Problem Set 1

Ph.D. Course 2012:
Nodal DG-FEM for solving partial differential equations

If you have not already done so, please download all the Matlab codes from the book from

http://www.nudg.org/

and store and unpack them in a directory you can use with Matlab.

To familarize ourselves with the setup for problems with one spatial direction (1D), we consider the prototype model for hyperbolic PDEs: the linear advection equation

$$\partial_t u + a\partial_x u = 0 \tag{1}$$

This equation describes translation of some quantity $u(x,t)$ with constant advection speed a.

Exact solutions to this equation can be shown to be of the form

$$u(x,t) = f(x - at)$$

where $f(\cdot)$ is a function that defines the initial condition.

Open the three Matlab scripts Advec1DDriver.m, Advec1D.m and AdvecRHS1D.m in your favorite Matlab editor. These scripts can be found in Codes1D/ and they solve the PDE using a strong DG-FEM formulation with a Lax-Friedrichs-type flux. The solver is initially setup for a finite domain $x \in [0,2]$ with a boundary condition imposed at the left boundary $x = 0$ in AdvecRHS1D.m such that $u(0,t) = g(t)$ with $g(t)$ defined using the variable 'uin' in the script (line 14).

The geometry of the domain, number of domains K and size of domain, is controlled by the statement (in AdvecDriver1D.m, line 8)

```
[Nv, VX, K, EToV] = MeshGen1D(0.0, 2.0, 10);
```

where the first two arguments are the left and right boundary, respectively, and the last argument is the number of elements.

- Run Advec1DDriver.m and familiarize yourself with the different input parameters, e.g. number of elements (K), polynomial order (N), time step size (dt) (found in Advec1D.m, line 16) and initial condition $f(x)$. Change the code such that the numerical solution u_h can be shown on the fly for visual inspection.

  To run the code, make sure that /Codes1D and /ServiceRoutines are both in your path - use Matlab command addpath.

Scientific Computing Section, DTU Informatics, Kgs.-Lyngby, Denmark.

- Define a smooth initial function $f(\cdot)$ and describe how the numerical solution behaves compared to the exact solution in time? For example, do you observe losses in amplitude? Changes of initial solution profile? What is the numerical advection speed? Change the numerical flux type by changing α in AdvecRHS1D.m, line 9. Recall that $\alpha = 0$ is an upwind flux and $\alpha = 1$ is a central flux.

- Define a smooth initial function and compute the global L^2-error for different sets of parameters (N,K) at the final time chosen. What is the order of accuracy p in the global asymptotic error estimate $||u - u_h||_{\Omega,h} \leq Ch^p$ for the upwind ($\alpha = 0$) and central ($\alpha = 1$) schemes? Does p depend on the choice of flux? expansion order? To implement the global L^2-norm for the error you may use the code snippet using matrix-based integration below.

```
err = ua - u; % compute point-wise error
M = inv(V*V'); % mass matrix
errL2 = zeros(K,1);
for k = 1 : K
    errL2(k) = err(:,k)'*diag(J(:,k))*M*err(:,k);
end
errL2 = sqrt(sum(errL2)); % Global L^2-norm of error
```

- Change the time step in Advec1D.m manually and determine how the stable time step size scale with the element size h and the polynomial order N, i.e. What impact does this scaling have on the efficiency of the scheme?

- To familiarize yourself with the derivation of DG-FEM schemes, derive analytically a DG-FEM scheme for solving the linear advection equation and try and compare with the implementation in AdvecRHS1D.m. What changes would be needed to change the strong formulation to a weak formulation for solving the same problem?

- Consider what changes are needed in the Advec1Dxxx.m codes to solve the linear advection equation with variable phase speed $a = a(x)$ and test it by modifying the code.

- Define a nonsmooth initial function, e.g. a step function, and describe how the numerical solution behaves compared to the exact solution in time? (HINT: you will need to adjust the domain size, the initial function and the definition of the left boundary condition.)

If time permits it

- Show using an Energy Method that solutions to the linear advection equation conserves energy if u is assumed periodic.

- Using an energy technique, discuss how many boundary conditions are needed in a finite domain at each end.