DG-FEM for Time-Dependent Problems
Lecture 3: Extensions, Applications and Software

Jan S. Hesthaven
Jan.Hesthaven@Brown.edu
www.cfm.brown.edu/people/jansh/

Division of Applied Mathematics
Brown University, Box F
Providence, RI 02912, USA
Overview of Lecture 3

- Extension to Multiple Dimensions
  - Formulations
  - Nodes, modes, and operators in 2D/3D
  - Overview of analysis
- Focus on Maxwell's equations
  - Time-domain schemes
  - Spurious modes and how to control them
- Briefly on other applications
- Available software - Sledge++
The Formulation I

Let us consider the more general problem

\[
\frac{\partial u(\mathbf{x}, t)}{\partial t} + \nabla \cdot f(u(\mathbf{x}, t), \mathbf{x}, t) = 0 , \quad x \in \Omega \subseteq \mathbb{R}^2 ,
\]

\[
u(\mathbf{x}, t) = g(x, t) , \quad x \in \partial \Omega \quad \hat{n} \cdot f_u < 0 ,
\]

\[
u(x, 0) = f(x) .
\]

As usual, we represent \( \Omega \) as

\[
\Omega = \bigcup_{k=1}^{K} D^k ,
\]

where \( D^k \) can be general elements
The Formulation II

We continue by representing the global solution as

\[ u(x, t) \simeq u_h(x, t) = \sum_{k=1}^{K} u_{kh}(x, t) . \]

with the local solution being

\[ x \in D^k : u_{kh}(x, t) = \sum_{i=1}^{N_p} u_{kh}(x_i, t) l_i(x) . \]

We can proceed exactly as in 1D and form the residual

\[ R_h(x, t) = \frac{\partial u_h(x, t)}{\partial t} + \nabla \cdot f_h(x, t) . \]
If we require this to vanish in the local sense that

$$\int_{D^k} R_h(x, t) \psi^k_m(x) \, dx = 0 ,$$

for the $K \times N_p$ testfunctions, $\psi^k_m(x)$, then integration by parts once yields the weak DG-FEM

$$\forall k, m : \int_{D^k} \left[ \frac{\partial u_h^k}{\partial t} l_m(x) - f_{h} \cdot \nabla l_m(x) \right] \, dx = - \int_{\partial D^k} \hat{n} \cdot f^* l_m(x) \, dx ,$$

and twice the strong DG-FEM

$$\forall k, m : \int_{D^k} \left[ \frac{\partial u_h^k}{\partial t} + \nabla \cdot f_{h} \right] l_m(x) \, dx = \int_{\partial D^k} \hat{n} \cdot [f_{h} - f^*] l_m(x) \, dx .$$
The Formulation IV

The missing piece is the numerical flux, $f^*$. For the linear case (see later) one can develop exact upwind fluxes using Riemann variables as in 1D.

For the general nonlinear case, a suitable choice is local Lax-Friedrich flux

$$f^*(a, b) = \frac{f(a) + f(b)}{2} + \frac{C}{2} \hat{n}(a - b),$$

where $C$ is the local maximum of the directional flux Jacobian, i.e.,

$$C = \max_{u \in [a, b]} |\hat{n} \cdot \frac{\partial f}{\partial u}|.$$
Let us first consider this polynomial representation

\[\forall \mathbf{x} \in D : \ u(\mathbf{x}, t) \simeq u_h(\mathbf{x}, t) = \sum_{i=1}^{N_p} \hat{u}_i(t) \psi_i(\mathbf{x}) \in P_n\]

As in 1D, we need to be careful when we specify

- The form of the polynomial basis, \( \psi_i(\mathbf{x}) \in P_n \)
- How to compute/define \( \hat{u}_i \)
Addressing these concerns, let us limit the complexity by considering a standard triangle

\[ I = \{(r, s) \in \mathbb{R}^2 | (r, s) \geq -1; r + s \leq 0\} \]

