ATHENS Course Introduction to Finite Elements

Computer Assignment Day 5

Galerkin Finite Element Solution of a Convection-Diffusion Equation in One Dimension

In this assignment we aim at solving the convection-diffusion equation in one dimension. We will consider both the stationary as well as the time-dependent case. In the stationary case we assume that $x \in \Omega = (0,1)$, and in the time-dependent case we assume that $x \in \Omega = (0,1)$ and $t \in (0,T)$.

For the stationary case we aim at solving, given a small number $\epsilon > 0$ the following problem for $u(x)$

$$\epsilon \frac{d^2 u}{dx^2} - \frac{du}{dx} = 0 \text{ for } 0 < x < 1 \tag{1}$$

where $u(x)$ satisfies homogeneous and non-homogeneous Dirichlet conditions at the left and right end point of Ω, respectively. This means that

$$u(x=0) = 0 \quad \text{and} \quad u(x=1) = 1. \tag{2}$$

The parameter ϵ in (1) is the inverse of the Péclet number

$$\epsilon = \frac{1}{Pe}\,. \tag{3}$$

The Péclet number is a dimensionless number that gives the relative importance of the convective and diffusive transport. The parameter ϵ approaches 0 as the convective term becomes dominant.

For the time-dependent case, we aim at solving

$$\frac{\partial u}{\partial t} = \epsilon \frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial x} \text{ for } 0 < x < 1 \text{ and } 0 < t < T \tag{4}$$

subject to the same boundary conditions as above and the initial condition

$$u(x, t=0) = x\,. \tag{5}$$

1 Stationary Case

Theoretical Part

Assignment 1 Find an analytical solution to the boundary value problem (1)-(2). Do so in two steps. Assume in the first step that $u(x) = \exp(r\,x)$ and derive and solve the characteristic equation for r. Expand $u(x)$ as a linear combination of two basis solution functions. Use in the second step the boundary conditions to determine the integration constants.

Assignment 2 Plot the analytical solution found for various values of ϵ. Use for instance $\epsilon = 0.1, 0.01, 0.001, \ldots$. Can you explain the observed phenomenon?

Assignment 3 Cast the above convection-diffusion equation supplied with the boundary conditions at the left and right end point in its weak or variational form. Pay in particular attention to the order of derivatives used and to the choice of the vector space of test functions to resolve the boundary conditions. In the remainder of the assignment we aim at a development of a one-dimensional finite element code to solve the above equation.

Implementation Part without Stabilization

Assignment 4 Reuse the Matlab routines *GenerateMesh.m* and *GenerateTopology.m* to generate the mesh and its topology on Ω. Reuse the routine *ExactSolu.m* that return the exact solution as a function of the coordinate $x \in \Omega$.

Assignment 5 Write a routine, called *GenerateElementStiffnessMatrix.m*, that given the coordinates x_i and x_{i+1} of the end points of element e_i generates the 2×2 element matrix S_{e_i} such that

$$S_{e_i} = \frac{\epsilon}{x_{i+1} - x_i} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} . \tag{6}$$

Assignment 6 Derive an expression for the contribution of the convective term of each element. Write a routine, called *GenerateElementConvectiveMatrix.m* that computes this contribution.

Assignment 7 Write a routine, called *AssembleStiffnessMatrix.m*, that assembles the element matrices S_{e_i} on each element into the global matrix $(n + 1) \times (n + 1)$ matrix S. Do likewise in *AssembleConvectiveMatrix.m* for the convective term.

Assignment 9 Add the two global matrices. Modify the first and last of the discrete equations to take the Dirichlet boundary conditions into account. This requires modifying the first and last row of the global matrix as well as the last entry of the right-hand side vector.

Assignment 9 Initialize the right-hand side vector to a zero vector of size $N + 1$.

Assignment 10 Compute the discrete solution for a fixed mesh for various values of ϵ and for a fixed value of ϵ for various meshes. Can you explain the observed behavior?

Implementation part with stabilization

Incorporate streamline-upwind Petrov-Galerkin stabilization in the implementation previously developed.

2 Time dependent case

Extend the code previously developed to the time-dependent case.