Prof. Dr. Stefan Funken
M.Sc. Mladjan Radic, Stefan Hain
Department of Numerical Mathematics
Ulm University

Numerik von gewöhnlichen Differenzialgleichungen
SoSe 2016

Sheet 10

Due July 07, 2016.

Exercise 1 (Well-posedness)

Explain what is wrong in both the variational setting and the classical setting for the following BVP:

$$\begin{aligned} -u''(x) &= f(x), \qquad && x \in (0,1), \\ u'(0) &= u'(1) = 0. \end{aligned}$$

More precise: Explain in both contexts why this problem is not well-posed with respect to $H^1(\Omega)$.

Exercise 2 (Graduated Grids)

Let the exact solution of a given BVP be given by $u(x) = x^\alpha$ for $\alpha > \frac{1}{2}$ on $\Omega = (0,1)$. Let $u_I(x)$ be the linear interpolant of u on a given grid $\mathcal{T} := \{0 = x_0 < x_1 < \cdots < x_N < x_{N+1} = 1\}$, i.e. $u_I \in \mathcal{S}^{1,1}(\mathcal{T})$ and on a given element (u_i, u_{i+1}), $u_I(x)$ may be expressed by

$$u_I(x) = u(x_{i+1}) \cdot \frac{x_{i+1} - x}{x_{i+1} - x_i} + u(x_i) \cdot \frac{x - x_i}{x_{i+1} - x_i} = x_{i+1}^\alpha \cdot \frac{x_{i+1} - x}{x_{i+1} - x_i} + x_i^\alpha \cdot \frac{x - x_i}{x_{i+1} - x_i}.$$

We define the error $e(x) := u(x) - u_I(x)$. Compute the error in the H^1-semi-norm $|e(x)|_{H^1(\Omega)}$, in the L_2-norm $\|e(x)\|_{L_2(\Omega)}$ and in the H^1-norm $\|e(x)\|_{H^1(\Omega)} = \sqrt{\|e(x)\|^2_{L_2(\Omega)} + |e(x)|^2_{H^1(\Omega)}}$ for the following grids:

(i) $\mathcal{T} := \{x_i \mid x_i := \frac{i}{N}, \quad i = 0, \ldots, N\}$, (equidistant grid),

(ii) $\mathcal{T} := \{x_i \mid x_i := \left(\frac{i}{N}\right)^\beta, \quad i = 0, \ldots, N\}$, (graduated grid),

by choosing $\beta = \frac{1}{\alpha - \frac{1}{2}}$, $\beta = \frac{1}{\alpha}$, $\beta = \frac{1}{\alpha + \frac{1}{2}}$, $\beta = \frac{5}{\alpha - \frac{1}{2}}$, $\alpha \in \{\frac{3}{4}, 1, 2, 5\}$ and $N := \{10^0, 10^1, 10^2, 10^3, 10^4, 10^5\}$. What do you observe? What have you expected? In matlab you can use the command `x = linspace(0,1,N+1)` to obtain an equidistant grid. What happens if you use `x = logspace(0,1,N+1)` instead and use this as another graduated grid? Plot the error for the above mentioned grids.

Hint: You can use the following equations to implement the norm of the error:

$$\|e(x)\|^2_{L_2(\Omega)} = \int_\Omega (u(x) - u_I(x))^2 \, dx = \sum_{i=0}^{N} \int_{x_i}^{x_{i+1}} (u(x) - u_I(x))^2 \, dx.$$

and analogously

$$|e(x)|^2_{H^1(\Omega)} = \int_\Omega (u'(x) - u_I'(x))^2 \, dx = \sum_{i=0}^{N} \int_{x_i}^{x_{i+1}} (u'(x) - u_I'(x))^2 \, dx.$$

Exercise 3 (FEM)

We consider the following BVP

$$
\begin{aligned}
&-\big(a(x)u'(x)\big)' + b(x)u'(x) + c(x)u(x) = f(x), \qquad x \in \Omega = (0,1) \qquad (1)\\
&u(0) = \alpha, \qquad u(1) = \beta
\end{aligned}
$$