To construct this mapping, use barycentric coordinates

\[ 0 \leq \lambda^i \leq 1, \quad \lambda^1 + \lambda^2 + \lambda^3 = 1 \]

Any point in the simplex, \( D^k \), can be expressed as

\[ x = \lambda^2 \nu^1 + \lambda^3 \nu^2 + \lambda^1 \nu^3 \]

where \((\nu_1, \nu_2, \nu_3)\) are the 3 vertices of \( D^k \)
In a similar fashion, we can express

\[
\begin{pmatrix}
  r \\
  s
\end{pmatrix} = \lambda^2 \begin{pmatrix}
  -1 \\
  -1
\end{pmatrix} + \lambda^3 \begin{pmatrix}
  1 \\
  -1
\end{pmatrix} + \lambda^1 \begin{pmatrix}
  -1 \\
  1
\end{pmatrix}.
\]

Combining these we recover

\[
\lambda^1 = \frac{s + 1}{2}, \quad \lambda^2 = -\frac{r + s}{2}, \quad \lambda^3 = \frac{r + 1}{2},
\]

and the direct mapping.

\[
x = -\frac{r + s}{2} v^1 + \frac{r + 1}{2} v^2 + \frac{s + 1}{2} v^3 = \Psi(r).
\]
Recall

\[ x = -\frac{r+s}{2}v^1 + \frac{r+1}{2}v^2 + \frac{s+1}{2}v^3 = \Psi(r) . \]

- Mapping is linear in \((r, s)\), i.e., an affine mapping
- Constant mapping Jacobian – as in 1D
- Simple computation of mapping metric

\[
\begin{align*}
rx &= \frac{ys}{J} , & ry &= -\frac{xs}{J} , & sx &= -\frac{yr}{J} , & sy &= \frac{xr}{J} , \\
J &= xr ys - xs yr .
\end{align*}
\]
Let us now focus on

\[ u(r) = \sum_{n=1}^{N_p} \hat{u}_n \psi_n(r) = \sum_{i=1}^{N_p} u(r_i) l_i(r), \]

where

\[ N_p = \frac{(N + 1)(N + 2)}{2}, \]

and define \( \hat{u}_n \) such that

\[ V \hat{u} = u, \quad V_{ij} = \psi_j(r_i). \]

- We need to define \( \psi_j(r) \).
- We need to identify, \( r_i \).
An orthonormal basis is found from the canonical basis

\[ \psi_m(r) = \sqrt{2} P_i(a) P_j^{(2i+1,0)}(b)(1 - b)^i, \quad a = 2 \frac{1 + r}{1 - s} - 1, \quad b = s, \]
The position of the nodes is more 'arbitrary' (several solutions) – need to be good for interpolation (see text)
With this in place, we can continue exactly as in 1D with

\[ u = V \hat{u} \, , \, \, V^T l(r) = \tilde{\psi}(r) \, , \, \, V_{ij} = \psi_j(r_i) \, . \]

Furthermore,

\[ M^k = J^k (VV^T)^{-1} \, . \]

and

\[ \frac{d}{dx} = r_x D_r + s_x D_s \, , \, \, \frac{d}{dy} = r_y D_r + s_y D_s \, , \]

with \( D_r \) and \( D_s \) defined just as in 1D, i.e.

\[ D_{r,(i,j)} = \frac{dl_j(r_i)}{dr} \, , \, \, D_{s,(i,j)} = \frac{dl_j(r_i)}{ds} \, . \]
Lagrange polynomials for $N = 4$. 
The final term we need to evaluate are of the type
\[ \int_{\partial D^k} \hat{n} \cdot g_h l_i(x) \, dx , \]
where \( g_h \) is some known polynomial.

This separates into terms like
\[ \int_{\text{face}_1} \hat{n} \cdot g_h l_i(x) \, dx = \sum_{j=0}^{N} \hat{n} \cdot g_j \int_{\text{face}_1} l_j(x) l_i(x) \, dx . \]

requiring us to compute (for a nodal basis)
\[ M_{ij} = \int_{\text{face}_1} l_j(x) l_i(x) \, dx \quad \Rightarrow \quad M^1 = J^1(V^{1D}(V^{1D})^T)^{-1} . \]
So – should one use modes or nodes?

