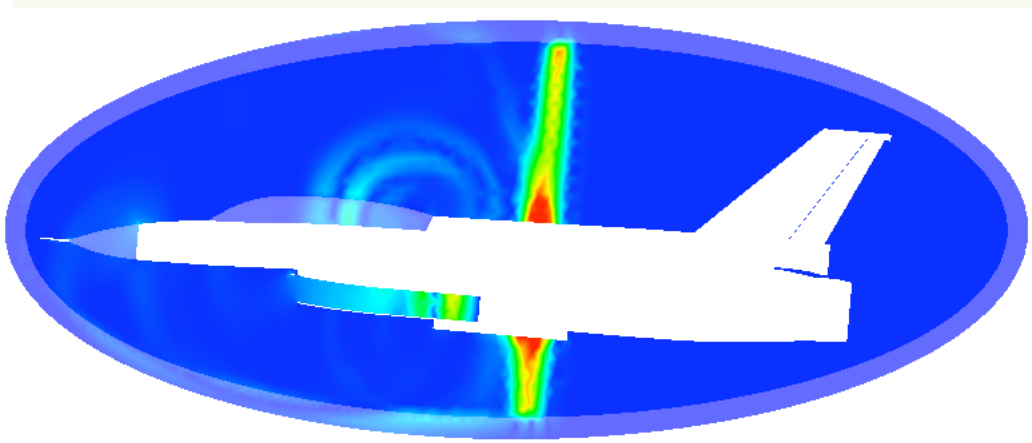
Computational Statistics

Tetrahedra	1,656,676
DOFs	52,641,744
DOFs in PML	13,213,392
Number of Classes	4
Elements per Class	0: 252, 1: 36158, 2: 395760, 3: 1224506
Solution time LTS	88.38 hours
Solution time no LTS	1,301 hours
CPU Gain with LTS	14.7245
Memory Matrix free	18.3 GB
Memory no Matrix free	$>$ 32 GB

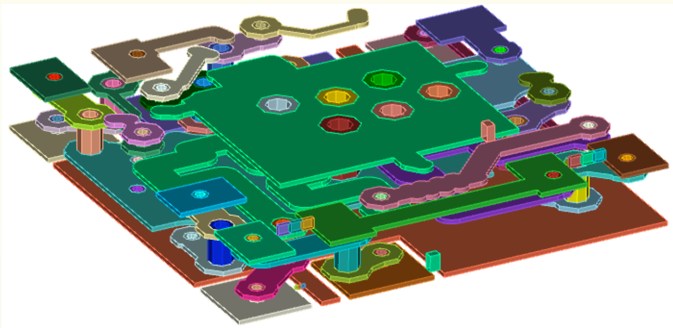
F-16 Scattering with Conformal PML

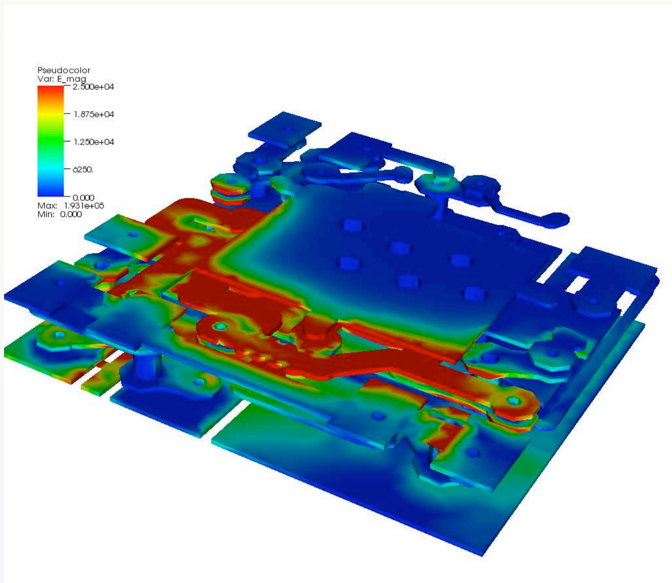
- F-16 aircraft with incident Gaussian pulse($f_{3dB} = 300MHz$)





DGTD + Lumped Elements

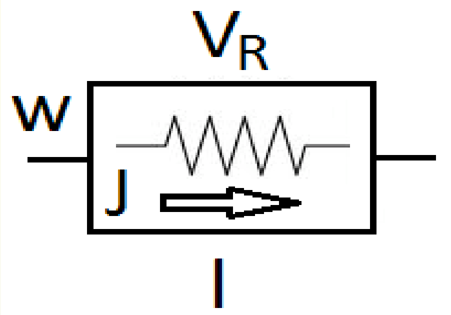


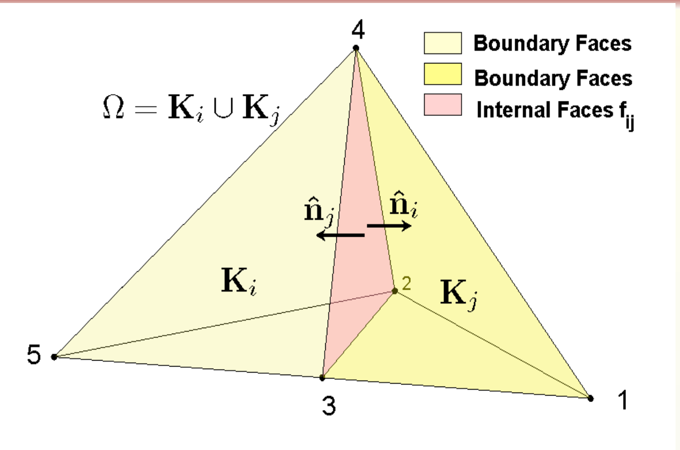


- Lumped elements are small compared to the wavelength. We can assume that the electric and magnetic fields are constant in the surface of the lumped element
- Start from the voltage and current relationships
- Derive the equivalent relationship that describe each of the R, L, C in terms of the electric and magnetic fields.
- Enforce these field expressions weakly through the IPDG formulation

Resistor $V_R = I_R R$

- Surface Impedance





Element K_i, $V_{Ri} = I_R R$ and $V_{Ri} = V_{Rj}$

$$\hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j = \frac{l}{Rw}\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \mathbf{E}_i \ \text{ on } \Gamma_R$$

$$\hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j = 0 \ \text{ on } \Gamma_R$$

Element K_j, $V_{Rj} = I_R R$ and $V_{Rj} = V_{Ri}$

$$\hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j = \frac{l}{Rw}\hat{\mathbf{n}}_j \times \hat{\mathbf{n}}_j \times \mathbf{E}_j \ \text{ on } \Gamma_R$$

$$\hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j = 0 \ \text{ on } \Gamma_R$$

- $V_h^k = \{\mathbf{v} \in [\mathbf{L}^2(\Omega)]^3 : \mathbf{v}|_K \in [\mathbf{P}^k(K)]^3, \quad \forall K \in \mathcal{T}_h\}$
- $\mathbf{E}_h(\mathbf{r},t)|_{K_i} \approx \sum_k^{N_e} e_k(t)\mathbf{v}_{ik}(\mathbf{r})$ and $\mathbf{H}_h(\mathbf{r},t)|_{K_i} \approx \sum_k^{N_h} h_k(t)\mathbf{w}_{ik}(\mathbf{r})$, $\mathbf{w},\mathbf{v} \in V_h^k$

Residuals in Element K_i

$$\mathbf{R}_{\Gamma_R}^{(1)} = \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{l}{Rw}\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \mathbf{E}_i \quad (\mathbf{J}^{err})$$

$$\mathbf{R}_{\Gamma_R}^{(2)} = \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j \quad (\mathbf{M}^{err})$$

Residuals in Element K_j

$$\mathbf{R}_{\Gamma_R}^{(3)} = \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{l}{Rw}\hat{\mathbf{n}}_j \times \hat{\mathbf{n}}_j \times \mathbf{E}_j \quad (\mathbf{J}^{err})$$

$$\mathbf{R}_{\Gamma_R}^{(2)} = \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j \quad (\mathbf{M}^{err})$$

Resistor $V_R = I_R R$ (Cont'd)

Testing Element K_i

$$\left\langle \pi_\tau(\mathbf{v}_i), \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{l}{Rw}\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \mathbf{E}_i \right\rangle = 0 \quad (\mathbf{E}\cdot\mathbf{J}^{err})$$

$$\left\langle \pi_\tau(\mathbf{w}_i), \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j \right\rangle = 0 \quad (\mathbf{H}\cdot\mathbf{M}^{err})$$

Testing Element K_j

$$\left\langle \pi_\tau(\mathbf{v}_j), \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{l}{Rw}\hat{\mathbf{n}}_j \times \hat{\mathbf{n}}_j \times \mathbf{E}_j \right\rangle = 0 \quad (\mathbf{E}\cdot\mathbf{J}^{err})$$

$$\left\langle \pi_\tau(\mathbf{w}_j), \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j \right\rangle = 0 \quad (\mathbf{H}\cdot\mathbf{M}^{err})$$

Time-Discretization

$$\mathbf{M}_\epsilon \frac{\partial \mathbf{e}_i}{\partial t} = \mathbf{S}_e\mathbf{h}_i - \mathbf{F}_e^{ii}\mathbf{h}_i - \frac{l}{Rw}\mathbf{B}_e^R\mathbf{e}_i - \mathbf{F}_e^{ij}\mathbf{h}_j \;\Big|_{t=n+\frac{1}{2}}$$

$$\mathbf{M}_\mu \frac{\partial \mathbf{h}_i}{\partial t} = -\mathbf{S}_h\mathbf{e}_i + \mathbf{F}_h^{ii}\mathbf{e}_i + \mathbf{F}_h^{ij}\mathbf{e}_j \;\Big|_{t=n+1}$$

- $\mathbf{H}_{i(j)}^{n+1/2}$ and $\mathbf{E}_{i(j)}^{n+1}$ are available
- $\mathbf{E}_i^{n+1/2} \approx \frac{\mathbf{E}_i^n + \mathbf{E}_i^{n+1}}{2}$

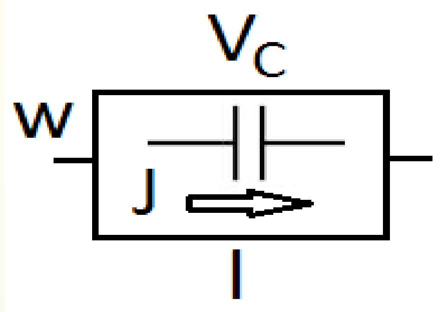
Fully Discretized System

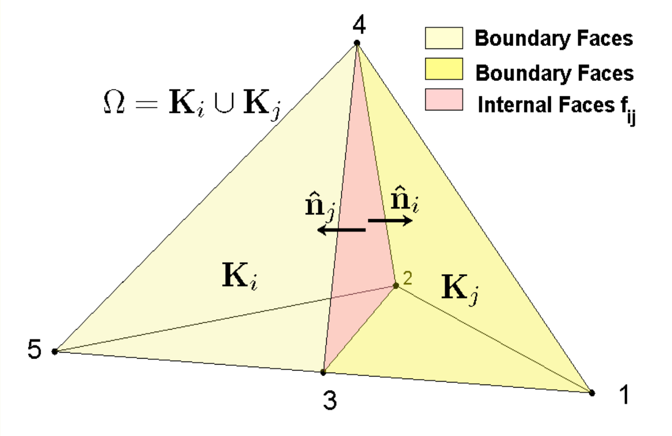
$$(\mathbf{M}_\epsilon + \frac{\delta t l}{2Rw}\mathbf{B}_e^R)\mathbf{e}_i^{n+1} = (\mathbf{M}_\epsilon - \frac{\delta t l}{2Rw}\mathbf{B}_e^R)\mathbf{e}_i^n + \delta t(\mathbf{S}_e - \mathbf{F}_e^{ii})\mathbf{h}_i^{n+\frac{1}{2}} - \delta t\mathbf{F}_e^{ij}\mathbf{h}_j^{n+\frac{1}{2}}$$

$$\mathbf{M}_\mu\mathbf{h}_i^{n+\frac{3}{2}} = \mathbf{M}_\mu\mathbf{h}_i^{n+\frac{1}{2}} + \delta t(-\mathbf{S}_h + \mathbf{F}_h^{ii})\mathbf{e}_i^{n+1} + \delta t\mathbf{F}_h^{ij}\mathbf{e}_j^{n+1}$$