Show, that the variational formulation (for $\alpha = \beta = 0$) is given by: Find $u \in V := H_0^1(\Omega)$, such that

$$\int_\Omega a(x)u'(x)v'(x)\;dx + \int_\Omega b(x)u'(x)v(x)\;dx + \int_\Omega c(x)u(x)v(x)\;dx = \int_\Omega f(x)v(x)\;dx$$

for all $v \in V$. In this sheet, we want to consider another strategy for the implementation of this more general equation and a slightly different strategy for assembling the stiffness matrix. The 1d-grid should be stored in the matrices `coordinates` $\in \mathbb{R}^{n_C \times 1}$ and `elements` $\in \mathbb{R}^{n_E \times 2}$. The matrix `coordinates` contains the coordinates of the grid points and `elements` contains the indices for the edge points of each interval. Let us at first consider the interval $(0,1)$, which is partitioned in only two elements, i.e. $E_1 = (0, \frac{1}{2}) = (x_1, x_2)$ and $E_2 = (\frac{1}{2}, 1) = (x_2, x_3)$. It is then clear, that the corresponding matrices are `coordinates` $= \begin{pmatrix} 0 & 0.5 & 1 \end{pmatrix}^T$ and `elements` $= \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}^T$. We have already seen, that for the Laplace problem, i.e. $a(x) = 1, b(x) = c(x) = 0$ and with $h_1 = x_1 - x_2 = \frac{1}{2}, h_2 = x_3 - x_2 = \frac{1}{2}$, we obtain the very small stiffness-matrix (by using the hat functions):

$$A = \begin{pmatrix} 2\frac{1}{h_1} & -\frac{1}{h_1} & 0 \\ -\frac{1}{h_1} & \frac{1}{h_1} + \frac{1}{h_2} & -\frac{1}{h_2} \\ 0 & -\frac{1}{h_2} & 2\frac{1}{h_2} \end{pmatrix} = \begin{pmatrix} 2\frac{1}{h_1} & -\frac{1}{h_1} & 0 \\ -\frac{1}{h_1} & \frac{1}{h_1} & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{h_2} & -\frac{1}{h_2} \\ 0 & -\frac{1}{h_2} & 2\frac{1}{h_2} \end{pmatrix}$$

which is now our first motivation for assembling the stiffness matrix element-wise. As we now see above, we just need to to compute smaller 2×2-matrices and afterwards summing them up with respect to the corresponding position. The next motivation for assembling the smaller 2×2-matrices is, that we do not have to number the nodes lexicographically, i.e. $x_1 = 0, x_3 = 0.5, x_2 = 1$, which ends up in $E_1 = (0, \frac{1}{2}) = (x_1, x_3)$ and $E_2 = (\frac{1}{2}, 1) = (x_3, x_2)$ and `coordinates` $= \begin{pmatrix} 0 & 0.5 & 1 \end{pmatrix}^T$ and `elements` $= \begin{pmatrix} 1 & 3 \\ 3 & 2 \end{pmatrix}^T$. And the corresponding matrix is now given by

$$A = \begin{pmatrix} 2\frac{1}{h_1} & 0 & -\frac{1}{h_1} \\ 0 & 2\frac{1}{h_2} & -\frac{1}{h_2} \\ -\frac{1}{h_1} & -\frac{1}{h_2} & \frac{1}{h_1} + \frac{1}{h_2} \end{pmatrix} = \begin{pmatrix} 2\frac{1}{h_1} & 0 & -\frac{1}{h_1} \\ 0 & 0 & 0 \\ \frac{1}{h_1} & 0 & \frac{1}{h_1} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2\frac{1}{h_2} & -\frac{1}{h_2} \\ 0 & -\frac{1}{h_2} & \frac{1}{h_2} \end{pmatrix}$$

This demonstrates, that we have to compute 4 entries on each element and afterwards just summing them up with respect to their position. This will be now discussed in more detail.

