MA557/MA578/CS557
Lecture 26

Spring 2003
Prof. Tim Warburton

timwar@math.unm.edu
Note on Homework 8

When you enter $h: 0.10$ you are actually specifying Max triangle size = 0.10

So for the order of accuracy in this case you would take $h=\sqrt{0.1}$
Results

• I ran the advection code in a square domain:

[-4,4]x[-4,4]

• With hash grids like – here $h=1$

• Solution used:

$$dt = \frac{.125}{(umNq)(umNq+1)\max(Fconst(:))}$$

• $5^{th}$ order Rk scheme used
### Error in L2 norm

<table>
<thead>
<tr>
<th>p \ h</th>
<th>2</th>
<th>1</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.938774444976985</td>
<td>0.38815661859066</td>
<td>0.10197215280871</td>
</tr>
<tr>
<td>2</td>
<td>0.30215330223980</td>
<td>0.15653021172476</td>
<td>0.02800152435822</td>
</tr>
<tr>
<td>3</td>
<td>0.22043684447109</td>
<td>0.07272803132766</td>
<td>0.00626031860408</td>
</tr>
<tr>
<td>4</td>
<td>0.11652413181376</td>
<td>0.02876534888079</td>
<td>0.00131839250177</td>
</tr>
<tr>
<td>5</td>
<td>0.11289073566956</td>
<td>0.01460227046098</td>
<td>0.00033161249084</td>
</tr>
<tr>
<td>6</td>
<td>0.06133875385613</td>
<td>0.00423786600885</td>
<td>0.00003293355651</td>
</tr>
<tr>
<td>7</td>
<td>0.04835896860374</td>
<td>0.00184010232130</td>
<td>0.00001364982037</td>
</tr>
<tr>
<td>8</td>
<td>0.03584481563628</td>
<td>0.00037174570177</td>
<td>0.00000091542340</td>
</tr>
</tbody>
</table>

- Note from h=2 to h=1 the convergence is slow (pre-exponential).
- From h=1 to h=0.5 convergence rate is pretty good.
- We estimate the error as: \( \text{Err} = C h^s \)
P=8, h1=1, h2=0.5

• Set:

\[ \text{err}(h_1) = Ch_1^s \]
\[ \text{err}(h_2) = Ch_2^s \]

\[ \log\left( \frac{\text{err}(h_1)}{\text{err}(h_2)} \right) \]
\[ \Rightarrow s = \frac{\log\left( \frac{h_1}{h_2} \right)}{\log\left( \frac{1}{0.5} \right)} \]

• Then

\[ s = \frac{\log\left( \frac{0.00037174570177}{0.00000091542340} \right)}{\log\left( \frac{1}{0.5} \right)} \approx 8.67 \]

• So we get approximately 8.5 order accuracy
Just Using the $h=1, h=0.5$ Errors

<table>
<thead>
<tr>
<th>$P$</th>
<th>approx. order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.93</td>
</tr>
<tr>
<td>2</td>
<td>2.48</td>
</tr>
<tr>
<td>3</td>
<td>3.54</td>
</tr>
<tr>
<td>4</td>
<td>4.45</td>
</tr>
<tr>
<td>5</td>
<td>5.46</td>
</tr>
<tr>
<td>6</td>
<td>7.01</td>
</tr>
<tr>
<td>7</td>
<td>7.07</td>
</tr>
<tr>
<td>8</td>
<td>8.67</td>
</tr>
</tbody>
</table>

Experimentally (apart from $p=6$) we find that the rate seems to be better than expected (we estimated a rate of $\sigma^{-1}=p$ since the solution is regular)
A general linear pde system will look like:

\[ \begin{align*}
\frac{\partial q}{\partial t} + A \frac{\partial q}{\partial x} + B \frac{\partial q}{\partial y} &= 0 \\
\left[0, t \right] \rightarrow \mathbb{R}^d \\
\end{align*} \]

where \( d = \dim(q) \) such that:

\[ \Omega \times \mathbb{R}^d \]
Lax-Friedrichs Fluxes

Suppose we know the maximum wave speed of the hyperbolic system:

\[ \frac{\partial q}{\partial t} + A \frac{\partial q}{\partial x} + B \frac{\partial q}{\partial y} = 0 \]

As long as the matrix \( A, B \in \mathbb{R}^n \times \mathbb{R}^n \) are codiagonalizable for any linear combination.

i.e. \( C = \alpha A + \beta B \) has real eigenvalues for any real \( \alpha, \beta \).