- Nodes gives a nicely separated basis – reduce cost and increase sparsity at higher order
- Nodes are (likely) preferable for nonlinear problems
- For linear problems, the modes may be simpler – but are more expensive to manipulate
- More advanced issues like preconditioning is often dealt with easier for nodes
Overview of Analysis

All main results for smooth problems from 1D carry over to 2D/3D – except

\[ \| u - u_h \| \leq h^{N+1/2} \| u \|_p , \]

which can not be improved for general grids (Peterson’91).

Optimal rates only simple rectangular grids (Raviart et al’74)

Stability can be established using energy methods

Approximation theory/error-estimates for multivariate approx-
imation/interpolations is incomplete – the computational evi-
dence is strong.
Generalizations to 3D etc

We have not said/done anything which limits the scheme

- Extensions to 3D using tetrahedra is straightforward
- The generalization to hybrid grids is possible
- Non-conforming grids only requires attention to surface terms (but care with spectral properties)
- Use of custom designed elements are possible
- Special/Non-polynomial local basis sets can be explored
- etc
Maxwell’s Equations I

Much is driven by new technology

- Broad band radars and SAR
- Complex/nonlinear materials
- Highly phase sensitive components – optics

- and new interests

- Sensitivity to design variations
- Rough surfaces and randomness
- Kinetic plasma physics
Maxwell’s Equations II

We consider Maxwell’s equations

\[ \varepsilon_r \frac{\partial E}{\partial t} = \nabla \times H + J, \quad \mu_r \frac{\partial H}{\partial t} = -\nabla \times E, \]

subject to boundary conditions at material interfaces as

\[ \hat{n} \times (E_1 - E_2) = 0, \quad \hat{n} \times (H_1 - H_2) = 0, \]

and conditions at metallic boundaries as

\[ \hat{n} \times E = 0, \quad \hat{n} \cdot B = 0. \]

Also

\[ Z = \sqrt{\frac{\mu_r}{\varepsilon_r}}, \quad Y = Z^{-1}. \]
Maxwell’s Equations III

Linearity of the equations yields the scattered field form

\[ \varepsilon_r \frac{\partial E^s}{\partial t} = \nabla \times H^s + \sigma E^s - (\varepsilon_r - \varepsilon_r^i) \frac{\partial E^i}{\partial t} + (\sigma - \sigma^i) E^i, \]

\[ \mu_r \frac{\partial H^s}{\partial t} = -\nabla \times E^s - (\mu_r - \mu_r^i) \frac{\partial H^i}{\partial t}, \]

\[ Q \frac{\partial q}{\partial t} + \nabla \cdot F(q) = S, \]

\[ q = \begin{bmatrix} E \\ H \end{bmatrix}, \quad F_i(q) = \begin{bmatrix} -e_i \times H \\ e_i \times E \end{bmatrix}, \quad Q = \begin{bmatrix} \varepsilon_r & 0 \\ 0 & \mu_r \end{bmatrix}. \]
Challenges in solving these equations include:

- Electrically large problems with long time wave propagation
- Geometric complexity and material variation
- Scale separation due to geometries and physics
- Unbounded domains
- Performance of solver

A DG-FEM scheme is an excellent candidate for this.
We consider the DG-FEM method on strong form

\[ \int_{D^k} \left( Q \frac{\partial q_h}{\partial t} + \nabla \cdot F_h \right) l_m(x) \, dx = - \int_{\partial D^k} \hat{n} \cdot [F_h - F^*_h] \, l_m(x) \, dx, \]

where the numerical flux is upwind as

\[ n \cdot [F_h - F^*_h] = \begin{cases} \frac{\bar{Z}^{-1}}{\bar{Y}^{-1}} n \times (\alpha n \times [E_h] - Z^+ [H_h]) \\ \frac{\bar{Z}^{-1}}{\bar{Y}^{-1}} n \times (\alpha n \times [H_h] + Y^+ [E_h]) \end{cases}. \]

Note that \( \alpha = 0 \) is a central flux.
Maxwell’s Equations VI

- **Local Stability** – i.e. elementwise
- **Global Stability** – i.e. stability of the coupling.