(a) Draw the grid for

`coordinates` $= \begin{pmatrix} 0 & 0.5 & 0.3 & 0.6 & 1.0 & 0.9 \end{pmatrix}^T$

`elements` $= \begin{pmatrix} 1 & 3 & 2 & 4 & 6 \\ 3 & 2 & 4 & 6 & 5 \end{pmatrix}^T$

The vector `dirichlet` contains the indices for the Dirichlet nodes, in our case this vector is given by `dirichlet`$= \begin{pmatrix} 1 & 5 \end{pmatrix}^T$.

We want to consider now the uniformly refinement of the grid. Each element will be halved. The procedure is then as follows:

(1) For each element compute the midpoint and store the coordinate of these midpoints at the end of the vector. In the above example, where we have considered the tow elements $E_1 = (0, \frac{1}{2}) = (x_1, x_2)$ and $E_2 = (\frac{1}{2}, 1) = (x_2, x_3)$, we compute the new midpoints $x_4 = 0.25$ and $x_5 = 0.75$. Therefore we obtain `coordinates` $= \begin{pmatrix} 0 & 0.5 & 1 & 0.25 & 0.75 \end{pmatrix}^T$

(2) Compute the new elements. In our example, we would obtain `elements` $= \begin{pmatrix} 1 & 4 & 2 & 5 \\ 4 & 2 & 5 & 3 \end{pmatrix}$.

With the help of Matlab, this procedure can be realized very efficient.

(b) Compute the matrices `coordinates` and `elements` for the refined grid in (a). Implement a function

```
[coordinates,elements] = function refineMesh(coordinates,elements)
```

which realized the refinement of a given grid as described above.

The next step is the assemblation of the stiffness matrix and this will be done element-wise, as we already tried to describe above. Consider the j-th element $T_j = [x_j, x_{j+1}]$ of the given grid. On T_j we only have to consider φ_j and φ_{j+1}, because all other functions vanish. For the j-th element we compute the small 2×2 matrix

$$\begin{pmatrix} \int_{T_j} a(x) \cdot \varphi_j'(x) \cdot \varphi_j'(x) dx & \int_{T_j} a(x) \cdot \varphi_j'(x) \cdot \varphi_{j+1}'(x) dx \\ \int_{T_j} a(x) \cdot \varphi_{j+1}'(x) \cdot \varphi_j'(x) dx & \int_{T_j} a(x) \cdot \varphi_{j+1}'(x) \cdot \varphi_{j+1}'(x) dx \end{pmatrix}.$$

This 2×2 matrix will then be added on the corresponding position. Note, that we have set $b(x) = c(x) = 0$.

(c) Visualize this procedure on a sheet of paper with (a).

The right-hand side of the equation will also be computed element-wise.

(d) Write a function `[A,B,C,b] = assemble(coordinates,elements,f)`, which assembles the matrices A, B and C and the right-hand side b element-wise. `f` is a function-handle for the right-hand side of the equation. Note, that A is the matrix associated to the diffusion-term $\int_\Omega a(x)u'(x)v'(x)\ dx$, B is the matrix associated to the convection term $\int_\Omega b(x)u'(x)v(x)\ dx$ and C is the matrix associated to the reaction term $\int_\Omega c(x)u(x)v(x)\ dx$.

(e) Complete the script `main.m`, which is given on the home-page. In this script, the grid is loaded and the BVP (1) is solved numerically. It is assumed, that $a(x), b(x), c(x)$ are constant in $\Omega = (0, 1)$. Test your implementation with the following BVPs

(i) $a = 1$, $b = 0$, $c = 0$, $f = 1$ and $u(0) = u(1) = 0$,

(ii) $a = 1$, $b = 0$, $c = 0$, $f = 1$ and $u(0) = 0$,$u(1) = 1$,

(iii) $a = 1$, $b = 0$, $c = 0$, $f = x$ and $u(0) = u(1) = 0$,

(iv) $a = 1$, $b = 0$, $c = 1/100$, $f = 1$ and $u(0) = u(1) = 0$,

(v) $a = 1$, $b = 1/10$, $c = 0$, $f = 1$ and $u(0) = u(1) = 0$.