And:

\[ \max_{x, \alpha, \beta} \lambda = \lambda \]

then we can use the following DG scheme:
Lax-Friedrichs DG Scheme

For Linear Systems

Find \( q : P^p(T) \times [0, t] \rightarrow \mathbb{R}^d \) such that \( \forall \phi \in P^p(T) \)

\[
\begin{pmatrix}
\phi, \\
\frac{\partial}{\partial t}
\end{pmatrix}
\begin{pmatrix}
\frac{\partial q}{\partial x} + A \frac{\partial q}{\partial x} + B \frac{\partial q}{\partial y} \\
\frac{\partial q}{\partial y}
\end{pmatrix}
+ \begin{pmatrix}
\phi, \\
\frac{\partial}{\partial t}
\end{pmatrix}
\begin{pmatrix}
(An_x + Bn_y) \frac{[q]}{2} \\
\frac{[q]}{2}
\end{pmatrix}
= 0
\]

where \( [q] = q^+ - q^- \).
Discrete LF/DG Scheme

- We choose to discretize the \( P_p \) polynomial space on each triangle using \( M = \frac{(p+1)(p+2)}{2} \) Lagrange interpolatory polynomials.

- i.e.

\[
\mathcal{P}_p(T) = \{ h_m \}_{1 \leq m \leq M}
\]

- The scheme now reads:

\[
\begin{align*}
\sum_{m=1}^{M} (h_n, h_m)_T + \sum_{m=1}^{M} \left( \frac{\partial h_m}{\partial x} \right)_T \left( \frac{\partial h_m}{\partial y} \right)_T \\
\sum_{m=1}^{M} A(h_n, h_m)_T + B(h_n, h_m)_T q_m \\
\end{align*}
\]

Find \( \{ q_m \}(t), m = 1, \ldots, M \) such that:

\[
\begin{bmatrix}
\mathcal{A} + \mathcal{B} h_n \mathcal{I} + \mathcal{D}_T \frac{\partial}{\partial t} - \mathcal{L} I \\
\end{bmatrix} q_m = 0
\]

\[
\begin{bmatrix}
\mathcal{A} + \mathcal{B} h_n \mathcal{I} + \mathcal{D}_T \frac{\partial}{\partial t} - \mathcal{L} I \\
\end{bmatrix} q_m = 0
\]
Tensor Form

Find \( \{q_{mcj}(t)\}, \ m = 1, \ldots, M, \ c = 1, \ldots, \dim(q), \ j = 1, \ldots, \#\text{tri}\} \) such that:

\[
\sum_{m=1}^{M} \left\{ \frac{d}{dt} \sum_{d=1}^{\dim(q)} \sum_{d'=1}^{\dim(q)} \partial_{\lambda} \left( h_{n}^{m} \right) \right\} + \ldots \\
+ \sum_{m=1}^{M} \left\{ \partial_{\lambda} \left( h_{n}^{m} \right) \right\} \sum_{m=1}^{M} \left\{ \frac{d}{dt} \sum_{d=1}^{\dim(q)} \sum_{d'=1}^{\dim(q)} \partial_{\lambda} \left( h_{n}^{m} \right) \right\} + \ldots \\
= 0
\]
Simplified Tensor Form

Find \( \{q_{mcj}(t), \ m=1,...,M, \ c=1,...,\dim(q), \ j=1,...,\#tri \} \) such that:

\[
\frac{d}{dt} q_{ncj} + \ldots
\]

\[
\sum_{d=1}^{\dim(q)} A_{cd} \left( \sum_{m=1}^{M} D_{nm}^x q_{mdj} \right) + \sum_{d=1}^{\dim(q)} B_{cd} \left( \sum_{m=1}^{M} D_{nm}^y q_{mdj} \right) + \ldots
\]

\[
\sum_{d=1}^{\dim(q)} \sum_{e=1}^{3} \left( \frac{\left( A_{cd} n_e^x + B_{cd} n_e^y \right) - \tilde{\lambda}I}{2} \right) \left( \sum_{m=1}^{M} S_{nm}^e \left[ q_{mdj} \right] \right) = 0
\]

where:

\( \mathbf{M}_{nm} = (h_n, h_m)_T \)

\( \tilde{\mathbf{D}}_{nm}^x = \left( h_n, \frac{\partial h_m}{\partial x} \right)_T \), \( \mathbf{D}_{nm}^x = \sum_{k=1}^{k=M} \left( \mathbf{M}^{-1} \right)_{nk} \tilde{\mathbf{D}}_{km}^x \)