Capacitor $I_C = C\frac{dV_C}{dt}$

- Surface Impedance





Element K_i, $I_C = C\frac{dV_{Ci}}{dt}$ and $V_{Ci} = V_{Cj}$

$$\hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j = \frac{Cl}{w}\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \frac{d\mathbf{E}_i}{dt} \text{ on } \Gamma_C$$

$$\hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j = 0 \text{ on } \Gamma_C$$

Element K_j, $I_C = C\frac{dV_{Cj}}{dt}$ and $V_{Ci} = V_{Cj}$

$$\hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j = \frac{Cl}{w}\hat{\mathbf{n}}_j \times \hat{\mathbf{n}}_j \times \frac{d\mathbf{E}_j}{dt} \text{ on } \Gamma_C$$

$$\hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j = 0 \text{ on } \Gamma_C$$

- $V_h^k = \{\mathbf{v} \in [\mathbf{L}^2(\Omega)]^3 : \mathbf{v}|_K \in [\mathbf{P}^k(K)]^3, \quad \forall K \in \mathcal{T}_h\}$
- $\mathbf{E}_h(\mathbf{r},t)|_{K_i} \approx \sum_k^{N_e} e_k(t)\mathbf{v}_{ik}(\mathbf{r})$ and $\mathbf{H}_h(\mathbf{r},t)|_{K_i} \approx \sum_k^{N_h} h_k(t)\mathbf{w}_{ik}(\mathbf{r})$, $\mathbf{w}, \mathbf{v} \in V_h^k$

Residuals in Element K_i

$$\mathbf{R}_{\Gamma_C}^{(1)} = \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{Cl}{w}\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \frac{d\mathbf{E}_i}{dt} \quad (\mathbf{J}^{err})$$

$$\mathbf{R}_{\Gamma_C}^{(2)} = \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j \quad (\mathbf{M}^{err})$$

Residuals in Element K_j

$$\mathbf{R}_{\Gamma_C}^{(3)} = \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{Cl}{w}\hat{\mathbf{n}}_j \times \hat{\mathbf{n}}_j \times \frac{d\mathbf{E}_j}{dt} \quad (\mathbf{J}^{err})$$

$$\mathbf{R}_{\Gamma_C}^{(2)} = \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j \quad (\mathbf{M}^{err})$$

Capacitor $I_C = C\frac{dV_C}{dt}$ (Cont'd)

Testing Element K_i

$$\langle \pi_\tau(\mathbf{v}_i), \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{Cl}{w}\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \frac{d\mathbf{E}_i}{dt}\rangle = 0 \quad (\mathbf{E}\cdot\mathbf{J}^{err})$$

$$\langle \pi_\tau(\mathbf{w}_i), \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j\rangle = 0 \quad (\mathbf{H}\cdot\mathbf{M}^{err})$$

Testing Element K_j

$$\langle \pi_\tau(\mathbf{v}_j), \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{Cl}{w}\hat{\mathbf{n}}_j \times \hat{\mathbf{n}}_j \times \frac{d\mathbf{E}_j}{dt}\rangle = 0 \quad (\mathbf{E}\cdot\mathbf{J}^{err})$$

$$\langle \pi_\tau(\mathbf{w}_j), \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j\rangle = 0 \quad (\mathbf{H}\cdot\mathbf{M}^{err})$$

Time-Discretization

$$\mathbf{M}_\epsilon \frac{\partial \mathbf{e}_i}{\partial t} = \mathbf{S}_e\mathbf{h}_i - \mathbf{F}_e^{ii}\mathbf{h}_i - \frac{Cl}{w}\mathbf{B}_e^R\frac{\partial \mathbf{e}_i}{\partial t} - \mathbf{F}_e^{ij}\mathbf{h}_j \ |_{t=n+\frac{1}{2}}$$

$$\mathbf{M}_\mu \frac{\partial \mathbf{h}_i}{\partial t} = -\mathbf{S}_h\mathbf{e}_i + \mathbf{F}_h^{ii}\mathbf{e}_i + \mathbf{F}_h^{ij}\mathbf{e}_j \ |_{t=n+1}$$

- $\mathbf{H}_{i(j)}^{n+1/2}$ and $\mathbf{E}_{i(j)}^{n+1}$ are available
- $\frac{\partial \mathbf{E}_i}{\partial t}|_{n+\frac{1}{2}} = \frac{\mathbf{E}_i^{n+1} - \mathbf{E}_i^n}{\delta t}$

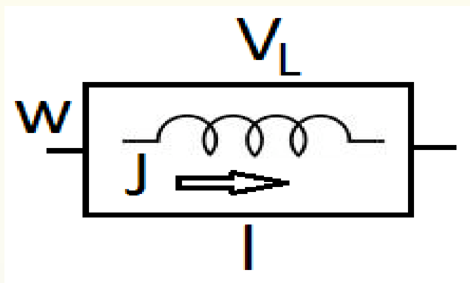
Fully Discretized System

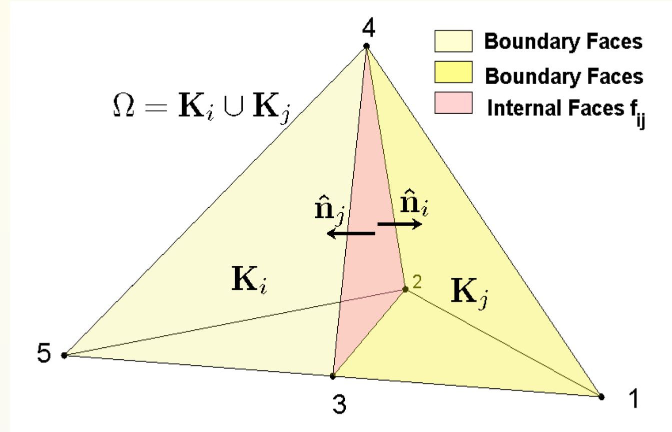
$$(\mathbf{M}_\epsilon + \frac{Cl}{w}\mathbf{B}_e^C)\mathbf{e}_i^{n+1} = (\mathbf{M}_\epsilon + \frac{Cl}{w}\mathbf{B}_e^C)\mathbf{e}_i^n + \delta t(\mathbf{S}_e - \mathbf{F}_e^{ii})\mathbf{h}_i^{n+\frac{1}{2}} - \delta t\mathbf{F}_e^{ij}\mathbf{h}_j^{n+\frac{1}{2}}$$

$$\mathbf{M}_\mu\mathbf{h}_i^{n+\frac{3}{2}} = \mathbf{M}_\mu\mathbf{h}_i^{n+\frac{1}{2}} + \delta t(-\mathbf{S}_h + \mathbf{F}_h^{ii})\mathbf{e}_i^{n+1} + \delta t\mathbf{F}_h^{ij}\mathbf{e}_j^{n+1}$$

Inductor $I_L = \frac{1}{L}\int_0^t V_L dt$

- Surface Impedance





Element K_i, $I_L = \frac{1}{L}\int_0^t V_L dt$ and and $V_{Li} = V_{Lj}$

$$\hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j = \frac{l}{Lw}\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \int_0^t \mathbf{E}_i dt \ \text{ on } \Gamma_L$$

$$\hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j = 0 \ \text{ on } \Gamma_L$$

Element K_j, $I_L = \frac{1}{L}\int_0^t V_L dt$ and and $V_{Li} = V_{Lj}$

$$\hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j = \frac{l}{Lw}\hat{\mathbf{n}}_j \times \hat{\mathbf{n}}_j \times \int_0^t \mathbf{E}_j dt \ \text{ on } \Gamma_L$$

$$\hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j = 0 \ \text{ on } \Gamma_L$$

- $V_h^k = \{\mathbf{v} \in [\mathbf{L}^2(\Omega)]^3 : \mathbf{v}|_K \in [\mathbf{P}^k(K)]^3, \quad \forall K \in \mathcal{T}_h\}$
- $\mathbf{E}_h(\mathbf{r},t)|_{K_i} \approx \sum_k^{N_e} e_k(t)\mathbf{v}_{ik}(\mathbf{r})$ and $\mathbf{H}_h(\mathbf{r},t)|_{K_i} \approx \sum_k^{N_h} h_k(t)\mathbf{w}_{ik}(\mathbf{r})$, $\mathbf{w},\mathbf{v} \in V_h^k$

Residuals in Element K_i

$$\mathbf{R}_{\Gamma_L}^{(1)} = \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{l}{Lw}\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \int_0^t \mathbf{E}_i dt \quad (\mathbf{J}^{err})$$

$$\mathbf{R}_{\Gamma_L}^{(2)} = \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j \quad (\mathbf{M}^{err})$$

Residuals in Element K_j

$$\mathbf{R}_{\Gamma_L}^{(3)}\hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{l}{Lw}\hat{\mathbf{n}}_j \times \hat{\mathbf{n}}_j \times \int_0^t \mathbf{E}_j dt \quad (\mathbf{J}^{err})$$

$$\mathbf{R}_{\Gamma_L}^{(2)} = \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j \quad (\mathbf{M}^{err})$$

Inductor $I_L = \frac{1}{L}\int_0^t V_L dt$ (Cont'd)

Testing Element K_i

$$\langle \pi_\tau(\mathbf{v}_i), \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{l}{Lw}\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \int_0^t \mathbf{E}_i dt\rangle = 0 \quad (\mathbf{E}\cdot\mathbf{J}^{err})$$

$$\langle \pi_\tau(\mathbf{w}_i), \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j\rangle = 0 \quad (\mathbf{H}\cdot\mathbf{M}^{err})$$

Testing Element K_j

$$\langle \pi_\tau(\mathbf{v}_j), \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{l}{Lw}\hat{\mathbf{n}}_j \times \hat{\mathbf{n}}_j \times \int_0^t \mathbf{E}_j dt\rangle = 0 \quad (\mathbf{E}\cdot\mathbf{J}^{err})$$

$$\langle \pi_\tau(\mathbf{w}_j), \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j\rangle = 0 \quad (\mathbf{H}\cdot\mathbf{M}^{err})$$

Time-Discretization

$$\mathbf{M}_\epsilon \frac{\partial \mathbf{e}_i}{\partial t} = \mathbf{S}_e \mathbf{h}_i - \mathbf{F}_e^{ii}\mathbf{h}_i - \frac{l}{Lw}\mathbf{B}_e^L \int_0^t \mathbf{e}_i\, dt - \mathbf{F}_e^{ij}\mathbf{h}_j \;|_{t=n+\frac{1}{2}}$$

$$\mathbf{M}_\mu \frac{\partial \mathbf{h}_i}{\partial t} = -\mathbf{S}_h \mathbf{e}_i + \mathbf{F}_h^{ii}\mathbf{e}_i + \mathbf{F}_h^{ij}\mathbf{e}_j \;|_{t=n+1}$$

- $\mathbf{H}_{i(j)}^{n+1/2}$ and $\mathbf{E}_{i(j)}^{n+1}$ are available
- $\int_0^{n+1/2} \mathbf{E}_i dt \approx \delta t \sum_{k=0}^n \frac{\mathbf{E}_i^k + \mathbf{E}_i^{k+1}}{2}$
- $\mathbf{\mathfrak{e}}_i^n = \sum_{k=0}^n \mathbf{e}_i^k$

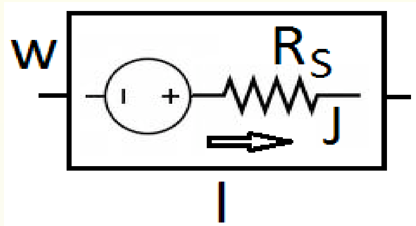
Fully Discretized System

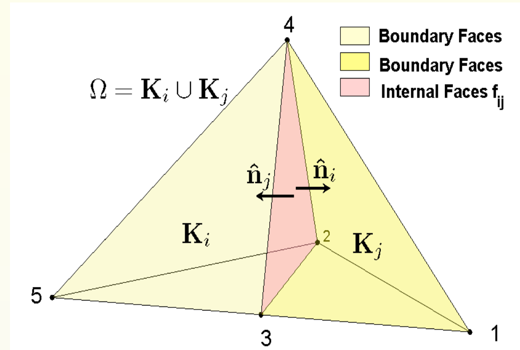
$$(\mathbf{M}_\epsilon + \frac{\delta t^2 l}{2Lw}\mathbf{B}_e^L)\mathbf{e}_i^{n+1} = \mathbf{M}_\epsilon \mathbf{e}_i^n + \frac{\delta t^2 l}{Lw}\mathbf{B}_e^L \mathbf{\mathfrak{e}}_i^n + \delta t(\mathbf{S}_e - \mathbf{F}_e^{ii})\mathbf{h}_i^{n+\frac{1}{2}} - \delta t \mathbf{F}_e^{ij}\mathbf{h}_j^{n+\frac{1}{2}}$$

$$\mathbf{M}_\mu \mathbf{h}_i^{n+\frac{3}{2}} = \mathbf{M}_\mu \mathbf{h}_i^{n+\frac{1}{2}} + \delta t(-\mathbf{S}_h + \mathbf{F}_h^{ii})\mathbf{e}_i^{n+1} + \delta t \mathbf{F}_h^{ij}\mathbf{e}_j^{n+1}$$

