Residual based artificial viscosity for advection-diffusion problems

Assignment A2

Introduction

In this assignment we are interested in solving time-dependent advection-diffusion equation in two space dimensions using a continuous Galerkin finite element approximation.

Consider the advection-diffusion equation:

$$\begin{aligned} \partial_t u(\boldsymbol{x},t) + \boldsymbol{\beta}(\boldsymbol{x}) \cdot \nabla u(\boldsymbol{x},t) - \nabla \cdot (\varepsilon \nabla u(\boldsymbol{x},t)) &= f(\boldsymbol{x},t), \quad (\boldsymbol{x},t) \in \Omega \times (0,T], \\ u(\boldsymbol{x},0) &= u_0(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega, \end{aligned} \tag{1}$$

with appropriate boundary conditions. Here u represents the temperature, $\boldsymbol{\beta} = (\beta_1, \beta_2)$ is the advection velocity, $\varepsilon > 0$ is the diffusion coefficient and T is the final time. Functions $f(\boldsymbol{x},t)$ and $u_0(\boldsymbol{x})$ are given data.

Part 1

Consider the case when $\partial_t u = 0$, $\varepsilon = 1$, $\boldsymbol{\beta} = (0,0)$ and homogeneous Dirichlet boundary condition. Then, the equation (1) converts into Poisson's equation:

$$\begin{aligned} -\Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x}) &= 0, \qquad \forall \boldsymbol{x} \in \partial\Omega. \end{aligned} \tag{2}$$

Let the computational domain be a square defined as $\Omega = \{\boldsymbol{x} : (x_1, x_2) \in [-1,1] \times [-1,1]\}$. Assume that the right hand side is

$$f(\boldsymbol{x}) = 8\pi^2 \sin(2\pi x_1) \sin(2\pi x_2),$$

which gives the following exact solution

$$u(\boldsymbol{x}) = \sin(2\pi x_1)\sin(2\pi x_2).$$

The computation domain Ω can be triangulated in Matlab as follows:

1. First, define the geometry:

   ```
   function r = Rectg(xmin,ymin,xmax,ymax)
   r = [2 xmin xmax ymin ymin 1 0;
        2 xmax xmax ymin ymax 1 0;
        2 xmax xmin ymax ymax 1 0;
        2 xmin xmin ymax ymin 1 0]';
   ```

2. Then, triangulate the geometry with respect to the maximum mesh-size $h_{\max}$:

   ```
   g = Rectg(-1,-1,1,1);
   hmax = 1/16;
   [p,e,t] = initmesh(g,'hmax',hmax);
   ```

   where `[p,e,t]` are point, edge and element data given by the triangulation in Matlab.

To apply the homogeneous Dirichlet boundary condition *strongly* (i.e. after assembling the linear system $\boldsymbol{A\xi} = \boldsymbol{b}$) one can do:

```
I = eye(length(p));          % construct the identity matrix
A = AssembleA(...);          % stiffness matrix
b = Assembleb(...);          % load vector
A(e(1,:),:) = I(e(1,:),:); % replace the rows corresponding
                             % to the boundary nodes by corresponding
                             % rows of I
b(e(1,:)) = 0;               % put the boundary value into the RHS
```

Problem 1.1. Write down a Galerkin finite element method for this problem using piecewise linear basis functions in space, then implement it in Matlab (or any other programming languages that you prefer). Plot the exact solution, Galerkin solution and the error for $h_{\max} = \frac{1}{8}$ and $h_{\max} = \frac{1}{16}$.

Problem 1.2. Compute the energy norm of the error $\|u-U\|_E^2 = \int_\Omega (\nabla u - \nabla U)\cdot(\nabla u - \nabla U)\,\mathrm{d}\boldsymbol{x}$ for sequence of meshes obtained by setting: $h_{\max} = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$. Then, find *the convergence rate* of the Galerkin approximation in the energy norm numerically, let us denote it by α. Plot $h_{\max}$ versus the energy norm of the error and $h_{\max}$ versus $h_{\max}^\alpha$ at the same figure using the *loglog*-plot in Matlab. Determine the value of α and motivate your answer.

The energy norm can be computed as `EnE=sqrt(e'*A*e);`, where `A` is the stiffness matrix and `e` is the error .

Part 2

Let us consider the problem (1) in the unit disk $\Omega = \{\boldsymbol{x} : x_1^2 + x_2^2 \le 1\}$ with two-dimensional rotational field $\boldsymbol{\beta}(\boldsymbol{x}) = 2\pi(-x_2, x_1)$, homogeneous Dirichlet boundary condition $u(\boldsymbol{x}, t) = 0, \forall \boldsymbol{x} \in \partial\Omega$ and $f(\boldsymbol{x}, t) = 0$. The mesh for this case can be easily constructed in Matlab as:

```
g = @circleg;
hmax = 1/16;
[p,e,t] = initmesh(g,'hmax',hmax);
```

Further, we need to compute the L^2-norm of the error several times in our computation: $\|e\|_{L^2(\Omega)} = (\int_\Omega e^2\,\mathrm{d}\boldsymbol{x})^{\frac{1}{2}}$. Provided that `M` is the mass matrix, this norm can easily be computed in Matlab as: `L2E=sqrt(e'*M*e);`.