The scheme is error bounded for \( t \in [0, T] \)

\[
\| \varepsilon_h(t) \| \leq C \frac{h^\sigma}{N^p} \left[ \| q(0) \|_p + t \frac{N^2}{h} \max_{s \in [0,t]} \| q(s) \|_p \right]
\]

\[
\varepsilon_h(t) = q(t) - q_h(t) , \quad \sigma = \min(p, N + 1) , \quad \text{i.e., if consistent, then convergent}
\]

As we have seen for much simpler cases.
Maxwell’s Equations VII

Uses METIS for grid-distribution and bandwidth minimization for cache-performance. Kernel in Fortran/MPI, rest in C.

<table>
<thead>
<tr>
<th>Polynomial order ((n))</th>
<th>Degrees of freedom ((\times 10^6))</th>
<th>Number of processors</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>7.4</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13.7(16)</td>
</tr>
<tr>
<td>3</td>
<td>14.8</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6.4(7.2)</td>
</tr>
<tr>
<td>4</td>
<td>25.8</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.6(4.0)</td>
</tr>
<tr>
<td>5</td>
<td>41.3</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.6(1.6)</td>
</tr>
</tbody>
</table>
Maxwell’s Equations VIII

A more extreme case is 120.000.000 DOF CEM testcase

<table>
<thead>
<tr>
<th>Number of Processors</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative time per RK step</td>
<td>1.00</td>
<td>0.48</td>
<td>0.24</td>
<td>0.14</td>
</tr>
<tr>
<td>Ideal Scaling</td>
<td>1.00</td>
<td>0.50</td>
<td>0.25</td>
<td>0.13</td>
</tr>
</tbody>
</table>

- 245.000 elements at 6th order
- Computations done on IBM SP
- Kernel reaches close to 500 GFlops on 512 processors (75% peak).
Maxwell’s Equations IX

Consider first

- A 2D cavity of \((x, y) \in [-1, 1] \times [-0.25, 0.25]\), PEC Walls
- Tiled using 8 triangles with edge length of 0.5.
Simple plane wave propagation through a prims.
Consider the 2D TM-polarized Maxwell’s equations

\[
\mu_r \frac{\partial H_x}{\partial t} = - \frac{\partial E_z}{\partial y},
\]

\[
\mu_r \frac{\partial H_y}{\partial t} = \frac{\partial E_z}{\partial x},
\]

\[
\varepsilon_r \frac{\partial E_z}{\partial t} = \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y},
\]

Similar results have been obtained for TE-polarization.

- \( ka = 15\pi \) PEC cylinder, 854 curvilinear triangles
- \( ka = 2\pi \) PEC cylinder – \( p \)-convergence
- \( ka = 7\pi, \varepsilon_r = 2 \) material cylinder – 1020 triangles.
Grid and details \((ka = 15\pi)\) showing the bodyconforming nature of the grid.
Spectral convergence of RCS.

![Graph of RCS vs. \( \theta \) for different values of \( n \)](image)

![Graph of \( ||\delta RCS(\text{dbm})||_2 \) vs. \( n \)](image)
Maxwell’s Equations XV

\[ \text{ka} = 7\pi, \ \varepsilon_r = 2 \]
Maxwell’s Equations XVI

- $ka = 10$ PEC sphere, 3,000 elements, edge-length at sphere approx $\frac{4}{5}\lambda$.
- EMCC Business card problem, 27,000 elements, 4th order.
- EMCC narrow Cone-Sphere problem, 270,000, 4th order.
- $5\lambda$ long, $\varepsilon_r = 2.25$ cylinder, 67,000 elements, 4th order.
- 600 MHz illumination of aircraft. 245,000 elements, order 1-3.
- 3.6 GHz illumination of aircraft. 245,000 elements, order 5-7.
- 2.4 GHz Missile with uncertainty in illumination
3D sanity test for plane wave scattering from $ka = 10$ sphere.
Maxwell’s Equations XVIII

