

**Instructions**

Put away your cell-phone and read this document from start to finish. Discuss in the group what the different instructions and tasks mean. Make sure you have a plan how to achieve the main task (and write up the results!) within the allotted time (team-work is probably needed). How will you check that the subtasks are correct?

**Main tasks**

The main task of this homework is to approximate the transport equation

$$u_t + (au)_x = 0, \quad x_L \leq x \leq x_R, \quad t > 0, \quad (1)$$

with initial conditions  $u(x, 0) = f(x)$ .

**Subtasks**

At the end of his homework you will have an approximation on the grid from the previous homework but you will get there in steps:

1. Approximate  $(au)_x$  on a single element with no boundary conditions.
2. Approximate  $(au)_x$  on a single element with boundary conditions.
3. Approximate  $u_t + (au)_x = 0$  on a single element with periodic boundary conditions.
4. Approximate  $(au)_x$  on a grid with inter-element and physical boundary conditions.
5. Approximate  $u_t + (au)_x = 0$  on a grid with inter-element and physical boundary conditions (you just wrote your first discontinuous Galerkin solver).

**Approximate  $(au)_x$  on a single element with no boundary conditions**

Let  $u$  be a function approximated on  $x_L \leq x \leq x_R$ , by a polynomial  $u^h$

$$u(x(r)) \approx u^h \equiv \sum_{k=0}^q \hat{u}_k P_k(r). \quad (2)$$

First consider the case when  $a$  is a constant. Your task is to find another polynomial (more precisely the coefficients of that polynomial)

$$b(x) = \sum_{k=0}^q \hat{b}_k P_k(r),$$

such that the coefficients  $\hat{b}_k$  solve the linear system of equations

$$\int_{x_L}^{x_R} b P_l dx = \int_{x_L}^{x_R} P_l \frac{d(au^h)}{dx} dx, \quad l = 0, \dots, q. \quad (3)$$

To write the system in matrix form we introduce the  $(q+1) \times 1$  vectors  $\hat{\mathbf{b}} = [\hat{b}_0, \hat{b}_1, \dots, \hat{b}_q]^T$ ,  $\hat{\mathbf{u}} = [\hat{u}_0, \hat{u}_1, \dots, \hat{u}_q]^T$  along with the matrices  $M$  and  $S$ . We then have (recall  $a$  is a constant for now)

$$M\hat{\mathbf{b}} = aS\hat{\mathbf{u}}.$$

When computing the elements of  $M$  and  $S$  you should use a quadrature of sufficient degree so that the integrals are computed exactly.

Report the error (max or  $L_2$ ) in the approximation to the derivative as a function of  $q$  for a few different  $u$ . Look up exponential / spectral / algebraic convergence in Boyd's book and indicate which type of convergence you have here. Take the largest value of  $q$  large enough so that you see the effects of roundoff. Redo at least one of the error vs.  $q$  plots in quadruple precision.

### Approximate $(au)_x$ on a single element with boundary conditions

Now consider the case when  $a = a(x)$  is not a constant. As we have subroutines for computing derivatives of Legendre polynomials,  $P'_l$ , it is convenient to perform an integration by parts in (3) to find

$$\int_{x_L}^{x_R} b P_l dx = - \int_{x_L}^{x_R} au^h \frac{dP_l}{dx} dx + [au^h P_l]_{x_L}^{x_R}, \quad l = 0, \dots, q. \quad (4)$$

Note that if  $a(x)$  is a polynomial the integral can be carried out exactly as long as the degree of the quadrature is high enough, but in the general case the integral will be inexact.

Note that the term  $[au^h P_l]_{x_L}^{x_R}$  can be thought of as a sum of two vectors

$$a(x_R)u^h(x_R) \underbrace{\begin{bmatrix} P_0(1) \\ P_1(1) \\ \vdots \\ P_q(1) \end{bmatrix}}_{\mathbf{L}_R} - a(x_L)u^h(x_L) \underbrace{\begin{bmatrix} P_0(-1) \\ P_1(-1) \\ \vdots \\ P_q(-1) \end{bmatrix}}_{\mathbf{L}_L}.$$

Again, we may write (4) as a system of equations

$$M\hat{\mathbf{b}} = -\tilde{S}^a\hat{\mathbf{u}} + a(x_R)u^h(x_R)\mathbf{L}_R - a(x_L)u^h(x_L)\mathbf{L}_L,$$

and solve for  $\hat{\mathbf{b}}$ .

Consider the case  $-\pi \leq x \leq \pi$ , for  $a(x) = 2/(2 + \sin(x))$  and  $u(x) = \exp(\sin(x))$  and find and plot the errors in  $(au)_x$  as a function of  $q$ . How accurate do you have to compute the integrals to not lose any accuracy in the approximation of  $(au)_x$ ?

We will now compute the eigenvalues of the matrix  $A$ , defined through  $\hat{\mathbf{b}} = A\hat{\mathbf{u}}$ , and must therefore rewrite boundary terms slightly. Note that to compute  $u^h(x_R)$  you may simply take the inner product  $\mathbf{L}_R^T \hat{\mathbf{u}}$  and we thus have

$$A = M^{-1} \left( -\tilde{S}^1 + \mathbf{L}_R \mathbf{L}_R^T - \mathbf{L}_L \mathbf{L}_L^T \right). \quad (5)$$

- Assume periodic boundary conditions (i.e. use  $u^h(x_R) = \mathbf{L}_L^T \hat{\mathbf{u}}$ ) and use the LAPACK routine `dgeev` to compute the eigenvalues,  $\lambda_k$ , of the matrix

$$A = M^{-1} \left( -\tilde{S}^1 + \mathbf{L}_R \mathbf{L}_L^T - \mathbf{L}_L \mathbf{L}_R^T \right) \quad (6)$$

and plot all of them in the complex plane for a few values of  $q$ . Also plot  $\lambda_{\max} = \max_k |\lambda_k|$  as a function of  $q$ . How does  $\lambda_{\max}$  scale with  $q$ ?

