**ATHENS Course Introduction to Finite Elements**

Computer Assignment Day 3

Galerkin Finite Element Solution of a Heat Equation in One Dimension

In this assignment, we solve a heat equation in one spatial dimension. A new issue is the integration of the partial differential equation in time. We consider the following problem for $u(x,t)$

$$(P_1)\begin{cases} \dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial x^2}, & \text{for } (x,t) \in (0,1)\times(0,T), \\ \dfrac{\partial u}{\partial x}(0,t) = \dfrac{\partial u}{\partial x}(1,t) = 0, & \text{for } t \in (0,T], \\ u(x,0) = \cos(\pi x) + 1, & \text{for } x \in (0,1). \end{cases} \tag{1}$$

# Compulsory Theoretical Part

**Assignment 1** *Show that the analytic solution of the above heat problem $(P_1)$, including the initial and boundary conditions, is given by*

$$u(x,t) = e^{-\pi^2 t}\cos(\pi x) + 1.$$

*We suggest using pen and paper. Plot this solution as a function of $x$ and $t$. You can use the* `meshgrid` *command in Matlab to generate this plot. Make sure to label the $x$ and $t$ axis in your plot.*

To solve problem $(P_1)$ by the use of the Galerkin finite element method, we have to formulate $(P_1)$ in a weak or variational sense.

**Assignment 2** *Give the continuous weak formulation of problem $(P_1)$, in which the order of the spatial derivatives is minimized. Take the boundary conditions and initial condition into account in this weak formulation.*

To obtain a finite element solution $u^h(x,t)$ that approximate the exact solution $u(x,t)$, we divide the spatial interval $(0,1)$ into $n$ elements $e_i$ such that $\cup_{i=1}^{n} e_i = (0,1)$. Each element is the interval $e_i = [x_i, x_{i+1}]$. We require that $x_1 = 0$ and $x_{n+1} = 1$. The finite element approximation can then be written as a linear combination of linear basis functions $\phi_j(x)$, for $j \in \{1,\ldots,n+1\}$ with time dependent coefficients, that is

$$u(x,t) = \sum_{j=1}^{n+1} c_j(t)\phi_j(x)\,. \tag{2}$$

The time evolution of the coefficients $\underline{c}$ is then given by the system of ordinary differential equations

$$M\underline{c}' + S\underline{c} = \underline{0}\,, \tag{3}$$

where $M$ and $S$ represent the *mass* matrix and *stiffness* matrix respectively. These matrices are formed by an element-by-element procedure.

**Assignment 3** *Give an expression for the element $2\times 2$ stiffness matrix $S_{e_i}$ and the mass matrix $M_{e_i}$. Use the trapezoidal rule to approximate the integrals on $e_i$.*

## Compulsory Implementation Part

**Assignment 4** *Write a matlab routine, called* GenerateMesh.m*, that returns as output a vector of equidistant meshpoints $x_i$, $i \in \{1, \ldots, n+1\}$ (You may use the file that you created previously).*

Further, we need to know which vertices belong to a certain element $e_i$. This is called the *topology* of the mesh.

**Assignment 5** *Write a routine, called* GenerateTopology.m*, that generates a two-dimensional array, called* elmat*, which contains the indices of the vertices of each element, that is*

$$\begin{matrix} elmat(i,1) = i \\ elmat(i,2) = i+1 \end{matrix}, \quad \text{for } i \in \{1, \ldots, n\} \tag{4}$$

*(You may use the file that you created previously).*

**Assignment 6** *Show that for linear basis functions, the element mass and element stiffness matrices, on element $i$, are given by*

$$M_{e_i} = \frac{x_{i+1} - x_i}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad S_{e_i} = \frac{1}{x_{i+1} - x_i} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}. \tag{5}$$

To obtain the global matrices $S$ and $M$, we need to do an assembly procedure that performs a loop over all the elements.

**Assignment 7** *Write a matlab routine, called* AssemblyStiffnessMatrix.m*, that performs this operation, where $S$ is initialized as a (sparse) zero n+1-by-n+1 matrix, and subsequently:*

$$\begin{aligned} &S(elmat(i,j), elmat(i,k)) = S(elmat(i,j), elmat(i,k)) + S_{e_i}(j,k), \\ &\text{for all elements } i \in \{1, \ldots, n\} \text{ and } j,k \in \{1,2\}. \end{aligned} \tag{6}$$

*Write a similar routine, called* AssemblyMassMatrix.m*, to get the global mass matrix.*

Now, we are at the stage of solving the problem

$$M\underline{u}' + S\underline{u} = \underline{0}, \qquad \text{supplied with initial conditions.}$$

We will consider several time integration methods.

**Assignment 8** *Write a routine for the (explicit) forward Euler time integration of this problem. Integrate from $t = 0$ to $t = 1$, with an element size of $h = 0.1$ and $\Delta t = 0.001$. Repeat the simulations with $\Delta t = 0.1$ and $h = 0.01$. What do you observe? Explain your results.*

**Assignment 9** *Change the routine in which the (implicit) backward Euler time integration. Repeat the simulations. Show plots of your solution and compare your result with the exact solution.*

**Assignment 10** *Change the routine in which the trapezoidal (Crank-Nicholson) method is used. Repeat the simulations. Compare your solution with the exact solution.*

## Elective Part

- use the Matlab command `ode45` to perform the time integration;
- implement second order Lagrangian elements in space;
- implement a non-zero forcing term;