Details of 3D bodyconforming grid.
Comparison with RCS computed using CFIE (Hanscom AFB).
Maxwell’s Equations XX

Test of surface wave propagation.

a)

b)
Maxwell’s Equations XXI

Test of penetration/refraction/reflection.

\[ \varepsilon_r = 2.25 \]

\[ k_i \]

\[ E, H \]

\[ \theta \]

\[ \text{RCS (θ,0)} \text{ (dBm)} \]

\[ \text{RCS (θ,90)} \text{ (dBm)} \]

\[ \text{θ} \]

\[ 0 \quad 30 \quad 60 \quad 90 \quad 120 \quad 150 \quad 180 \]

\[ -30 \quad -20 \quad -10 \quad 0 \quad 10 \quad 20 \quad 30 \]

\[ -30 \quad -20 \quad -10 \quad 0 \quad 10 \quad 20 \quad 30 \quad 40 \quad 50 \]
Maxwell’s Equations XXIII
Comparison of RCS for 3 different orders - 600MHz
Very large scale scattering problem – 3.6GHz - \( > 300\lambda \).

(a) TE Polarization
\( \theta^\text{inc} = 90 \text{ deg} \)
\( \phi^\text{inc} = 0 \text{ deg} \)

(b) TE Polarization
\( \theta^\text{inc} = 90 \text{ deg} \)
\( \phi^\text{inc} = 0 \text{ deg} \)
.. but what about

$$\nabla \cdot \mu_r H = \nabla \cdot \varepsilon_r E = 0$$.

Often not included for computational convenience.

Consequences ..

- The problem is easier to solve
- .. but the solutions are not the same!

This applies both to time-domain and frequency domain solvers.
Scheme preserves divergence to the order of the scheme

\[ \| \nabla \cdot \varepsilon_h(t) \| \leq C \frac{h^{\sigma-1}}{n^{p-2}} \left[ \| q(0) \|_p + \frac{N^{3/2}}{h} \max_{s \in [0,t]} \| q(s) \|_p \right] \]

So accuracy is reasonable.

\[ \ldots \text{unfortunately the situation is a little more complex than that.} \]
Consider the Maxwell eigenvalue problem on curl-curl form

$$\nabla \times \begin{bmatrix} \mu_r^{-1} \\ \varepsilon_r^{-1} \end{bmatrix} \nabla \times \begin{bmatrix} E \\ H \end{bmatrix} = \omega^2 \begin{bmatrix} \varepsilon_r E \\ \mu_r H \end{bmatrix}, \ x \in \Omega,$$

subject to boundary conditions

$$\mathbf{n} \times \mathbf{E} = 0 \quad \text{or} \quad \frac{\partial H}{\partial n} = 0, \ x \in \partial \Omega.$$

- Reduces the unknown field to $\mathbf{E}$ or $\mathbf{H}$.
- Allows for a selfadjoint form.
Solving the Maxwell eigenvalue problem is interesting in its own right

- Resonators and cavities
- Antennas

... but an understanding also has derived qualities

- Time behavior of modes
- Dispersion characteristics
- Spurious modes
Maxwell’s Equations XXXI

We shall consider the 2D problem

\[ \nabla \times \nabla \times E = \omega^2 E \quad , \quad x \in [-1, 1]^2 \]
\[ n \times E = 0 \quad , \quad x \in \partial \Omega \]

We solve the problem using

- A 2nd order nodal Galerkin finite element scheme
- A fully unstructured grid
Problem - poor approximation of large null space.

Any \( E = \nabla \phi \) implies \( \omega = 0 \).
A cross-hatch grid ensures a correct null space, but ...