\( \tilde{\mathbf{D}}_{nm}^y = \left( h_n, \frac{\partial h_m}{\partial y} \right)_T \), \( \mathbf{D}_{nm}^y = \sum_{k=1}^{k=M} \left( \mathbf{M}^{-1} \right)_{nk} \tilde{\mathbf{D}}_{km}^y \)

\( \tilde{\mathbf{S}}_{nm}^e = (h_n, h_m)_{\partial T_e} \), \( \mathbf{S}_{nm}^e = \sum_{k=1}^{k=M} \left( \mathbf{M}^{-1} \right)_{nk} \tilde{\mathbf{S}}_{km}^e \)
Geometric Factors

- We use the chain rule to compute the $D_x$ and $D_y$ matrices:

\[
D^x = \frac{\partial r}{\partial x} D^r + \frac{\partial s}{\partial x} D^s
\]

\[
D^y = \frac{\partial r}{\partial y} D^r + \frac{\partial s}{\partial y} D^s
\]

- In the umSCALAR2d scripts the

\[
\text{umDr} = D^r
\]

\[
\text{umDs} = D^s
\]

\[
rx = \frac{\partial r}{\partial x}, sx = \frac{\partial s}{\partial x}, ry = \frac{\partial r}{\partial y}, sy = \frac{\partial s}{\partial y}
\]
Surface Terms

• In the Matlab code first we extract the nodes on the edges of the elements:

• \( fC = \text{umFtoN}^*\text{C} \)

• The resulting matrix \( fC \) has dimension

\[
(\text{umNfaces}^*(\text{umP}+1))\times\text{umNel}
\]

• This lists all nodes from element 1, edge 1 then element 1, edge 2 then…

• In order to multiply, say \( fC \), by \( S^e \) we use \( \text{umNtoF}^*(\text{Fscale}.*fC) \) where:

\[
\text{Fscale} = \text{sjac.}./(\text{umNtoF}\times\text{jac});
\]
Example Matlab Code
(Upwind DG Advection Scalar Eqn)

• This is not strictly a global Lax-Friedrichs scheme since the lambda is chosen locally!!!
% inner multi-stage Runge-Kutta loop
sig = C;
for s=4:-1:1

% extract fields at face nodes (note NtoF is a sparse matrix)
fsig = umNtoF*sig;
% form centered differences at faces
dsigs = fsig(mapR)-fsig(mapL);
% set external boundary inflow state to zero
dsig(mapB) = zeros(Nboundary, 1)-fsig(mapB);
% compute surface flux
sflux = umFtoN*(Fconst.*dsigs);

% compute derivatives
dsigsdr = umDr*sig;
dsigds = umDs*sig;
dsigdx = rx.*dsigsdr + sx.*dsigds;
dsigy = ry.*dsigsdr + sy.*dsigds;

% evaluate residual (ax*Dx + ay*Dy + surface integral)
sig = C + (dt/s)*(ax*dssigdx + ay*dssigdy+sflux);

end;

% finalize Runge-Kutta
C = sig;
Specific Example

- We will try out this scheme on the 2D, transverse mode, Maxwell’s equations.
Maxwell's Equations (TM mode)

In the absence of sources, Maxwell's equations are:

\[ \frac{\partial (\mu H_x)}{\partial t} + \frac{\partial E_z}{\partial y} + \frac{\partial E_z}{\partial x} = 0 \]

\[ \frac{\partial (\mu H_y)}{\partial t} - \frac{\partial E_z}{\partial x} = 0 \]

\[ \frac{\partial (\varepsilon E_x)}{\partial t} + \frac{\partial H_y}{\partial y} - \frac{\partial H_y}{\partial x} = 0 \]

\[ \frac{\partial (\mu E_z)}{\partial t} + \frac{\partial H_x}{\partial y} + \frac{\partial H_x}{\partial x} = 0 \]

Where

\( \varepsilon = \text{permittivity} \)

\( \mu = \text{permeability} \)

\( H = \text{magnetic field} \)

\( E = \text{electric field} \)
Free Space..