Interior Port

- Interior Port





Element K_i, $I^{tot} = \frac{1}{R_S}(V_i^{tot} - V^s)$

$$\hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j = \frac{I}{R_S w} \hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \mathbf{E}_i - \frac{I}{R_S w} \hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \mathbf{E}_i^{inc} \quad \text{on } \Gamma$$

$$\hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j = 0 \quad \text{on } \Gamma$$

Element K_j, $I^{tot} = \frac{1}{R_S}(V_j^{tot} - V^s)$

$$\hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j = \frac{I}{R_S w} \hat{\mathbf{n}}_j \times \hat{\mathbf{n}}_j \times \mathbf{E}_j - \frac{I}{R_S w} \hat{\mathbf{n}}_j \times \hat{\mathbf{n}}_j \times \mathbf{E}_j^{inc} \quad \text{on } \Gamma$$

$$\hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j = 0 \quad \text{on } \Gamma$$

Residuals in $K_{i(j)}$

$$\mathbf{R}_\Gamma^{(1)} = \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{I}{R_S w} \hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \mathbf{E}_i + \frac{I}{R_S w} \hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \mathbf{E}_i^{inc} \quad (\mathbf{J}^{err})$$

$$\mathbf{R}_\Gamma^{(2)} = \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j \quad (\mathbf{M}^{err})$$

$$\mathbf{R}_\Gamma^{(3)} = \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{I}{R_S w} \hat{\mathbf{n}}_j \times \hat{\mathbf{n}}_j \times \mathbf{E}_j + \frac{I}{R_S w} \hat{\mathbf{n}}_j \times \hat{\mathbf{n}}_j \times \mathbf{E}_j^{inc} \quad (\mathbf{J}^{err})$$

Interior Port (cont'd)

Testing Element K_i

$$\langle \pi_\tau(\mathbf{v}_i), \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{l}{R_s w}\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \mathbf{E}_i + \frac{l}{R_s w}\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \mathbf{E}_i^{inc} - \hat{\mathbf{n}}_i \times \mathbf{H}_i^{inc}\rangle = 0 \quad (\mathbf{E}\cdot\mathbf{J}^{err})$$

$$\langle \pi_\tau\left(\mathbf{w}_i\right), \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j\rangle = 0 \quad (\mathbf{H}\cdot\mathbf{M}^{err})$$

Testing Element K_j

$$\langle \pi_\tau\left(\mathbf{v}_j\right), \hat{\mathbf{n}}_i \times \mathbf{H}_i + \hat{\mathbf{n}}_j \times \mathbf{H}_j - \frac{l}{R_s w}\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_j \times \mathbf{E}_j + \frac{l}{R_s w}\hat{\mathbf{n}}_j \times \hat{\mathbf{n}}_j \times \mathbf{E}_j^{inc}\rangle = 0 \quad (\mathbf{E}\cdot\mathbf{J}^{err})$$

$$\langle \pi_\tau\left(\mathbf{w}_j\right), \hat{\mathbf{n}}_i \times \mathbf{E}_i + \hat{\mathbf{n}}_j \times \mathbf{E}_j\rangle = 0 \quad (\mathbf{H}\cdot\mathbf{M}^{err})$$

Time-Discretization

$$\mathbf{M}_\epsilon \frac{\partial \mathbf{e}_i}{\partial t} = \mathbf{S}_e\mathbf{h}_i - \mathbf{F}_e^{ii}\mathbf{h}_i - \frac{l}{2R_s w}\mathbf{B}_e^{R_s}\mathbf{e}_i - \mathbf{F}_e^{ij}\mathbf{h}_j + \mathbf{e}^{inc} \;|_{t=n+\frac{1}{2}}$$

$$\mathbf{M}_\mu \frac{\partial \mathbf{h}_i}{\partial t} = -\mathbf{S}_h\mathbf{e}_i + \mathbf{F}_h^{ii}\mathbf{e}_i + \mathbf{F}_h^{ij}\mathbf{e}_j \;|_{t=n+1}$$

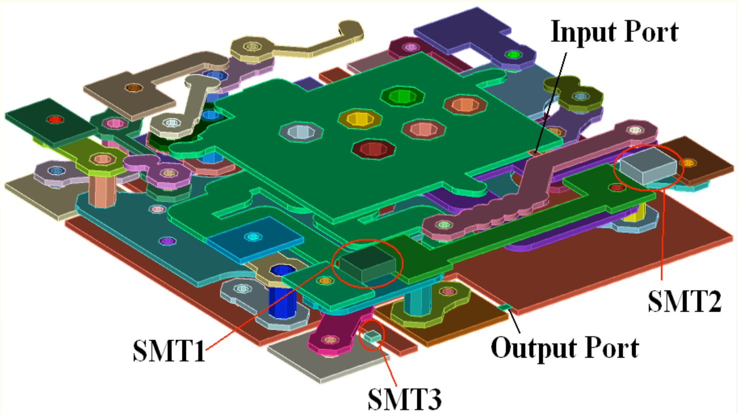
Fully Discretized System

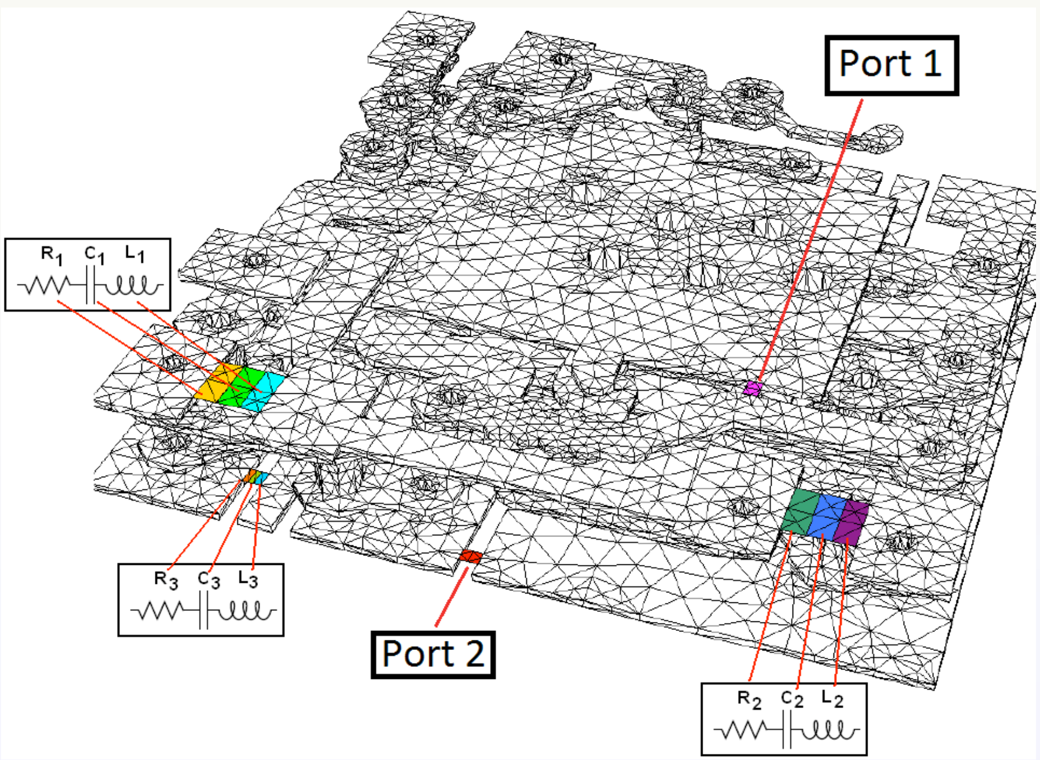
$$(\mathbf{M}_\epsilon + \frac{\delta t l}{2R_s w}\mathbf{B}_e^{R_s})\mathbf{e}_i^{n+1} = (\mathbf{M}_\epsilon - \frac{\delta t l}{2R_s w}\mathbf{B}_e^{R_s})\mathbf{e}_i^{n} + \delta t(\mathbf{S}_e - \mathbf{F}_e^{ii})\mathbf{h}_i^{n+\frac{1}{2}} - \delta t\mathbf{F}_e^{ij}\mathbf{h}_j^{n+\frac{1}{2}} + \delta t\mathbf{e}^{inc}$$

$$\mathbf{M}_\mu\mathbf{h}_i^{n+\frac{3}{2}} = \mathbf{M}_\mu\mathbf{h}_i^{n+\frac{1}{2}} + \delta t(-\mathbf{S}_h + \mathbf{F}_h^{ii})\mathbf{e}_i^{n+1} + \delta t\mathbf{F}_h^{ij}\mathbf{e}_j^{n+1}$$

Interconnect Device with Lumped Elements

- 3mm$\times$3mm$\times$ 0.6mm ($\delta t_{min} = 8.6297 \times 10^{-16} s$, $\delta t_{max} = 1.5804 \times 10^{-13} s$)
- 4-metal layer interconnect device inside a dielectric $\epsilon_r = 3.8$