<table>
<thead>
<tr>
<th>$h$</th>
<th>1.00</th>
<th>0.500</th>
<th>0.250</th>
<th>0.125</th>
<th>Exact</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.0661</td>
<td>1.0170</td>
<td>1.0043</td>
<td>1.0011</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>1.0661</td>
<td>1.0170</td>
<td>1.0043</td>
<td>1.0011</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>2.2035</td>
<td>2.0678</td>
<td>2.0171</td>
<td>2.0043</td>
<td>2.0000</td>
</tr>
<tr>
<td>$\omega^2$</td>
<td>4.8634</td>
<td>4.2647</td>
<td>4.0680</td>
<td>4.0171</td>
<td>4.0000</td>
</tr>
<tr>
<td></td>
<td>4.8634</td>
<td>4.2647</td>
<td>4.0680</td>
<td>4.0171</td>
<td>4.0000</td>
</tr>
<tr>
<td></td>
<td>6.4846</td>
<td>5.3971</td>
<td>5.1063</td>
<td>5.0267</td>
<td>5.0000</td>
</tr>
<tr>
<td></td>
<td>6.4846</td>
<td>5.3971</td>
<td>5.1063</td>
<td>5.0267</td>
<td>5.0000</td>
</tr>
<tr>
<td></td>
<td>6.1338</td>
<td>5.6712</td>
<td>5.9229</td>
<td>5.9807</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>11.092</td>
<td>8.8141</td>
<td>8.2713</td>
<td>8.0685</td>
<td>8.0000</td>
</tr>
</tbody>
</table>
Implications of this is

- Computation of wrong eigensolutions
- Polluted solutions of frequency solutions
- Polluted time-domain solution

- Curl-conforming elements
- Penalizing by $\nabla \cdot E$
- Higher order helps
Maxwell’s Equations XXXV

Same problem as before solved with DG-FEM and central fluxes

No null space pollution for the DG-FEM case!
Maxwell’s Equations XXXVI

For 2D there appears to be no need to do anything for conforming grids (non-conforming is a different story). Verified for

- Having both smooth and nonsmooth eigenfunctions
- Using different orders of approximation
- Using both highly regular and fully unstructured grids.
- Nothing special is done at singular points.
- Eigenvalues computed using Arpack.
Maxwell’s Equations XXXVIII

\[
\begin{array}{cccc}
\varepsilon=1 & \varepsilon=0.1 \\
\varepsilon=0.1 & \varepsilon=1 \\
\end{array}
\]

![Graph showing error in computed eigenvalue vs ID of computed eigenvalue with different markers for each order.](image)

- Error in computed eigenvalue
- ID of computed eigenvalue
- Markers for different orders (\(p=1\) to \(p=6\))
Let us again consider the simple test case

$$\nabla \times \nabla \times \mathbf{E} = \omega^2 \mathbf{E} , \quad x \in [-1, 1]^3$$

$$\mathbf{n} \times \mathbf{E} = 0 , \quad x \in \partial \Omega$$
Maxwell’s Equations XL

... so we are back to square one?

Not quite as we have a lot of freedom in the formulation

- Stabilization by penalization
- Change of numerical flux, $F_h^*$. 
We consider the locally modified form

\[ \omega^2 (E_h, \phi)_{D^k} = \left( \tilde{\nabla}_D \times \phi, p_h \right)_{D^k} - \tau (n \times [\phi], n \times [E_h]) \partial_{D^k} \]

\[ (p_h, \phi)_{D^k} = \left( \tilde{\nabla}_D \times E_h, \phi \right)_{D^k} . \]

- Operators maintain symmetry
- \( \tau \) measures stabilization – by scaling

\[ \tau \propto \frac{N(N + 1)}{h} . \]
Consider the equivalent (local) 1st order form

\[
\begin{align*}
i \omega (H_h, \psi)_{D^k} &= - \left( \tilde{\nabla}_D \times E_h, \psi \right)_{D^k}, \\
i \omega (E_h, \psi)_{D^k} &= \left( \tilde{\nabla}_N \times H_h, \psi \right)_{D^k} + \sqrt{\tau} \left( n \times Q, \psi \right)_{\partial D^k}, \\
i \omega (Q, \psi)_{\partial D^k} &= \sqrt{\tau} \left( n \times [E_h], \psi \right)_{\partial D^k}.
\end{align*}
\]

For \( Q \neq 0 \), \( E_h \) must be curl-conforming for \( \omega = 0 \).