Problem 2.1. Derive a Galerkin finite element formulation for problem (1) using piecewise continuous linear polynomials in space and Crank-Nicholson discretization in time and implement it in Matlab (or any other programming languages that you prefer).

Problem 2.2. Now set $\varepsilon = 0$, the initial data to:

$$u_0(\boldsymbol{x}, 0) = \tfrac{1}{2}\left(1 - \tanh\left(\tfrac{(x_1 - x_1^0)^2 + (x_2 - x_2^0)^2}{r_0^2} - 1\right)\right), \tag{3}$$

where, with $r_0 = 0.25$, $(x_1^0, x_2^0) = (0.3, 0)$, and solve the problem until $T = 1$. Test with different time-steps Δt. How large could you choose the time-step with respect to $h_{\max}$? Plot your solution at the final time for $h_{\max} = \frac{1}{8}$ and $h_{\max} = \frac{1}{16}$.

For the rest of your computation you can set the time-step to $\Delta t = \mathrm{CFL}\frac{h_{\max}}{\|\boldsymbol{\beta}\|_{L^\infty(\Omega)}}$, where CFL=0.5. You also need an exact solution for the problem to do convergence analysis.

For example the exact solution for time $T = 1$ is the initial data itself.

Problem 2.3. Compute the L^2-norm of the error for sequence of meshes obtained by setting: $h_{\max} = \{\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$. Then, find the convergence rate α of the approximation in the L^2-norm. Plot $h_{\max}$ versus the L^2-norm of the error and $h_{\max}$ versus $h_{\max}^{\alpha}$ at the same figure using the *loglog*-plot in Matlab. What is the convergence order?

Problem 2.4. What is the convergence order if you set $\varepsilon = 0.1$ instead? Is it better or worse? Why is it so?

Problem 2.5. Now set $\varepsilon = 0$, the initial data to:

$$u_0(\boldsymbol{x}, 0) = \begin{cases} 1 & \text{if } (x_1 - x_1^0)^2 + (x_2 - x_2^0)^2 \le r_0^2, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where, with $r_0 = 0.25$, $(x_1^0, x_2^0) = (0.3, 0)$,

Perform the same analysis as in Problem 2.3, obtain the plots and report them. What is the convergence order for this case? What happens with the solution? Motivate.

Problem 2.6. Set $\varepsilon = 0.001, 0.01, 0.1$, $h_{\max} = \frac{1}{16}$ and solve the Problem 2.5. What do you get? Which value of ε gives the best result? Why?

Part 3

We consider the problem (1) with the same advection speed $\boldsymbol{\beta}(\boldsymbol{x})$, right hand side $f(\boldsymbol{x}, t)$ and domain Ω as in Part 2. Let us define the residual based nonlinear artificial viscosity $\varepsilon(U_n)$ (RV method) as follows: For time $t = t_n$ compute:

$$\varepsilon(U_n)|_K = \min\left(C_{\max} h_K \||\boldsymbol{\beta}|\|_{L^\infty(K)}, C_{\text{res}} h_K^2 \frac{\|R(U_n)\|_{L^\infty(K)}}{\|U_n - \bar{U}_n\|_{L^\infty(\Omega)}} \right), \qquad \forall K \in \mathcal{T}_h, \tag{5}$$

$$R(U_n) = \frac{U_n - U_{n-1}}{\Delta t} + \boldsymbol{\beta} \cdot \nabla U_{n-1}, \tag{6}$$

where $C_{\max}, C_{\text{res}} > 0$ are user-defined $\mathcal{O}(1)$ constants, h_K is the size of element K, that can be the diameter or smallest edge, $\||\boldsymbol{\beta}|\|_{L^\infty(K)} = \max_{N_i \in K, i=0,1,2}\left((\beta_1^2 + \beta_2^2)^{\frac{1}{2}}(N_i) \right)$, where N_i, $i = 0, 1, 2$ is the nodes of the element K, is the local wave speed for element K, $\bar{U}_n$ is the average of U_n. $C_{\max} = 0.25$ and $C_{\text{res}} = 1$ work most of the times.

The residual based viscosity is a piecewise constant function defined in each element. You can implement in Matlab through the following loop:

```
...
nt = size(t,2);                     % get #-of elements
Cmax = 0.25;
Cres = 1.0;
Res = ...;                          % compute the residual
                                    % using the L2-projection
Res = Res/...;                      % normalize the residual
for K = 1:nt                        % iterate over all cells
    ...
    beta_K = ...;                   % get max-speed for K
    Res_K = ...;                    % get max-residual for K
    eps(K) = min(C1*h*beta_K, C2*h^2*Res_K);
end
```

Problem 3.1. Implement the RV method in Matlab. Solve Problem 2.3 with the RV method (i.e. obtain the figures and do the convergence analysis).