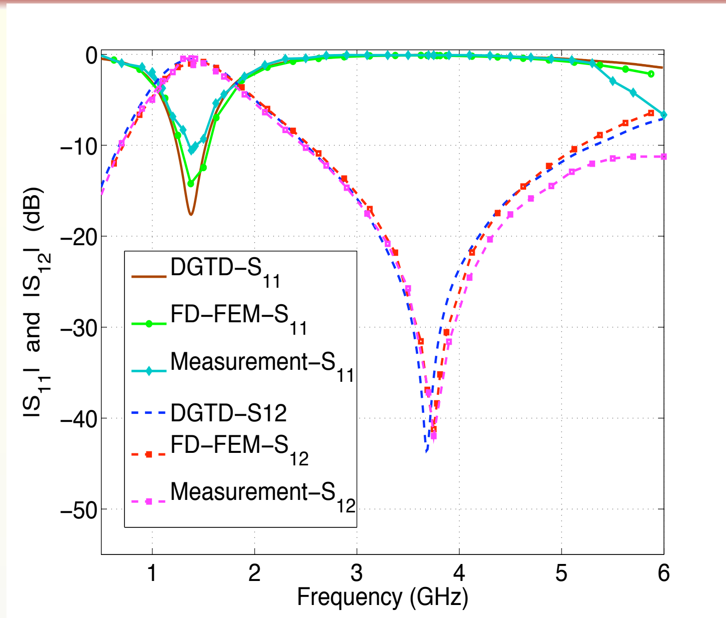


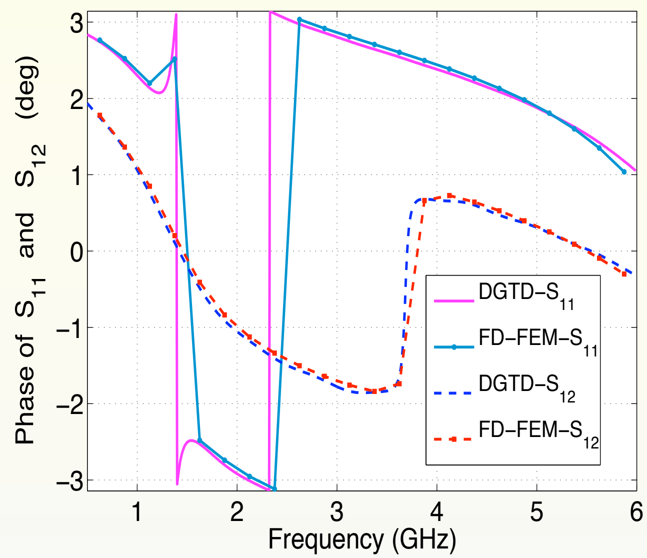


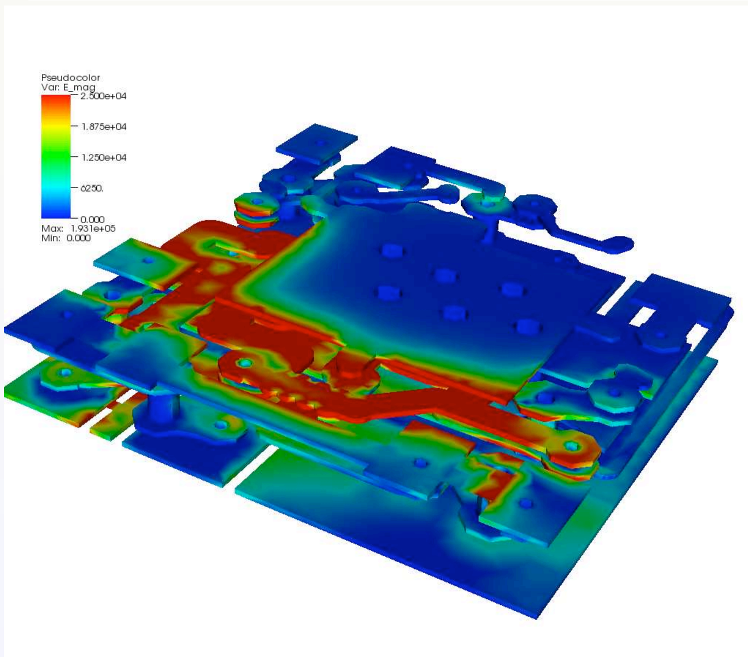
Computational Statistics

Tetrahedra	65,676
DOFs	1,290,608
Number of Classes	5
Elements per Class	0: 49, 1: 4687, 2: 38938, 3: 21515, 4: 649,
Solution time LTS	50.15 hrs
Solution time no LTS	466.72 hrs
CPU Gain with LTS	8.78
Memory	583 MB

S-Parameters







S-parameters

$$S_{11} = \frac{2V_1^{tot}(\omega) - V_s(\omega)}{V_s(\omega)}$$

$$S_{21} = \frac{2V_2^{tot}(\omega)}{V_s(\omega)}$$

Conclusions

- DGTD results in an explicit time-marching algorithm
- Conformal PML and Lumped Elements can be incorporated with DGTD methods
- Upwind flux has optimal convergence whereas Central flux has suboptimal
- Local Time Stepping is a useful strategy for multi-scale applications

Upcoming Work

- MPI/GPU Implemantation
- DGTD on non-conformal meshes

An MPI/GPU Implementation of Interior Penalty Discontinuous Galerkin Time Domain Methods

Stylianos Dosopoulos and Jin-Fa Lee

Features of DG Methods

- General Principles of DG Methods
  - Partition the computational domain into polyhedra
  - In each polyhedron the field is represented as a linear combination of a local set of basis
  - Interelement continuity at polyhedra interfaces is weakly enforced
- DG Pros
  - Explicit time marching schemes in time-domain
  - Non-conformal meshes
  - Easier *hp*− refinement
  - High parallel efficiency. By nature DGTD methods are suitable for parallel hardware (multi-core CPUs, GPUs).
- DG Cons
  - High number of degrees of freedom.

IP-DGTD Formulation

Weak Problem Statement

Find $(\mathbf{H},\mathbf{E}) \in V_h^k \times V_h^k$ *such that*

$$\underbrace{\int_\Omega \mathbf{w}\cdot(\nabla\times\mathbf{E}+\frac{\mu\partial\mathbf{H}}{\partial t})\,d\Omega-\int_\Omega \mathbf{v}\cdot(\nabla\times\mathbf{H}-\frac{\epsilon\partial\mathbf{E}}{\partial t})\,d\Omega}_{\text{VolumeTerms}}+\underbrace{\int_{\mathcal{F}_h}\{\!\{\mathbf{v}\}\!\}\cdot[\![\mathbf{H}]\!]_\tau\,ds}_{SurfaceTerms}$$

$$\underbrace{-\int_{\mathcal{F}_h}\{\!\{\mathbf{w}\}\!\}\cdot[\![\mathbf{E}]\!]_\tau\,ds-e\int_{\mathcal{F}_h}[\![\mathbf{v}]\!]_\tau\cdot[\![\mathbf{E}]\!]_\tau\,ds-f\int_{\mathcal{F}_h}[\![\mathbf{w}]\!]_\tau\cdot[\![\mathbf{H}]\!]_\tau\,ds}_{SurfaceTerms}=0$$

Coefficients

- $V_h^k=\{\mathbf{v}\in[\mathbf{L}^2(\Omega)]^3:\mathbf{v}|_K\in[\mathbf{P}^k(K)]^3,\ \forall K\in\mathcal{T}_h\}$, $\{\!\{\mathbf{u}\}\!\}=(\pi_\tau(\mathbf{u}_i)+\pi_\tau(\mathbf{u}_j))/2$ and $[\![\mathbf{u}]\!]_\tau=\gamma_\tau(\mathbf{u}_i)+\gamma_\tau(\mathbf{u}_j)$
- $e=f=0$ will give rise to a conservative formulation but suboptimal convergence(**central flux**)
- $e=\frac{\gamma}{Z_\Gamma}$ and $f=\frac{\gamma}{Y_\Gamma}$ with $Z_\Gamma=\frac{1}{2}(\sqrt{\frac{\mu_i}{\epsilon_i}}+\sqrt{\frac{\mu_j}{\epsilon_j}})$ and $Y_\Gamma=\frac{1}{2}(\sqrt{\frac{\epsilon_i}{\mu_i}}+\sqrt{\frac{\epsilon_j}{\mu_j}})$ will give rise to a lossy formulation but optimal convergence(**upwind flux**)
- Descretize in time using leap-frog scheme

$$\mathbf{M}_\epsilon\mathbf{e}_i^{n+1}=(\mathbf{M}_\epsilon+e\delta t\mathbf{P}_e)\mathbf{e}_i^n+\delta t(\mathbf{S}_e-\mathbf{F}_e^{ii})\mathbf{h}_i^{n+\frac{1}{2}}-\delta t\mathbf{F}_e^{ij}\mathbf{h}_j^{n+\frac{1}{2}}+e\delta t\mathbf{P}_e^{ij}\mathbf{e}_j^n$$

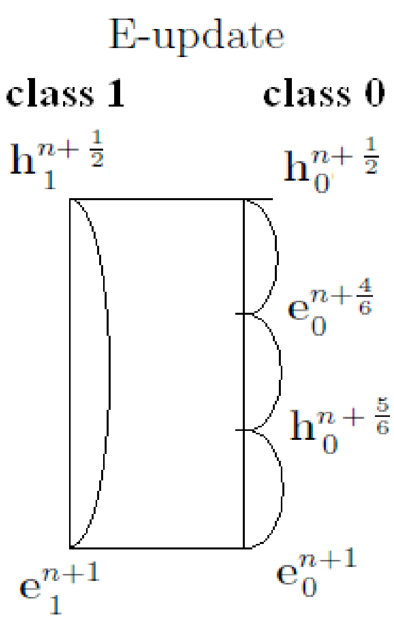
$$\mathbf{M}_\mu\mathbf{h}_i^{n+\frac{3}{2}}=(\mathbf{M}_\mu+f\delta t\mathbf{P}_h)\mathbf{h}_i^{n+\frac{1}{2}}+\delta t(-\mathbf{S}_h+\mathbf{F}_h^{ii})\mathbf{e}_i^{n+1}+\delta t\mathbf{F}_h^{ij}\mathbf{e}_j^{n+1}+f\delta t\mathbf{P}_h^{ij}\mathbf{h}_j^{n+\frac{1}{2}}$$

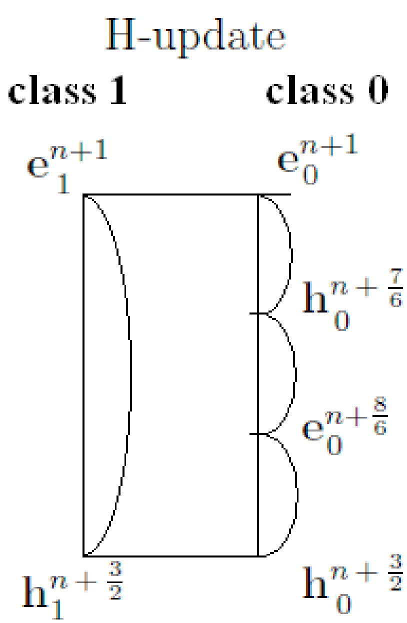
Stability Condition And Local Time-Stepping Update

Stability Condition (S. Piperno)

$$\forall i, \forall k\ , c_i \delta t[2\alpha_i + \beta_{ik} max(\sqrt{\frac{\mu_i}{\mu_k}}, \sqrt{\frac{\epsilon_i}{\epsilon_k}})] < \frac{4V_i}{P_i}$$

- The set of elements is partitioned into N classes. This partition is done before the time-marching simulation and is based on the stability condition
- For the i^{th} class $\delta t_i = (2m+1)^i \delta t_{min}$
- We choose $m = 1$ so each class has three times larger time step from its previous class





[1]: G. Cohen et.al."Dissipative terms and local time-stepping improvements in a spatial high order Discontinuous Galerkin scheme for the time-domain Maxwell's equations". J. Comput. Phys.,Vol. 227, 2008.

Parallel Computing for Scientific Applications

Early era: Moore's law!

- Number of transistors on chip doubles every 2 years
  - More transistors means more data cacheing, better flow control, and higher arithmetic capability [cite]
- Still observed today, but methods for design are very costly (millions today as compared to tens in '70s) [2]
- Push limits of Moore's law by adding more processors on a single chip.
- Now data and task parallelism is possible on single chip by utilizing multiple processors.

Can Moore's Law help physics simulations?

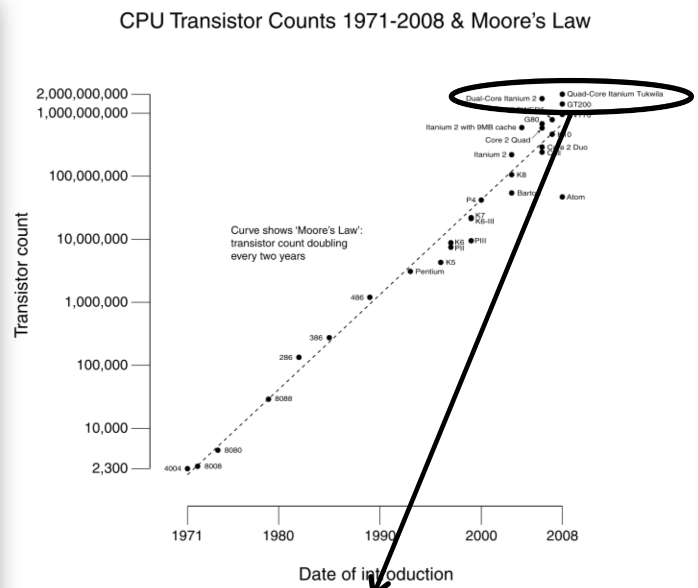
- Realistic simulations require much more memory than is available on single CPU.
- Scientist and engineers pursue implementation of algorithms across multiple connected CPUs
  - Cheap alternative – memory is distributed,

Serial algorithms require re-do!