- How does this compare with the “size” of  $d/dx$  when applied to  $u(x) = e^{iqx}$ ?
- Compare  $\lambda_{\max}$  with known formulas for  $P'_q(\pm 1)$  and  $P'_q(0)$ .
- Why must  $A$  always have an eigenvalue that is identically zero?

### Approximate $u_t + (au)_x = 0$ on a single element with periodic boundary conditions

For this task let  $a$  be a constant. To approximate the solution to  $u_t + (au)_x = 0$  we simply replace the role of  $b(x)$  with  $u_t^h$ . Assuming that  $u^h(x, t)$  depends on space and time we have

$$u_t(x, t) \approx \frac{\partial u^h(x(r), t)}{\partial t} \equiv \sum_{k=0}^q \hat{u}'_k(t) P_k(r). \quad (7)$$

You immediately obtain a system of ordinary differential equations for  $\hat{\mathbf{u}}'(t)$

$$M \hat{\mathbf{u}}'(t) = -\tilde{S}^a \hat{\mathbf{u}} + a(x_R) u^h(x_R) \mathbf{L}_R - a(x_L) u^h(x_L) \mathbf{L}_L. \quad (8)$$

To find the initial data use your codes from homework 3.

To timestep the system you can use RK4. If that is too slow due to excessively small timesteps (stability requires  $\lambda_{\max} \Delta t \lesssim 1$ ) you can try an implicit method like the trapezoidal method or if you want to be fancy, SDIRK-5-4 or RadauIIA.

Show me some movies of the results of this assignment. Also report (by a **semilogy** plot of error vs. time) how the  $L_2$  error grows in time for some combinations of timesteps and  $q$ . Can you detect the advertised rate of convergence for the time marching scheme you chose?

*Note that the description of this assignment is rather terse, ask for help if you need it.*

### Approximate $(au)_x$ on a grid with inter-element and physical boundary conditions

Now consider the grid from homework 3. That is, consider a grid with  $N + 1$  grid-points  $x_L = x_0 < x_1 < \dots < x_N = x_R$ , defining  $N$  elements  $\Omega_i = \{x \in [x_{i-1}, x_i]\}$ ,  $i = 1, \dots, N$ . You are to approximate a function  $b(x)$  composed of polynomials of degree  $q$  on each element. We denote the polynomials representing  $b(x)$  and  $u^h(x)$  on element  $i$  by  $b_i(x)$  and  $u_i^h(x)$ .

Equation (4) on two adjacent element are

$$\int_{x_i}^{x_{i+1}} b_{i+1} P_l dx = - \int_{x_i}^{x_{i+1}} au_{i+1}^h \frac{dP_l}{dx} dx + [au_{i+1}^h P_l]_{x_i}^{x_{i+1}}, \quad (9)$$

$$\int_{x_{i-1}}^{x_i} b_i P_l dx = - \int_{x_{i-1}}^{x_i} au_i^h \frac{dP_l}{dx} dx + [au_i^h P_l]_{x_{i-1}}^{x_i}. \quad (10)$$

Now, strictly speaking we do not need to connect the elements if we only want to approximate the derivative of  $u$ . However if we want to solve  $u_t + (au)_x = 0$  we will need for information to travel between elements.

In the above formulas the values  $u_{i+1}^h(x_i)$  and  $u_i^h(x_i)$  both exist at the boundary in-between the elements and we therefore use these to “connect” the elements. Precisely we replace the two quantities  $au_{i+1}^h(x_i)$  and  $au_i^h(x_i)$  **by a single value** called the numerical flux and denoted  $(au)^*$ . We will discuss how to chose  $(au)^*$  in detail later but for now we will simply take it as an average

$$(au)^* = \beta au_{i+1}^h(x_i) + (1 - \beta) au_i^h(x_i).$$

You have probably noticed that the mass matrix on the left hand sides in the above system of equations can be inverted locally so a good strategy for finding  $b(x)$  is

1. For each element compute the values of  $u_h$  on the edges (you probably want to add these “trace values” or “traces” to your type `quad_1d`).
2. For each element compute the numerical fluxes by averaging. Let  $\beta$  be a parameter.
3. For each element find  $\hat{\mathbf{b}}_i$  from

$$M\hat{\mathbf{b}}_i = -\tilde{S}^a \hat{\mathbf{u}}_i + (au)_R^* \mathbf{L}_R - (au)_L^* \mathbf{L}_L, \quad i = 1, \dots, N.$$

Now you have two parameters that will effect the error, the degree of the polynomials  $q$  and the number of elements  $N$ . Let  $u$  be a periodic function and set  $a = 1$  and fix  $q$  and investigate how the rate of convergence in the error depends on the numerical flux corresponding to  $\beta = 0, 1/2, 1$ . You may see different results for even and odd  $q$  so repeat the computation for several different choices.

### Approximate $u_t + (au)_x = 0$ on a grid with inter-element and physical boundary conditions (you just wrote your first discontinuous Galerkin solver)

Repeat the single elements experiments above but now for your full grid. Here the timestep will need to satisfy  $a\lambda_{\max}\Delta t \lesssim h$ , so if you keep the degree  $q$  around 5 you should be able to use RK4 without any problems.