Project 3: 1D Wave Equation

1 Continuous Problem

The governing partial differential equation (PDE) is

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} = 0 \tag{1}$$

where $f = qu$ and $u = 2$ is a constant. Thus, an initial wave $q(x, 0)$ will take exactly $t = 1$ time in order to complete one full revolution (loop) of the domain.

1.1 Initial Condition

Since the governing PDE is a hyperbolic system, then this problem represents an initial value problem (IVP or Cauchy Problem). We, therefore, need an initial condition. Let it be the following Gaussian

$$q(x, 0) = e^{\frac{x}{2\sigma}^{2}} \tag{2}$$

where $\sigma = \frac{1}{8}$ and $x \in [-1, 1]$

1.2 Boundary Condition

This problem also requires a boundary condition: let us impose periodic boundary conditions, meaning that the domain at $x = +1$ should wrap around and back to $x = 1$. Your solution variable q should have the same solution at $x = 1$ and $x = +1$.

2 Simulations

Write a code (or two) that uses both CG and DG. I strongly recommend that you code the CG version first. It is better to use the same code to do both CG and DG with a switch (if statement) to handle the communicator in both CG and DG. You need to show results for exact (let Q=N+1 be exact) AND inexact integration (Q=N) so write your codes in a general way.

2.1 Results to be shown

You must show results for linear elements N = 1 with increasing number of elements N_e and then show results for N = 4,N = 8,and N = 16 with increasing numbers of elements.

N=1 Simulations For linear elements, use N_e = 16, 32 and 64 elements. Plot the normalized L2 error norm versus N_P (given below) for these 3 simulations on one plot.

N=4 Simulations For N = 4 use Ne = 4, 8 and 16 elements and plot the norms as above.

N=8 Simulations For N = 8 use Ne = 2,4 and 8 elements and plot the norms as above.

N=16 Simulations For N = 16 use Ne = 1, 2 and 4 elements and plot the norms as above.

3 Helpful Relations

Error Norm the normalized L_2 error norm that you should use is:

$$||error|| = \sqrt{\frac{\sum_{k=1}^{N_P}(q^{numerical}(x_k) - q^{exact}(x_k))^2}{\sum_{k=1}^{N_P} q^{exact}(x_k)^2}} \tag{3}$$

where k = 1, ..., N_P are $N_p = N_e N + 1$ global gridpoints and $q^{numerical}$ and q^{exact} are the numerical and exact solutions after one full revolution of the wave. Note that the wave should just stop where it began without changing shape (in a perfect world). Your solution will do that for lots of gridpoints (high resolution). At low resolution, you will see much error. **Time Integrator** To solve the time-dependent portion of the problem use the 2nd order RK method: for $\frac{\partial q}{\partial t} = R(q)$ let

$$q^{n+\frac{1}{2}} = q^n + \frac{\Delta t}{2} R(q^n) \tag{4}$$

$$q^{n+1} = q^n + \frac{\Delta t}{2} R(q^{n+\frac{1}{2}}) \tag{5}$$

or a better time-integrator of your choice (DO NOT USE FORWARD EULER); feel free to use ODE45 in Matlab if you know how to use it. Make sure that your time-step Δt is small enough to ensure stability.