- Large communication overheads

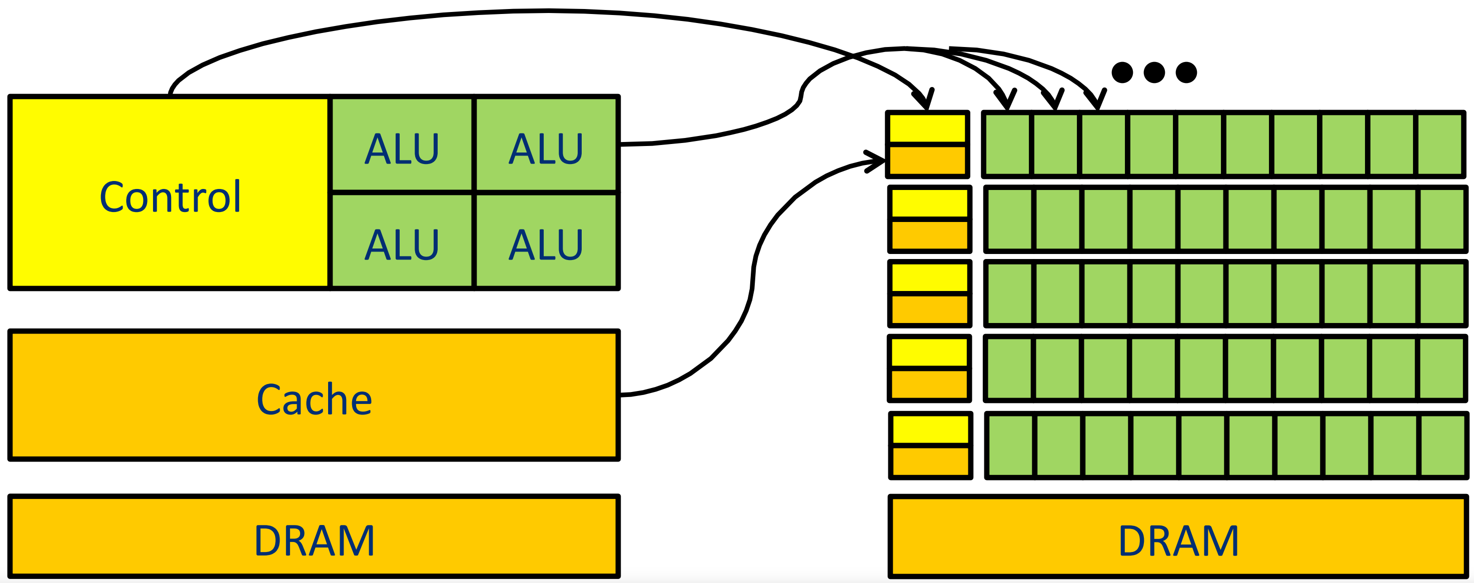
Current State of Parallelization

- Implementation of scientific algorithms on MASSIVELY PARALLEL architectures



Dual Core Itanium 2
Quad Core Itanium Tukwilla

Transistor Utilization of GPGPU vs CPU



CPU Architecture

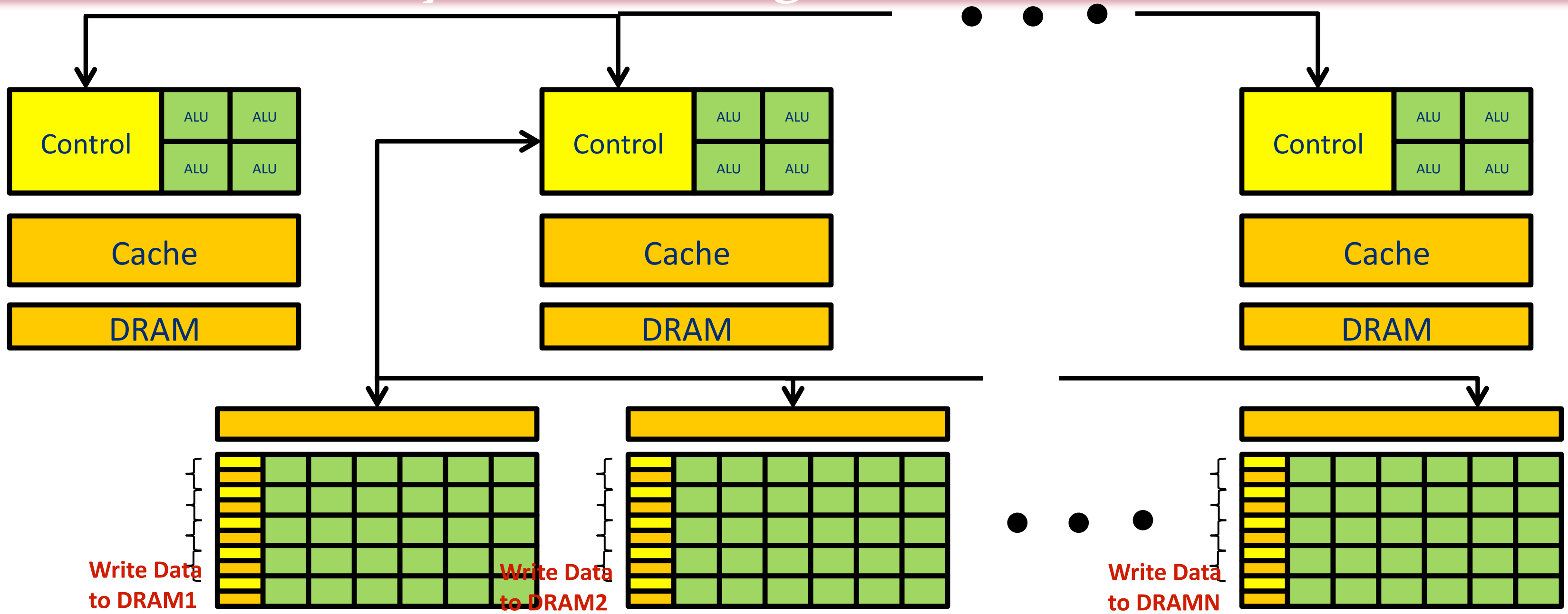
- **Main Transitor Utilizations**
  - Data Caching
  - Flow Control
- Excellent for retreiving and setting data
- Makes CPU great for general-purpose computing

Note: See [4] for further details.

GPU Architecture

- **Main Transitor Utilization**
  - Arithmetic Manipulation
- Excellent for “on-the-fly” computing
- Limited caching and flow control on a processor
- Inter-processor communication is handled through DRAM access.
- DRAM (global memory) access requires hundreds of cycles.

Distributed Memory Parallel Program with GPGPU vs CPU



CPU-CPU Parallelism

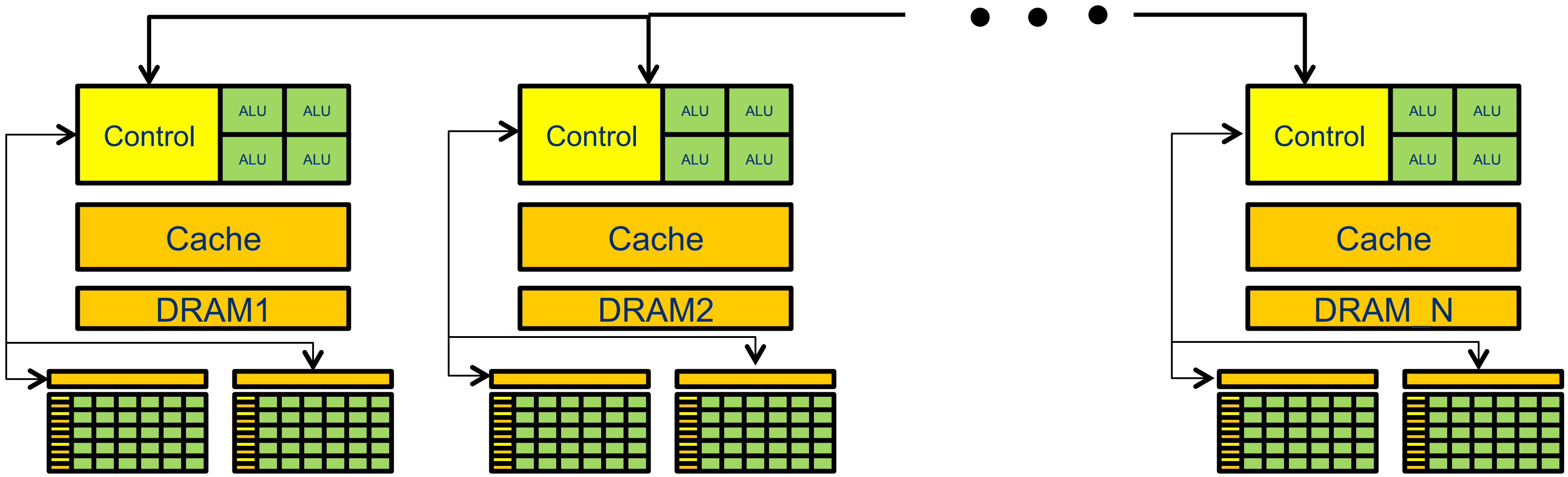
- CPUs are connected via high-speed bus.
- Each node has large DRAM and large cache levels.
- Control units are responsible for symantics of data transfer
- Communication via the high-speed bus is major bottle-neck

GPU-GPU Parallelism

- GPUs are viewed as separate co-processors.
- Each co-processor has its own DRAM.
- LARGE communication bottle-neck between GPU1 and GPU2.
- Individual control units xfer to DRAM (GPU memory), and CPU control accesses GPU

THE OHIO STATE UNIVERSITY

A Heterogeneous Architecture



- **Parallel Computing on Single GPU**
  - Algorithms have been tested on single GPU for scalability (both n-body and EM)
    - FDTD [5-7]
    - FMM [8-9]
    - Direct Methods [12]
    - MLFMM [10-11]
- **Problem**: Acceleration capabilities of GPGPU for frequency-domain CEM has not been explored!
- **GPGPU main purpose: CO-PROCESSOR**
- **True Heterogeneous System**
  - Many connected CPUs
  - Within each CPU – several GPUs
- **Algorithm Challenges:**
  - Minimization of communication
  - Data Locality
    - CPU & GPU
  - Load Balancing
    - CPU & GPU
    - Intra-GPU Balance

Motivation - CPU vs GPU

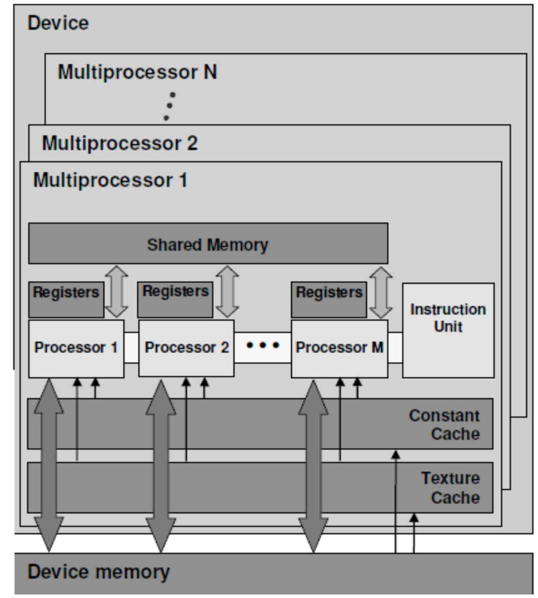
CPU for DGTD

- Modern CPUs with 4 quad-core processors can run up to 16 threads
- Fast global memory access time but memory bandwidth may be a limiting factor in the performance.
- Multi threading will swap the threads execution channels on and off and this is slow and expensive.