• For simplicity we will assume that the permeability and permittivity are =1 we then obtain:

\[
\begin{align*}
\frac{\partial H_x}{\partial t} + \frac{\partial E_z}{\partial y} &= 0 \\
\frac{\partial H_y}{\partial t} - \frac{\partial E_z}{\partial x} &= 0 \\
\frac{\partial E_z}{\partial t} + \frac{\partial H_x}{\partial y} - \frac{\partial H_y}{\partial x} &= 0 \\
\frac{\partial H_x}{\partial x} + \frac{\partial H_y}{\partial y} &= 0
\end{align*}
\]
Divergence Condition

• We next notice that by taking the x derivative of the first equation and the y derivative of the second equation we find that:

\[
\frac{\partial}{\partial t} \left( \frac{\partial H_x}{\partial x} + \frac{\partial H_y}{\partial y} \right) = 0
\]

• i.e. the fourth equation (divergence condition) is a natural result of the equations – assuming that the initial condition for the magnetic field \( \mathbf{H} \) is divergence free.
The Maxwell’s Equations We Will Use

\[
\frac{\partial H_x}{\partial t} + \frac{\partial E_z}{\partial y} = 0
\]

\[
\frac{\partial H_y}{\partial t} - \frac{\partial E_z}{\partial x} = 0
\]

\[
\frac{\partial E_z}{\partial t} + \frac{\partial H_x}{\partial y} - \frac{\partial H_y}{\partial x} = 0
\]

\[
\frac{\partial H_x}{\partial x}(x, y, t = 0) + \frac{\partial H_y}{\partial y}(x, y, t = 0) = 0
\]
The Maxwell’s Equations As Conservation Rule

\[
\begin{align*}
\frac{\partial H_x}{\partial t} + \frac{\partial}{\partial x} (0) + \frac{\partial}{\partial y} (E_z) &= 0 \\
\frac{\partial H_y}{\partial t} + \frac{\partial}{\partial x} (-E_z) + \frac{\partial}{\partial y} (0) &= 0 \\
\frac{\partial E_z}{\partial t} + \frac{\partial}{\partial x} (-H_y) + \frac{\partial}{\partial y} (H_x) &= 0 \\
\frac{\partial H_x}{\partial x} (x, y, t = 0) + \frac{\partial H_y}{\partial y} (x, y, t = 0) &= 0
\end{align*}
\]
TM Maxwell’s Boundary Condition

• We will use the following PEC boundary condition – which corresponds to the boundary being a perfectly, electrical conducting material:

\[
H_x^+ = H_x^- \\
H_y^+ = H_y^- \\
E_z^+ = -E_z^-
\]
The Maxwell’s Equations In Matrix Form

\[
\frac{\partial \mathbf{q}}{\partial t} + \mathbf{A} \frac{\partial \mathbf{q}}{\partial x} + \mathbf{B} \frac{\partial \mathbf{q}}{\partial y} = 0
\]

\[
\mathbf{q} = \begin{pmatrix} H_x \\ H_y \\ E_z \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}
\]

- We are now assuming that the magnetic field is going to be divergence free – so we just forget about it for the moment.
Given $I = \frac{g}{\varepsilon^2} + \frac{\gamma}{\varepsilon}$, let $\gamma \geq |\gamma|

So the matrices are co-diagonalizable for all real alpha, beta and

\[
\frac{g}{\varepsilon^2} + \frac{\gamma}{\varepsilon} = 0, \quad \gamma \leq
\]

\[
(g\gamma)g + (\gamma \gamma - \gamma)\gamma = \begin{vmatrix} I \gamma - C \end{vmatrix} = 0 \iff
\]

\[
\begin{pmatrix}
\gamma & \gamma - \gamma & \gamma - \\
\gamma - \gamma & \gamma - \gamma & \gamma - \\
g & 0 & \gamma - \\
\end{pmatrix}
= \begin{vmatrix} I \gamma - C \end{vmatrix} = 0
\]

\[
C \text{ has eigenvalues which satisfy:}
\]

\[
\begin{pmatrix}
0 & \gamma - \\
\gamma - & 0 \\
g & 0 \\
\end{pmatrix}
= C \iff
\begin{pmatrix}
0 & 0 & 1 \\
0 & 0 & 0 \\
1 & 0 & 0 \\
\end{pmatrix}
= B, \quad
\begin{pmatrix}
0 & I & - 0 \\
I & 0 & 0 \\
0 & 0 & 0 \\
\end{pmatrix}
= A
\]

Then:

Eigenvalues
Project Time

- Due 04/07/03

- Code up a 2D DG solver for an arbitrary hyperbolic system based on the 2D DG advection code provided.

- Test it on TM Maxwell’s
  - Set A and B as described, set \( \text{lamdatilde} = 1 \) and test on some domain of your choosing.
  - Use PEC boundary conditions all round.

- Test it on a second set of named equations of your own choosing (determine these equations by research)

- Compute convergence rates. For smooth solutions.
Stability

• If A, B are both symmetric then it is straightforward to prove stability
  1) Set $\phi=q$
  2) Integrate by parts
  3) Sum over all elements
  4) Simplify
  5) Repeat the approaches previously used..