Exercise set 10: Discontinuous Galerkin

Numerical Methods for Hyperbolic Partial Differential Equations
IMATH, FS-2020
Lecturer: Dr. Philipp Öffner
Teaching Assistant: Davide Torlo

Problem 10.1 Discontinouus Galerkin (DG) (8pts)

Consider the conservation law

$$\partial_t u(x,t) + \partial_x f(u(x,t)) = 0, \quad x \in \Omega = [a,b],\, t \in \mathbb{R}^+,\, f \in \mathcal{C}^2(\mathbb{R}). \tag{1}$$

Consider a high order DG approximation. Subdivide Ω in N cells $K_j = [x_{j-1/2}, x_{j+1/2}]$, where $\{x_{j-1/2}\}_{j=1}^{N+1}$ are equispaced points and $x_{1/2} = a$ and $x_{N+1/2} = b$. Consider as finite dimensional functional space

$$V := \{v \in \mathbb{L}^2(\Omega) : v|_{K_j} \in \mathbb{P}^p(K_j)\ \forall j = 1, \dots, N\},$$

where $p \in \mathbb{N}$ is the maximum degree of the polynomial considered.

1. Write the weak formulation of the equation for an element K_j.
2. Let $\{\tilde{\varphi}^i(y)\}_{i=0}^p$ be a basis for $\mathbb{P}^p([-1,1])$ and, through an affine transformation, let $\{\varphi^i_{K_j}(x)\}_{i=0}^p$ be the corresponding basis for $\mathbb{P}^p(K_j)$, hence

   $$\text{span}\langle\{\varphi^i_{K_j}|_{K_j}(x), j = 1, \dots, N,\, i = 0, \dots, p\}\rangle = V.$$

   Consider the approximation solution on the cell K_j as

   $$u|_{K_j}(x,t) = u_j(x,t) = \sum_{i=0}^{p} u_{j,i}(t)\varphi^i_{K_j}(x). \tag{2}$$

   Here, with an abuse of notation we use u as the approximation solution, instead of the exact one. From now on, u will be the approximation solution. Show that the semidiscrete formulation of DG can be written as

   $$\int_{K_j} \varphi^i_{K_j}(x)\, \partial_t u(x,t) dx - \int_{K_j} f(u(x,t)) \partial_x \varphi^i_{K_j}(x)\, dx + \Big[f(u(x,t))\varphi^i_{K_j}(x) \Big]_{x_{j-1/2}}^{x_{j+1/2}} = 0, \quad j = 1, \dots, N \tag{3}$$

   and write and expression for $\Big[f(u(x,t))\varphi^i_{K_j}(x) \Big]_{x_{j-1/2}}^{x_{j+1/2}}$. Are the solution values at the interface well defined? How can one define these terms?
3. Using explicit Euler time discretization the (3) can be rewritten as a linear system for every cell K_j

   $$\sum_{i_2=0}^{p} M_{i_1,i_2} \frac{u^{n+1}_{j,i_2} - u^n_{j,i_2}}{\Delta t} = -r_{i_1}, \quad \text{for } i_1 = 0, \dots, p. \tag{4}$$

   Write explicitly the definition of the mass matrix M and the right hand side r. What happens if the basis functions $\tilde{\varphi}^i$ are orthonormal?
4. Derive the classical FV scheme starting from the DG scheme.

5. Consider the linear advection equation, i.e. $f(u) = c\,u$, a central difference flux

$$\begin{aligned}\Big[f(u(x,t))\varphi^i_{K_j}(x)\Big]_{x_{j-1/2}}^{x_{j+1/2}} &= c\frac{u_j(x_{j+1/2}) + u_{j+1}(x_{j+1/2})}{2}\varphi^i_{K_j}(x_{j+1/2}) \\ &\quad - c\frac{u_{j-1}(x_{j-1/2}) + u_j(x_{j-1/2})}{2}\varphi^i_{K_j}(x_{j-1/2}),\end{aligned} \tag{5}$$

periodic boundary conditions ($x_{1/2} = x_{N+1/2}$) and the semidiscrete form (3). Prove that

$$\frac{1}{2}\frac{d}{dt}\int_\Omega u(x,t)^2 dx = 0. \tag{6}$$

Problem 10.2 Implementation of a DG scheme (12 pts)

In this exercise you will implement a 1-dimensional discontinuous Galerkin code for the conservation law

$$\partial_t u + \partial_x f(u) = 0$$

defined in some domain $\Omega = [a, b]$ and periodic boundary conditions. Consider a high order DG approximation given by (3) and (4).