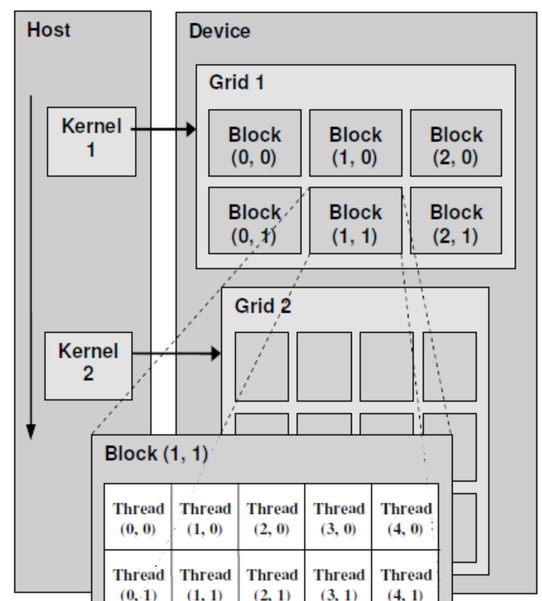
GPU for DGTD

- Modern GPUs have 30 multiprocessors and 1,024 active threads per multiprocessor.
- No swapping occurs between GPU threads.
- Slower global memory access time but memory bandwidth and the FLOPS on a GPU is about one order of magnitude higher than its CPU counterpart
- Nvidia's CUDA model offers an C-like language to program on GPUs
- [1]. http://developer.nvidia.com

GPU Architecture [1]



CUDA Model NVIDIA [1]

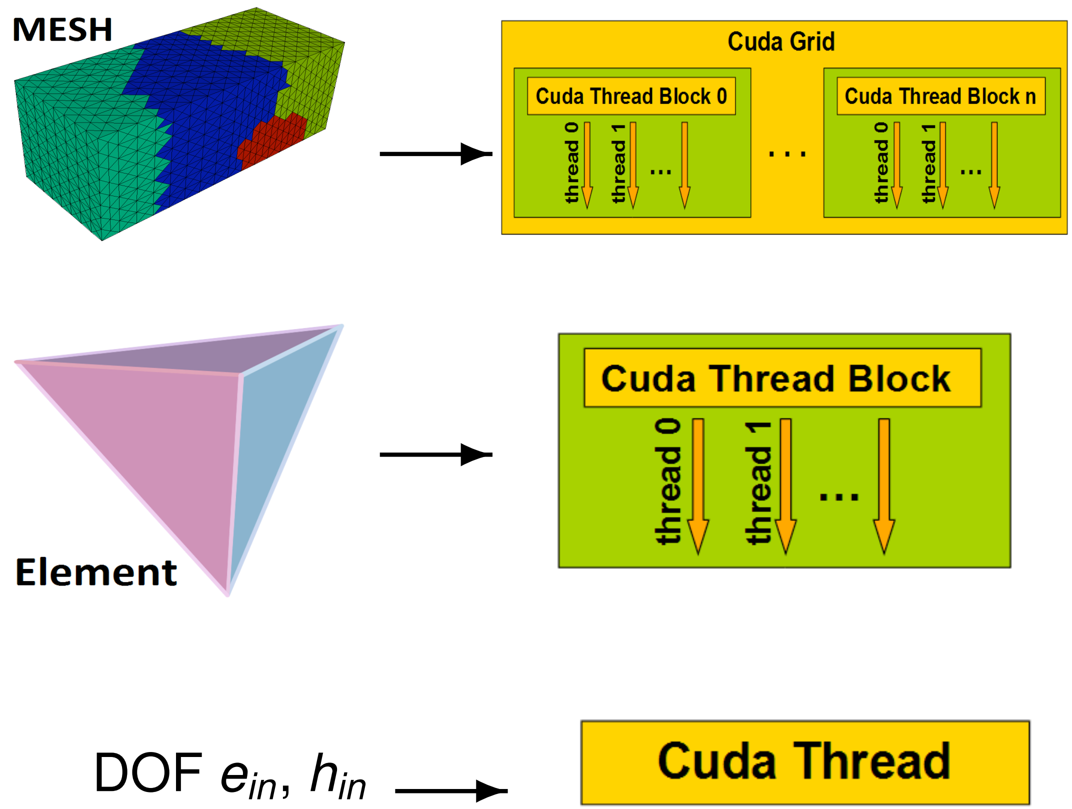


GPU Computational Layout

Establish a computational layout by defining mapping between DGTD and the CUDA architecture

DGTD to CUDA Mapping

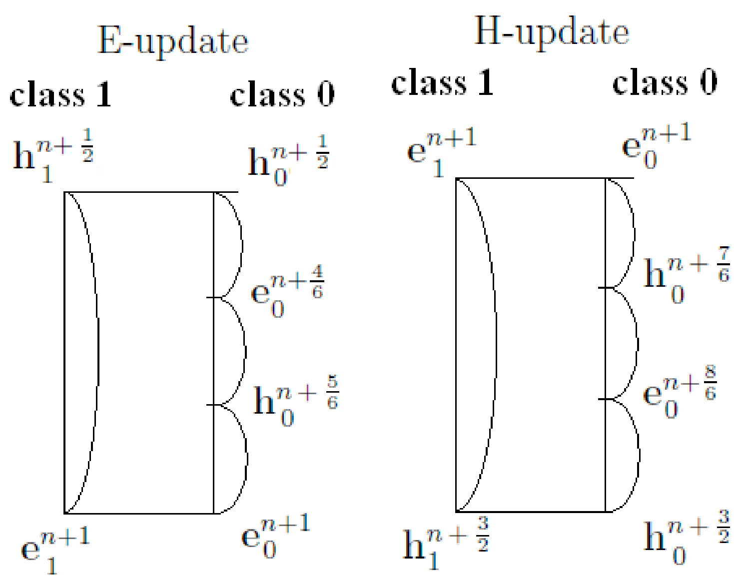
- FEM Mesh is mapped to a CUDA Grid
- Finite element is mapped to CUDA $d_i \times 1$ thread block, where d_i are the local DOFs. Maximum 512 threads per block on Tesla C0170
- $\mathbf{E}_h|_{K_i} = \sum_{n=0}^{d_i} e_{in}(t)\mathbf{w}(\mathbf{r})_{in}$. Every DOF e_{in} is updated by a CUDA thread. Same for $\mathbf{H}_h$ and h_{in}.
- 1 GPU 10x speed up vs 1 CPU in double precision.



N. Godel, et.al., Scalability of Higher-Order Discontinuous Galerkin FEM Computations for Solving Electromagnetic Wave Propagation Problems on GPU Clusters.

CUDA and Local Time Stepping

- Use global memory to store the matrices in the update equations. The data are copied once before the time stepping and reused through the time marching.
- Use shared memory for field data.



- LTS algorithm is unchanged. However now each class executes multiple CUDA Grids for its elements
- Data are copied to CPU-Host every $N_{plot}\delta t$ using *cudaMemcpyAsync* to overlap computation with communication.

CUDA Kernels

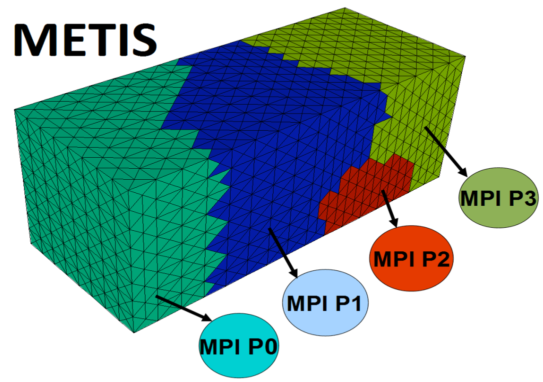
- LE_Vol_kernel and LH_Vol_kernel, update contributions from volume terms for leapfrog E and leapfrog H accordingly.
- LE_Surf_kernel and LH_Surf_kernel, update contributions from flux terms for leapfrog E and leapfrog H accordingly

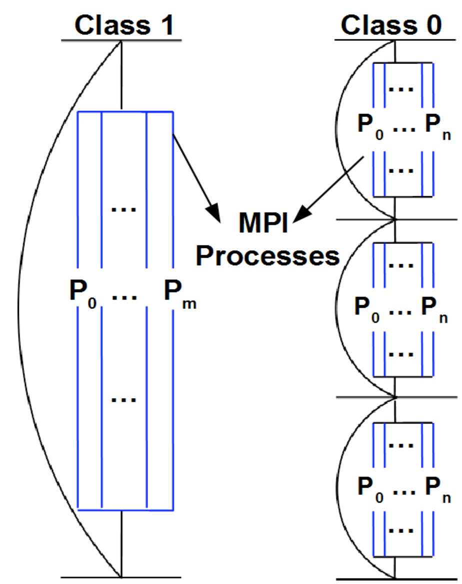
MPI/GPU Coarse Grained Level

- Quadro FX 5800 has only 4GB of global memory.
- Big problems require a multi-GPU approach like MPI+CUDA.

MPI General Layout

- MPI is used is the coarse grained parallelization level
- Metis partitions the elements to sub-domains.
- Each MPI process works on one sub-domain.
- Within a class all MPI processes P_i that have elements in the class will work to perform the update
- At the end of each class update we communicate only between processes that work on neighboring sub-domains.
- Each MPI process write its own data to disk. In post processing we use pvtu files and vtkMergeCells of VTK to merge partitions.



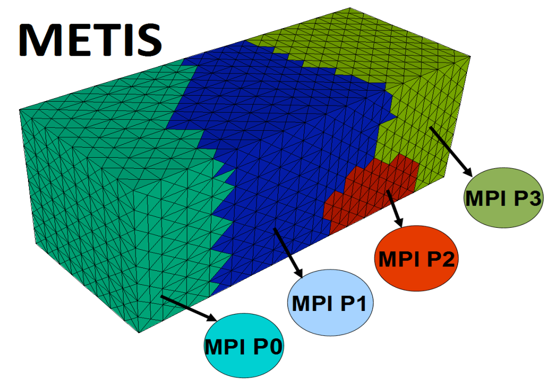


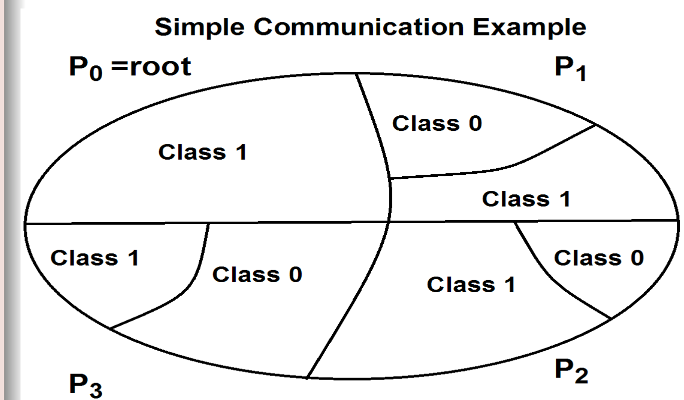
MPI/GPU Coarse Grained Level

- Quadro FX 5800 has only 4GB of global memory.
- Big problems require a multi-GPU approach like MPI+CUDA.

MPI General Layout

- MPI is used is the coarse grained parallelization level
- Metis partitions the elements to sub-domains.
- Each MPI process works on one sub-domain.
- Within a class all MPI processes P_i that have elements in the class will work to perform the update
- At the end of each class update we communicate only between processes that work on neighboring sub-domains.
- Each MPI process write its own data to disk. In post processing we use pvtu files and vtkMergeCells of VTK to merge partitions.



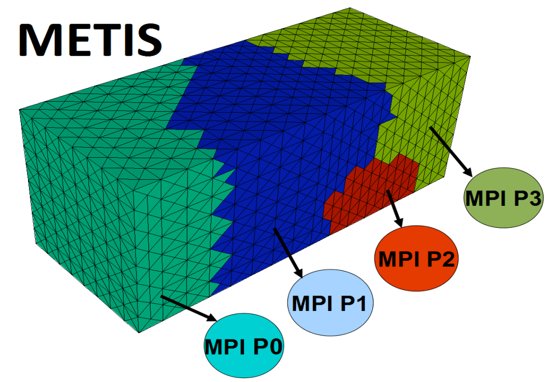


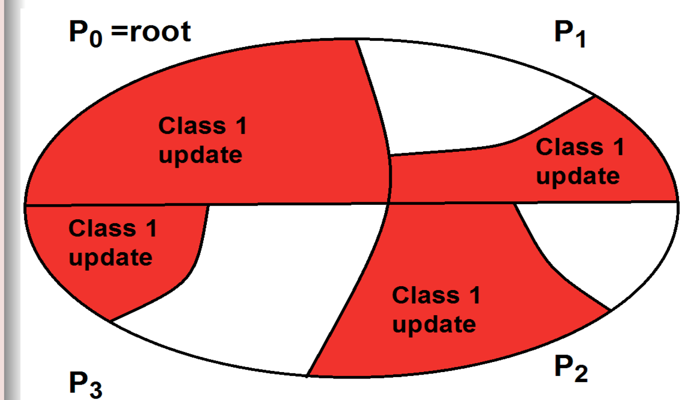
MPI/GPU Coarse Grained Level

- Quadro FX 5800 has only 4GB of global memory.
- Big problems require a multi-GPU approach like MPI+CUDA.

MPI General Layout

- MPI is used is the coarse grained parallelization level
- Metis partitions the elements to sub-domains.
- Each MPI process works on one sub-domain.
- Within a class all MPI processes P_i that have elements in the class will work to perform the update
- At the end of each class update we communicate only between processes that work on neighboring sub-domains.
- Each MPI process write its own data to disk. In post processing we use pvtu files and vtkMergeCells of VTK to merge partitions.