Increasing \( \tau \) forces the solution towards a curl-conforming discretization.

Discretizations based on curl conforming elements are free of null space pollution.
Clearly Improves matters
’Bad’ modes pushed about $\tau$ away.
The central flux is just one among many, in particular one could use an upwind flux

\[ n \times [\psi - \psi^*] = \frac{1}{2} (n \times [\psi] - \alpha n \times n \times [\psi]) . \]

- Dissipative – standard use in time-domain
- Operator loose symmetry
- Recall – severe damping for high \( kh \)
- – and small spurious modes also severely damped.
PS - The results are for 1st order form.
In the time-domain this means that

- For 2D computations there are no problems to be expected.
- The use of energy conserving central fluxes is not a good idea in 3D.
- Ramping of initial conditions is likely to lessen the impact of spurious modes.
- Upwind fluxes suppresses the spurious modes which are severely damped.
Maxwell’s Equations XLVI

... but for problems with strong singularities, it may be worth using the modified (energy conserving) scheme

\[
\frac{d}{dt} (E_N, \phi)_D^k = \left( \tilde{\nabla}_N \times H_N, \phi \right)_D^k + \sqrt{\tau_1} (n \times P, \phi) \partial_D^k ,
\]

\[
\frac{d}{dt} (P, \phi) \partial_D^k = \frac{\sqrt{\tau_1}}{2} (n \times [E_N], \phi) \partial_D^k ,
\]

\[
\frac{d}{dt} (H_N, \phi)_D^k = - \left( \tilde{\nabla}_D \times E_N, \phi \right)_D^k + \sqrt{\tau_2} (n \times Q, \phi) \partial_D^k ,
\]

\[
\frac{d}{dt} (\phi, Q) \partial_D^k = \frac{\sqrt{\tau_2}}{2} (n \times [H_N], \phi) \partial_D^k .
\]

Only one \( \tau \neq 0 \) is needed.
We consider the following time-domain eigenvalue solver

- A $[-1, 1]^3$ PEC cube
- 286 tetrahedra with 4th order
- An initial sharp off center pulse
- Fourier transform of timetrace to extract eigenfrequencies
- Exact solutions are

$$\omega = \pi \sqrt{m^2 + n^2 + l^2}, \quad m + n + l > 1.$$  

- $\tau_1 = 20$, $\tau_2 = 0$ in stabilized scheme.
Maxwell’s Equations XLVIII

Powerspectrum of $E_z$

$\omega$

$0 \quad 5 \quad 10 \quad 15 \quad 20$

$10^{-1}$

$10^{-2}$

$10^{-3}$

$10^{-4}$

$10^{-5}$

$10^{-6}$

$10^{-7}$

$10^{-8}$

$10^{-9}$

$0 \quad 5 \quad 10 \quad 15 \quad 20$

$10^{-1}$

$10^{-2}$

$10^{-3}$

$10^{-4}$

$10^{-5}$

$10^{-6}$

$10^{-7}$

$10^{-8}$

$10^{-9}$

Ecole de Ondes, INRIA 2006 – p.68
Summary on Maxwell’s Equations

The story is pretty complete

- Well understood theoretically
- Some issues with spurious modes – see talk by Perugia
- Extensively tested for large scale problems
- Bottleneck in time-domain is timestepping
- Some experience also in frequency-domain
- Open problem here is efficient preconditioning (multilevel)
- Slowly making impact in industry
DG-FEM is now widely used for wave problems in 2D/3D

- Acoustics
- Elastics
- Water/surface waves
- Plasma waves
- Gravitational waves
- etc

