Feedback on Problem Set 5

Ph.D. Course:
Nodal DG-FEM for solving partial differential equations

Allan P. Engsig-Karup
Scientific Computing Section
DTU Informatics
Technical University of Denmark

August 15, 2012

Feedback On Problem Set 5

Linear Advection equation in 2D

$$u_t + \mathbf{c} \cdot \nabla u = 0, \quad \mathbf{x} \in \Omega([-1,1]^2)$$

By the energy method

$$\frac{d}{dt}||u||^2 + c_x \int_\Omega (u^2)_x d\mathbf{x} + c_y \int_\Omega (u^2)_y d\mathbf{x} = 0$$

$$\Leftrightarrow \quad \frac{d}{dt}||u||^2 = -c_x \oint_{\partial\Omega} n_x u^2 d\mathbf{x} - c_y \oint_{\partial\Omega} n_y u^2 d\mathbf{x}$$

Clearly, boundary conditions must be specified where

$$\mathbf{c} \cdot \hat{n} \leq 0$$

which corresponds to incoming characteristics.

If square domain is periodic in each Cartesian direction the energy must be conserved.

Feedback on Problem Set 5

$$\frac{du_h^k}{dt} = -(c_x \mathcal{D}_x^k + c_y \mathcal{D}_y^k) u_h^k + (\mathcal{M}^k)^{-1} \oint_{\partial D^k} \hat{n} \cdot (f^k - f^*) h(\mathbf{x}) d\mathbf{x}^k$$

$$f^*(u^-, u^+) = \tfrac{1}{2}\left(u^- + u^+\right)$$

```
function [rhsu] = AdvecRHS2Dcentral(u, timelocal, cx, cy, alpha)
% function [rhsu] = AdvecRHS2D(u, timelocal, a, alpha)
% Purpose  : Evaluate RHS flux in 2D advection equation
Globals2D;

du = zeros(Nfp*Nfaces,K);
du(:) = u(vmapM) - u(vmapP);

% Impose boundary conditions at inflow
Lx = 2; Ly = 2;
uexact = sin(2*pi/Lx*(x-cx*timelocal)).*sin(2*pi/Ly*(y-cy*timelocal));
uin = uexact(vmapD);
du(mapD) = u(vmapD) - uin;
du(mapN) = 0;

df = zeros(Nfp*Nfaces,K);
df(:) = (nx(:)*cx+ny(:)*cy).*du(:)*0.5;

% local derivatives of fields
[ux,uy] = Grad2D(u);

% compute right hand sides of the PDE's
rhsu  = -(cx*ux + cy*uy) + LIFT*(Fscale.*df);
return
```


Feedback on Problem Set 5

$$\frac{du_h^k}{dt} = -(c_x \mathcal{D}_x^k + c_y \mathcal{D}_y^k) u_h^k + (\mathcal{M}^k)^{-1} \oint_{\partial D^k} \hat{n} \cdot (f^k - f^*) h(\mathbf{x}) d\mathbf{x}^k$$

$$f^*(u^-, u^+) = \kappa \mathbf{c} u^- + (1-\kappa)\mathbf{c} u^+, \quad \kappa = \tfrac{1}{2}\left(1 + \frac{\hat{\mathbf{n}} \cdot \mathbf{c}}{|\hat{\mathbf{n}} \cdot \mathbf{c}|}\right)$$

```
function [rhsu] = AdvecRHS2Dupwind(u, timelocal, cx, cy)
% Purpose  : Evaluate RHS flux in 2D advection equation by upwinding
Globals2D;
% Define flux differences at faces
df = zeros(Nfp*Nfaces,K);
cn = cx*nx(:) + cy*ny(:); % normalized phase speed in normal direction
kappa = 0.5*(1+cn./abs(cn)); % if kappa=1 then outflow => pick uM, if kappa=0 then inflow  => pick uP
ustar = (kappa .* u(vmapM) + (1-kappa) .* u(vmapP));
df(:) = u(vmapM) - ustar;

% Impose boundary conditions at inflow
Lx = 2; Ly = 2; uexact = sin(2*pi/Lx*(Fx-cx*timelocal)).*sin(2*pi/Ly*(Fy-cy*timelocal));
uin = uexact(mapD);
uM  = u(vmapD);
df(mapD) = uM - (kappa(mapD) .* uM(:) + (1-kappa(mapD)) .* uin(:));
df(:) = cn.*df(:);

% compute right hand sides of the PDE's
[ux,uy] = Grad2D(u);
rhsu  = -(cx*ux + cy*uy) + LIFT*(Fscale.*df);
return
```

Feedback on Problem Set 5

```
function [rhsu] = RotatingHillRHS2Dupwindex5(u, timelocal, cx, cy, ufun)
% function [rhsu] = RotatingHillRHS2Dupwindex5(u, timelocal, a, alpha)
% Purpose  : Evaluate RHS flux in 2D advection equation with
%            varying coefficients using upwinding fluxes.
%
% By Allan P. Engsig-Karup, apek@imm.dtu.dk.
Globals2D;

% Define flux differences at faces
df = zeros(Nfp*Nfaces,K);

% phase speed in normal directions
cn = cx(vmapM).*nx(:) + cy(vmapM).*ny(:);

% if cn>=1 then outflow => pick uM,     if cn<1 then inflow   => pick uP
ustar = 0.5*(cn+abs(cn)).*u(vmapM) + 0.5*(cn-abs(cn)).*u(vmapP);

% Impose boundary conditions at inflow
uin = ufun(Fx,Fy,timelocal);
uM  = u(vmapM);
ustar(mapB) = 0.5*(cn(mapB)+abs(cn(mapB))).*uM(mapB) +...
              0.5*(cn(mapB)-abs(cn(mapB))).*uin(mapB);

df(:) = cn.*uM - ustar;

% local derivatives of fields
[ux,uy] = Grad2D(u);

% compute right hand sides of the PDE's,
% discrete approximation to varying coefficient terms
rhsu  = -(cx.*ux + cy.*uy) + LIFT*(Fscale.*df);
return
```