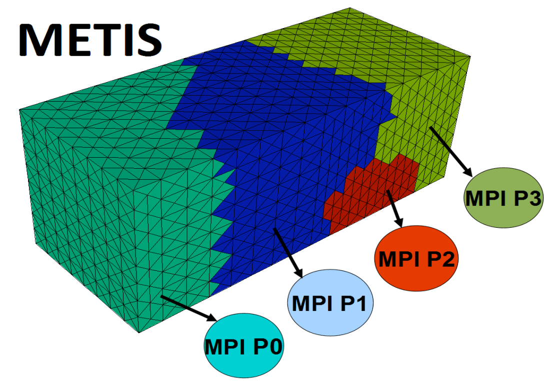


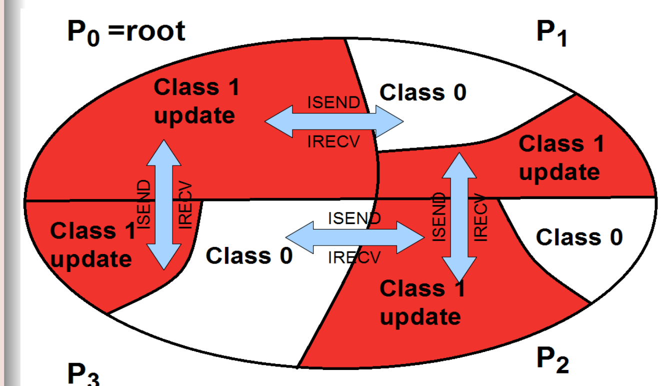
MPI/GPU Coarse Grained Level

- Quadro FX 5800 has only 4GB of global memory.
- Big problems require a multi-GPU approach like MPI+CUDA.

MPI General Layout

- MPI is used is the coarse grained parallelization level
- Metis partitions the elements to sub-domains.
- Each MPI process works on one sub-domain.
- Within a class all MPI processes P_i that have elements in the class will work to perform the update
- At the end of each class update we communicate only between processes that work on neighboring sub-domains.
- Each MPI process write its own data to disk. In post processing we use pvtu files and vtkMergeCells of VTK to merge partitions.



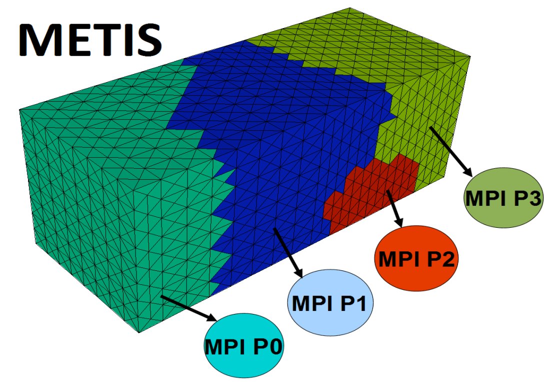


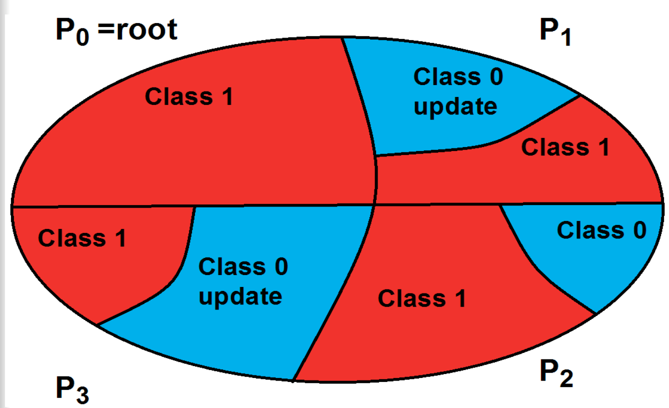
MPI/GPU Coarse Grained Level

- Quadro FX 5800 has only 4GB of global memory.
- Big problems require a multi-GPU approach like MPI+CUDA.

MPI General Layout

- MPI is used is the coarse grained parallelization level
- Metis partitions the elements to sub-domains.
- Each MPI process works on one sub-domain.
- Within a class all MPI processes P_i that have elements in the class will work to perform the update
- At the end of each class update we communicate only between processes that work on neighboring sub-domains.
- Each MPI process write its own data to disk. In post processing we use pvtu files and vtkMergeCells of VTK to merge partitions.



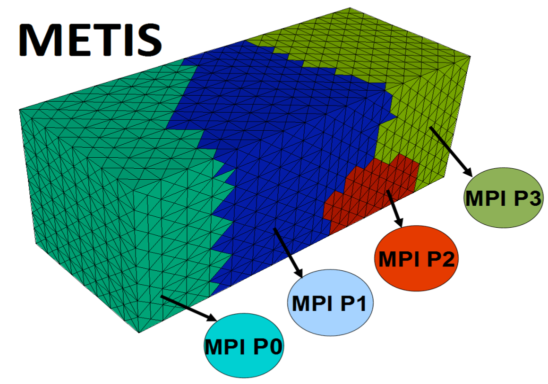


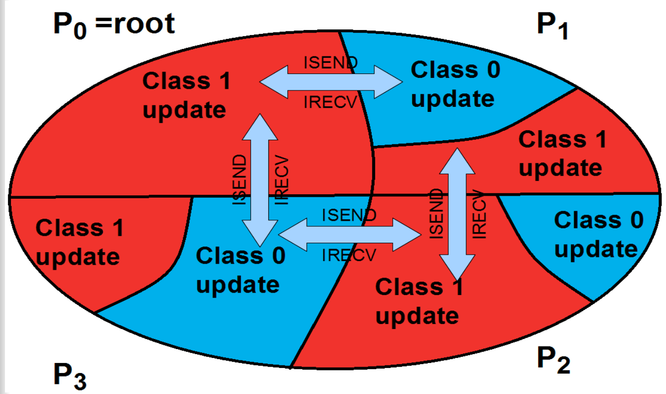
MPI/GPU Coarse Grained Level

- Quadro FX 5800 has only 4GB of global memory.
- Big problems require a multi-GPU approach like MPI+CUDA.

MPI General Layout

- MPI is used is the coarse grained parallelization level
- Metis partitions the elements to sub-domains.
- Each MPI process works on one sub-domain.
- Within a class all MPI processes P_i that have elements in the class will work to perform the update
- At the end of each class update we communicate only between processes that work on neighboring sub-domains.
- Each MPI process write its own data to disk. In post processing we use pvtu files and vtkMergeCells of VTK to merge partitions.





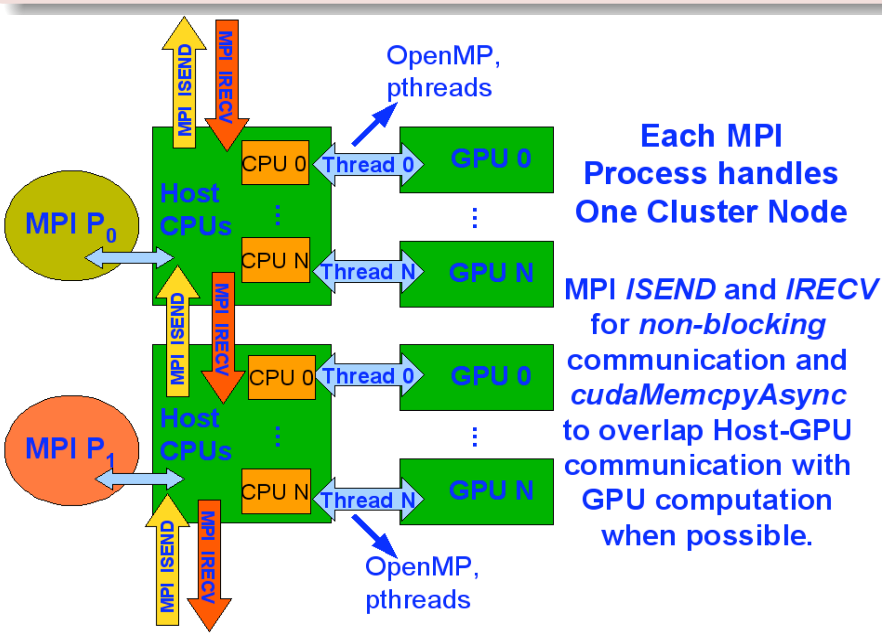
MPI+CUDA

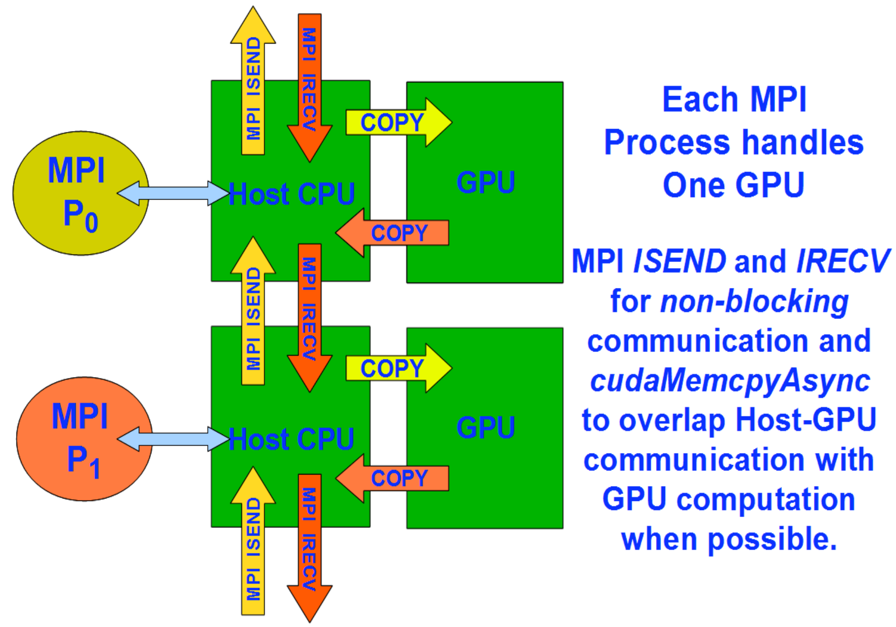
Approach 1

- Use host threads to run multiple GPUs on each cluster node, and MPI for inter-node communications. (MPI processes is equal to the number of nodes).

Approach 2

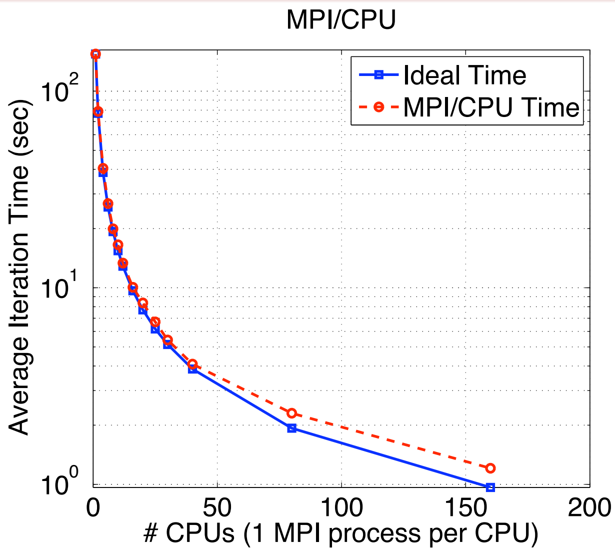
- Use MPI and run one process per GPU.(MPI processes is equal to the number of GPUs)

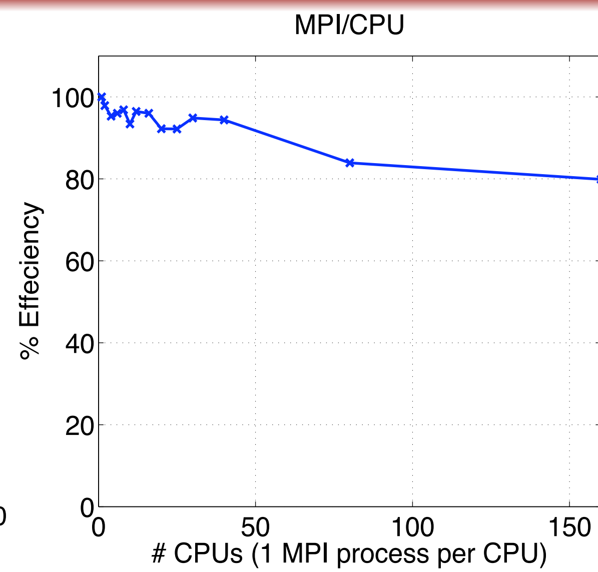




- Approach 2 is simpler, since it uses one API. However, the MVAPICH implementation of MPI in the Ohio SuperComputer Center will use shared memory to communicate between MPI processes that reside in the same cluster node. Therefore, there is no additional overhead of Approach 1 compared to Approach 2.