It has matured to become a tool!
DG-FEM is a good/mature candidate to be the underlying computational approach for a more general discretization tool

- There is a significant amount of analysis available
- Nothing structural which prohibits hp-adaptivity, non-conformity, curvilinear elements etc
- Overhead compared to ’optimal’ methods not dramatic
- Robust and flexible

How do we harvest this?
This is an attempt to provide a tool which can be helpful in developing DG-FEM software

The basic philosophy is to provide

- Discrete forms of vector operators with user specified fluxes
- A simple, Matlab-style language to build operators
- Tight links to solvers
- Fast/robust enough to be applicable to a range of problems with limited overhead.
The software is C++ based
It is based on nodal simplices (2D/3D)
It is general in order and element size
It allows for conforming and non-conforming discretization
It supports high-order surface operators
There is some level of parallel support
The tool is closely linked to Trilinos (SNL)
There is a Matlab toolbox (on disc) in which one can build prototypes and explore algorithms
Current use of Sledge++ include

- Adaptivity for stationary problems (L. Wilcoxon, UT Austin)
- MHD plasma dynamics (L Wilcos, UT Austin)
- Reduced basis methods for Maxwell’s equations (J. Rodriguez, ENSTA/CNRS).
- Kinetic plasma physics, PIC (G. Jacobs, SDSU)
- Free surface flows (A. Engsig-Karup, DTU)
- … and a few others

Interested ? – see me about access
We consider the adaptive solution of

$$\mathcal{L}u = f, \quad u(\partial\Omega) = 0.$$

We consider *goal oriented adaptation*, i.e., optimize grid to obtain

$$J(u) = \int_{\Omega} g \cdot u \, dx + \int_{\partial\Omega} h \cdot C u \, ds.$$

- One is rarely interested in the full solution everywhere
- .. and even if, $g$ can be chosen to mimic this
Primal problem

\[
J(u) = \int_{\Omega} gu \, dx + \int_{\partial\Omega} h \frac{\partial u}{\partial n} \, ds
\]

\[
Lu = f \quad \text{in } \Omega
\]

\[
Bu = e \quad \text{on } \partial\Omega
\]

Dual problem

\[
l(v) = \int_{\Omega} vf \, dx + \int_{\partial\Omega} e \frac{\partial v}{\partial n} \, ds
\]

\[
L^*v = g \quad \text{in } \Omega
\]

\[
B^*v = h \quad \text{on } \partial\Omega
\]
We consider the simple example

- Poissons equation: \( \nabla^2 u = f \).
- Problem is self-adjoint – we must solve same problem twice.
- \( \Omega \) is L-shaped
- \( f \) chosen such that
  \[
  u(r, \theta) = r^{2/3} \sin(2\pi/3\theta) ,
  \]
- \( g = u \) - mimics mean error
- Fully non-conforming \( hp \)-adaptivity is allowed.
| $|J(u)-J(u_h)|$ | DOF $^{1/2}$ |
|----------------|-------------|
| $10^{-7}$      | $10^{-5}$   |
| $10^{-6}$      | $10^{-5}$   |
| $10^{-5}$      | $10^{-5}$   |
| $10^{-4}$      | $10^{-5}$   |

- $hp$ TOL=$10^{-7}$
- $hp$ TOL=$10^{-7}$ coarfrac 0.1
- $hp$ p=1 TOL=$10^{-5}$
- $hp$ p=2 TOL=$10^{-5}$
- $hp$ p=3 TOL=$10^{-5}$
- $hp$ refper 20
- $hp$ refper 20 coarper 10

Ecole de Ondes, INRIA 2006 – p.79
Summary of Lecture 3

- The extension to 2D/3D is relatively straightforward
- Significant flexibility in element/flux choices
- Most analysis carries over with no/minor changes
- Thoroughly developed for Maxwell’s equations
- Beware of spurious modes
- Mature enough to attempt 'tool' development

So far we have focused on linear problems – what about variable coefficients and nonlinear problems – Lecture 4!