Remember that in $\Big[f(u(x,t))\varphi^i_{K_j}(x)\Big]_{x_{j-1/2}}^{x_{j+1/2}}$ you will need to use a numerical flux to evaluate the interface values of $f(u)$, i.e.

$$f(u(x_{j+1/2},t))\varphi^i_{K_j}(x_{j+1/2}) := f^{\text{num}}(u_j(x_{j+1/2},t), u_{j+1}(x_{j+1/2},t))\varphi^i_{K_j}(x_{j+1/2}).$$

Consider nodal DG of degree $p = 2$ on Gauss Lobatto points, this means that the basis function considered are Lagrangian basis functions defined on $[-1, 1]$ as

$$\tilde{\varphi}^1 = \frac{x(x-1)}{2}, \tag{7}$$

$$\tilde{\varphi}^2 = (1+x)(1-x), \tag{8}$$

$$\tilde{\varphi}^3 = \frac{x(1+x)}{2}. \tag{9}$$

To compute the integrals (mass matrix and volume term) use a Gauss Lobatto quadrature rule on $[-1, 1]$

$$\int_{-1}^{1} g(x)dx \approx \sum_{q=1}^{3} w_q g(x_q), \quad x_q = [-1, 0, 1],\ w_q = [1/3, 4/3, 1/3]. \tag{10}$$

Hint:

- Which are the integrals that must be computed in (4)? How can they be transformed onto $[-1, 1]$? How does the quadrature rule scale for these integrals?

Add a 3rd order time integration method SSPRK3:

$$\begin{aligned}\mathbf{u}^{(1)} &= \mathbf{u}^n + \Delta t L(\mathbf{u}^n) \\ \mathbf{u}^{(2)} &= \frac{3}{4}\mathbf{u}^n + \frac{1}{4}\mathbf{u}^{(1)} + \frac{1}{4}\Delta t L(\mathbf{u}^{(1)}) \\ \mathbf{u}^{n+1} &= \frac{1}{3}\mathbf{u}^n + \frac{2}{3}\mathbf{u}^{(2)} + \frac{2}{3}\Delta t L(\mathbf{u}^{(2)}),\end{aligned}$$

where L is the evolution operator defined, using (4), as

$$L(\mathbf{u}) = -M^{-1}\mathbf{r}(\mathbf{u}). \tag{11}$$

Test your code for the linear advection equation $\partial_t u + \partial_x u = 0$ on two initial conditions:

1. For $x \in [-2, 2]$, $t \in [0, 1]$, with periodic boundary conditions and initial data given by:

$$u_0 = \cos(\pi x) \tag{12}$$

2. For $x \in [-2, 2]$, $t \in [0, 1]$, with periodic boundary conditions and initial data given by:

$$u_0 = \begin{cases} 1 & \text{if } -1 \leq x \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

For the first initial condition, compute the accuracy of the scheme.

What do you notice for the second initial conditions? Do you have ideas on how to fix it?

Remarks

The point of this exercise is to gain some experience in implementing an end-to-end numerical scheme. The important steps should be independently checked and debugged, in order to be sure that everything works. Feel free to follow the following suggested code structure (way more simplified than a pseudocode):

Data: initial condition, model information, solver information
Result: solution at T
$u^0_{i,j} \leftarrow$ initial condition;
while $t < T$ **do**
  $\mathbf{u}^{n+1}$=SSPRK3 ($\mathbf{u}^n$,Δt);
  $t = t + \Delta t$
end

Algorithm 1: main

Data: $u^n_{i,j}$ (all coefficients), Δt
Result: Using the method of lines, you can write the space and time update independently: $\partial_t \mathbf{u} = L(\mathbf{u})$. Below is the time-marching algorithm given by SSPRK3:

$\mathbf{u}^{(1)} = \mathbf{u}^n + \Delta t L(\mathbf{u}^n)$;
$\mathbf{u}^{(2)} = \frac{3}{4}\mathbf{u}^n + \frac{1}{4}\mathbf{u}^{(1)} + \frac{1}{4}\Delta t L(\mathbf{u}^{(1)})$;
$\mathbf{u}^{n+1} = \frac{1}{3}\mathbf{u}^n + \frac{2}{3}\mathbf{u}^{(2)} + \frac{2}{3}\Delta t L(\mathbf{u}^{(2)})$;

Algorithm 2: SSPRK3

Data: $u_{i,j}$ (modes)
Result: Computes the evolution operator $L(\mathbf{u})$

compute volume integral $\int_{K_j} f(u_j(x))\partial_x \varphi^i_{K_j} dx$ using the quadrature rule, $j = 1, \ldots, N$, $i = 0, \ldots, p$;
compute numerical fluxes at the interface of the cells $f^{\text{num}}_{j-1/2}$ for $j = 1, \ldots, N+1$;
compute surface integral $[f(u(x))\varphi^i_{K_j}]^{x_{j+1/2}}_{x_{j-1/2}}$ using the numerical fluxes ;
compute the final operator $-M^{-1}\mathbf{r}_{K_j}$ for every cell $j = 1, \ldots, N$;

Algorithm 3: compute the evolution operator L(u)

Hints

- Note that the mass matrix doesn't change in time and for different cells, so you can compute it just once, invert it just once, store it and use it every time you need it.
- Check by hand that your mass matrix computed with that quadrature rule is correct
- Implement a visualization function that plots the solution, plotting in each interval the corresponding solution. This can be used to debug to check whether your reconstruction makes sense or not.
- For the numerical flux use Rusanov
- Note that for $p = 0$ and $\tilde{\varphi}^1(y) = 1$ you should get the 1st order FVM. You can use this fact to debug your code!

Organiser: Davide Torlo, Office: home (davide.torlo@math.uzh.ch)

Published: May 14, 2020

Due date: May 28, 2020, h10.00 (use the upload tool of my.math.uzh.ch, see wiki.math.uzh.ch/public/student_upload_homework or if you have troubles send me an email).