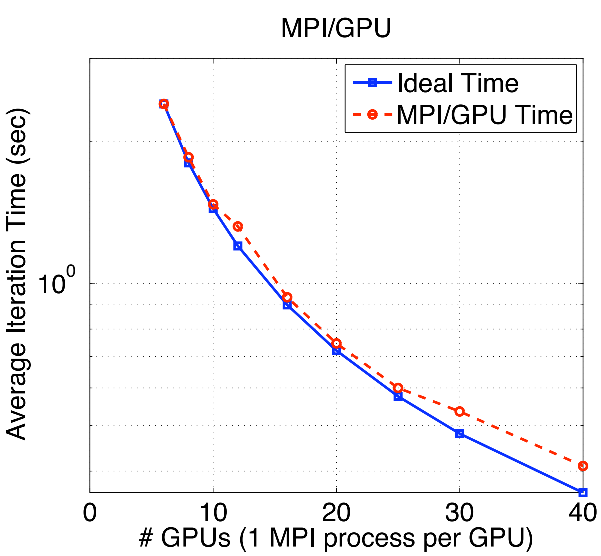
Performance Analysis

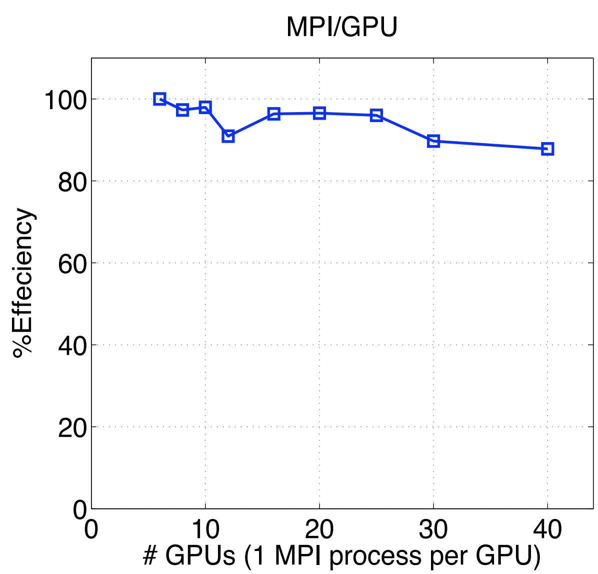


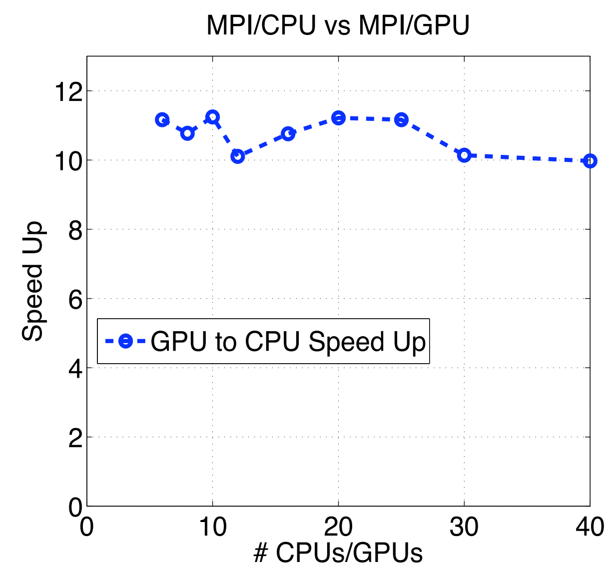


Computational Statistics

- Coated Sphere
- 1.6×10^6 elements, 36×10^6 DOFs and 2 LTS classes

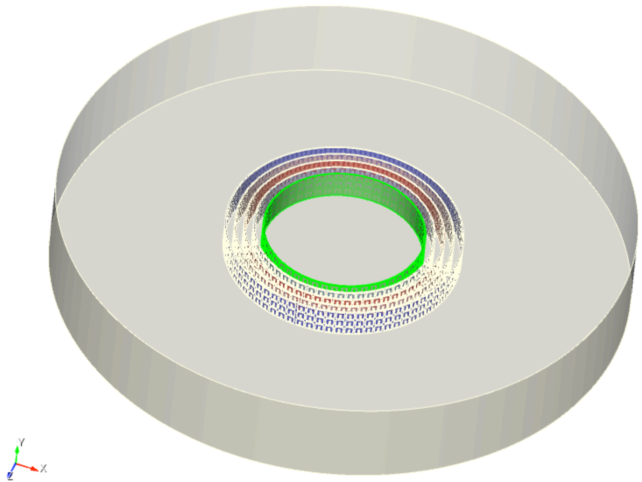


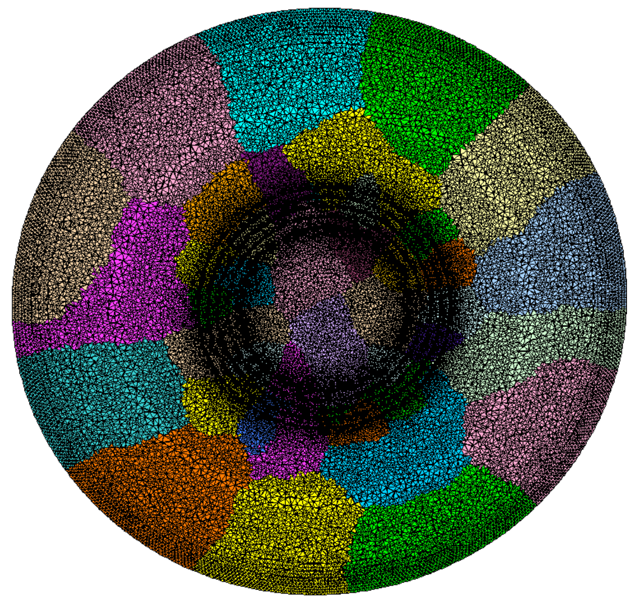


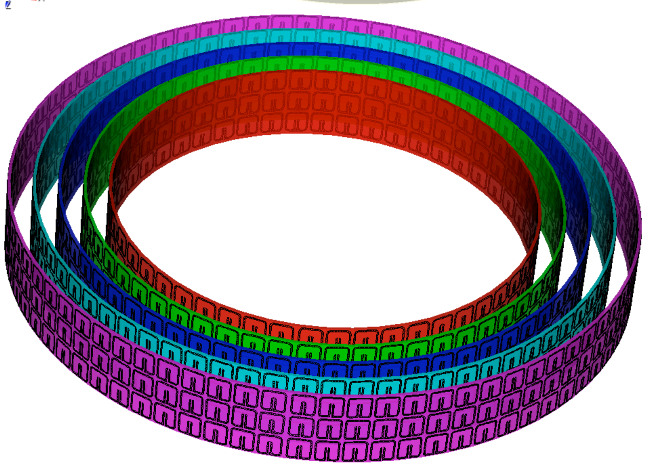


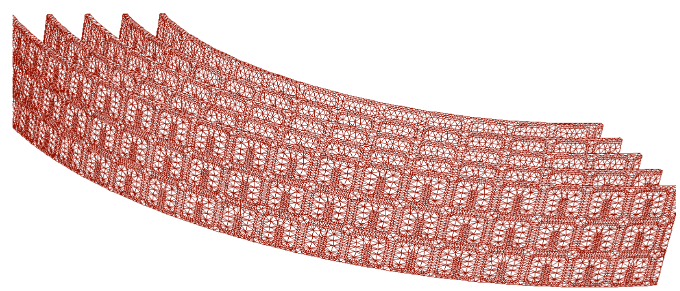
Study of SRR Cloaking Device in Time-Domain

- SRR metamaterial cloaking device [Schurig, D. et. al., *Science*, vol 314, 2006]. Geometry featuring fine detail ($\delta t_{min} = 1.11e-14$, $\delta t_{max} = 5.38e-13$)
- Substrate RT Duroid 5780 $\epsilon_r = 2.33$ and 1^{st} order ABC for domain truncation
- ($h_{air} \approx \lambda/15$, $h_{SRR} \approx \lambda/50$) 6,685,671 elements, 150×10^6 unknowns ($p = 1$ elements were used).







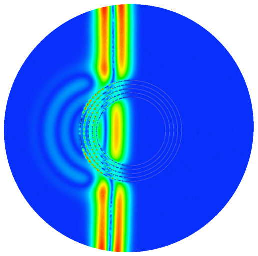


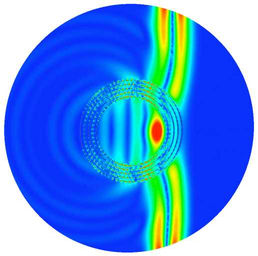
Study of SRR Cloaking Device in Time-Domain

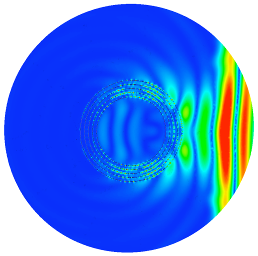
- Neuman pulse with 3_{dB} bandwidth at 8-11 GHz (Xband). **E** field polarization along cylinder axis
- 22 compute nodes, with 8 CPU cores, 2 Quadro FX 5800 GPUs and 24GB of RAM each at Ohio Supercomputing Center(OSC) Glenn cluster.
- We simulated the same problem using the 2 CPUs per node and also using 2 GPUs per node. All simulations were done in double precision arithmetic.

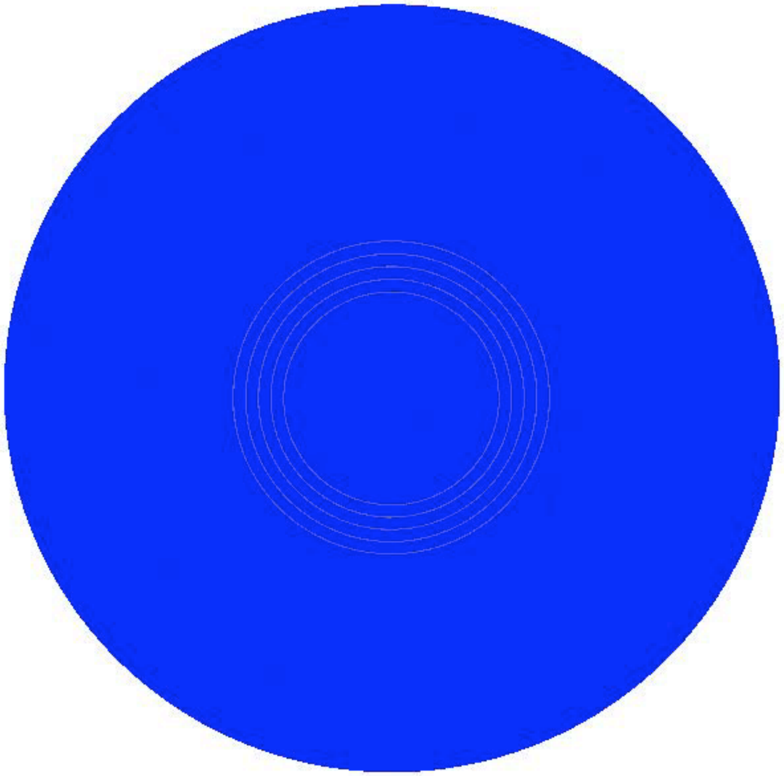
Computational Statistics

Tetrahedra	6,685,671 elements
DOFs	150×10^6 DOFs
# LTS Classes	4
22 CPUs LTS	225.83 hrs
44 CPUs LTS	117.17 hrs
80 CPUs LTS	66.75 hrs
44 GPUs LTS	11.88 hrs
GPU Gain	9.86









Conclusions

Conclusions

- We have presented an approach to map a DGTD method with local time stepping on GPUs.
- To account for the limited amount of memory in one GPU we presented an MPI/GPU approach suitable for large problems
- A speed up of 10x times compared to MPI/CPU was obtain for double precision arithmetic and a 90% parallelization effeciency was achived up to 40 GPUs
- Finally a study of cloacking device was performed in time domain to show the potential of the proposed MPI/GPU approach.