Problem 3.2. Now solve Problem 2.5 with the RV method. What happens with the solution? Do you gain anything in the convergence rate with compare to the result of Problem 2.5?

Part 4.

Goal-oriented error estimates

Advanced, non obligatory task

Consider the advection-diffusion equation with data from Part 2, *i.e.* $\Omega = \{\boldsymbol{x} : x_1^2 + x_2^2 \leq 1\}$, $\boldsymbol{\beta}(\boldsymbol{x}) = 2\pi(-x_2, x_1)$, $f(\boldsymbol{x}, t) = 0$, with boundary condition $u(\boldsymbol{x}, t) = 0$, $\forall \boldsymbol{x} \in \partial\Omega$.

The so-called *the adjoint* or *the dual* problem corresponding to the problem (1) is defined as following: find $z := z(\boldsymbol{x}, t)$ such that

$$\begin{aligned} -\partial_t z - \boldsymbol{\beta}\cdot\nabla z - \nabla\cdot(\varepsilon\nabla z) &= \psi_\Omega, && (\boldsymbol{x},t) \in \Omega \times [T,0),\\ -\boldsymbol{\beta}\cdot\boldsymbol{n} z &= \psi_{\partial\Omega}, && (\boldsymbol{x},t) \in \partial\Omega \times [T,0),\\ z(\boldsymbol{x},T) &= 0, && \boldsymbol{x} \in \Omega, \end{aligned} \tag{7}$$

where $\boldsymbol{n} = (n_1, n_2)^T$ is the outward normal to the boundary, $\psi_\Omega := \psi_\Omega(\boldsymbol{x}, t)$ and $\psi_{\partial\Omega} := \psi_{\partial\Omega}(\boldsymbol{x}, t)$ are the source terms with $\psi_\Omega \in L_2(\Omega \times [0, T])$ and $\psi_{\partial\Omega} \in L_2(\partial\Omega \times [0, T])$, which define a linear target functional by

$$\mathcal{M}(u) = \int_{\Omega\times[0,T]} u\psi_\Omega \,\mathrm{d}x\,\mathrm{d}t + \int_{\partial\Omega\times[0,T]} u\psi_{\partial\Omega}\,\mathrm{d}S\,\mathrm{d}t,$$

For example, the average of the error in domain Ω can be one example of the target functional, *i.e.* set $\psi_\Omega = 1$, $\psi_{\partial\Omega} = 0$, then:

$$\mathcal{M}(u) - \mathcal{M}(U) = \int_{\Omega\times[0,T]} (u - U)\,\mathrm{d}x\,\mathrm{d}t.$$

Theorem 1. *For the finite element approximation U:*
1. the following error representation holds:

$$|\mathcal{M}(u) - \mathcal{M}(U)| = \int_0^T \sum_{K\in\mathcal{T}_h} \int_K \mathcal{R}(U)(z - \pi_h z)\,\mathrm{d}x\,\mathrm{d}t, \tag{8}$$

2. the following error estimates holds:

$$|\mathcal{M}(u) - \mathcal{M}(U)| \le C\int_0^T \sum_{K\in\mathcal{T}_h} h_K \|\mathcal{R}(U)\|_{L^2(K)} \|Dz\|_{L^2(K)}\,\mathrm{d}t, \tag{9}$$

where U is finite element approximation of u, π_h is the standard interpolation function, $\mathcal{R}(U)$ is the residual, Dz the first derivative of z, h_K is the cell diameter.

Let us now define the following *error indicator* for each cell K:

$$\eta_K = Ch_K \|\mathcal{R}(U)\|_{L^2(K)} \|Dz\|_{L^2(K)}. \tag{10}$$

Then, the fully automatic goal oriented h-adaptive mesh refinement algorithm is:

1. Initialize coarse grid $\mathcal{T}_N$, choose $TOL > 0$, and set $N := 0$.
2. Solve primal problem (1).
3. Solve dual problem (7).
4. For every $K \in \mathcal{T}_N$: compute a posteriori error estimator η_K.
5. If $T\sum_{K\in\mathcal{T}_h} \eta_K < TOL$: STOP
6. Refine 10% of cells with the largest η_K.
7. Set $N := N + 1$ and goto step 2.

Problem 4.1. Prove Theorem 1.

Problem 4.2. Formulate a Galerkin finite element method of (7) using continuous piecewise linear polynomial approximation.

Problem 4.3. Implement in Matlab (or any other programming languages that you prefer) the goal-oriented adaptive algorithm defined above. Report the convergence history with respect to the average error in the domain as the target functional. Motivate your answer. What does the dual solution indicate?

Good luck!
Murtazo
Uppsala, November 2015