Priv.-Doz. Dr. Christian Rieger
Christopher Kacwin

universitätbonn

Sheet 4

Submission on **Thursday, 7.6.18.**

Exercise 1. (1D heat equation)

We consider a metal rod and its temperature distribution

$$y\colon [0,1] \times [0,T] \longrightarrow \mathbb{R}$$

with initial condition $y(\cdot,0) = y^0$. Additionally,we assume that we are able to control the heat flux of the metal rod at the end points. More precisely, we model $y(x,t)$ to satisfy the partial differential equation

$$\begin{aligned} y_t - y_{xx} &= f \quad \text{in } [0,1] \times [0,T] \\ -y_x(0,\cdot) &= l \quad \text{in } [0,T] \\ y_x(1,\cdot) &= r \quad \text{in } [0,T] \\ y(\cdot,0) &= y^0 \quad \text{in } [0,1] \end{aligned}$$

with control parameters $l(t)$, $r(t)$ and additional enviromental influence $f(x,t)$ (material conditions, additional heat source...).

As a first step, we want to do a time discretization of the partial differential equation. We interpret $y(x,t) = y(t)(x) = y(t) \in V$ (f likewise), where y is now a function of time mapping into a function space V, which consists of functions defined on $[0,1]$ (for instance $C[0,1]$). Introducing the time steps $t_n = nT/N$ for $n = 0,\dots,N$ we define $y^n = y(t_n) \in V$, $f^n = f(t_n) \in V$, $l(t_n) = l^n \in \mathbb{R}$, $r(t_n) = r^n \in \mathbb{R}$.

a) Use the implicit Euler scheme to derive the time-discretized formulation

$$\begin{aligned} y^n &= y^{n-1} + \frac{T}{N}(f^n + y^n_{xx}), \quad n = 1,\dots,N \\ -y^n_x(0) &= l^n, \quad n = 1,\dots,N \\ y^n_x(1) &= r^n, \quad n = 1,\dots,N\,. \end{aligned}$$

This is a system of N Poisson-like differential equations, which from now on we consider in their weak form. Next, we do a spatial discretization $V_h \subset V$ with basis functions $\{\phi_0,\dots,\phi_m\}$.

b) Using the coefficient vector $\underline{y}^n \in \mathbb{R}^{m+1}$ with the Ansatz

$$y^n \approx \sum_{i=0}^{m} \underline{y}^n_i \phi_i\,,$$

derive the time-space discretized formulation

$$\left(M + \frac{T}{N}K\right)\underline{y}^n = \frac{T}{N}L^n + M\underline{y}^{n-1}\,, \quad n = 1,\dots,N\,. \tag{1}$$

Here, $M \in \mathbb{R}^{(m+1)\times(m+1)}$ is the mass matrix with $M_{ij} = \int \phi_i\phi_j$, $K \in \mathbb{R}^{(m+1)\times(m+1)}$ is the stiffness matrix with $K_{ij} = \int (\phi_i)_x(\phi_j)_x$, and $L^n \in \mathbb{R}^{m+1}$ is the load vector with $L^n_i = \int f^n\phi_i + \phi_i(0)l^n + \phi_i(1)r^n$.

(10 points)

Programmieraufgabe 1. (1D heat equation)

The goal of this programming exercise is the implementation of the routine outlined in Exercise 1. This will take several steps:

- Calculation and Assemblation of M, K and L for a given finite element basis. For M and K, we assemble the matrices as sparse multiplication routines to gain computational speed.
- Implementation of an iterative solver for the equations (1).
- Solution plotting, variation of input parameters, ...

For the space discretization, we use piecewise linear, continuous elements.
Let $m \in \mathbb{N}$. We consider the spacial nodes $x_i = i/m$, $i = 0, \dots, m$. The nodal basis to the piecewise linear, continuous finite element space is given by

$$\phi_i(x) = \begin{cases} m(x - (i-1)/m) & x \in [(i - 1/m), i/m] \\ 1 - m(x - i/m) & x \in [i/m, (i+1)/m] \\ 0 & \text{else} \end{cases}$$

for $i = 0, \dots, m$.

a) Calculate $K_m, M_m \in \mathbb{R}^{(m+1)\times(m+1)}$ for this basis. Calculate $L^n_{m,i}$ for the case that f^n is piecewise constant on the intervals $[i/m, (i+1)/m]$ for $i = 0, \dots, m-1$.

b) Implement a routine which takes $m \in \mathbb{N}$, $l, r \in \mathbb{R}$ and $b \in \mathbb{R}^m$ as input and returns as output the vector

$$L = [L_i]_{i=0}^{m+1} = \left[\int_0^1 \phi_i(x) b(x)\, \mathrm{d}x + \phi_i(0) l + \phi_i(1) r\right]_{i=0}^{m+1}.$$

Here, $b(x)$ is the piecewise constant function with $b|_{[i/m,(i+1)/m]} \equiv b_{i+1}$ for $i = 0, \dots, m-1$.

c) Implement a routine which takes $m \in \mathbb{N}$, $a \in \mathbb{R}$ and a vector $b \in \mathbb{R}^{m+1}$ as input and returns as output the vector $(M_m + aK_m)b \in \mathbb{R}^{m+1}$. Make sure that this routine has runtime $\mathcal{O}(m)$ (use the sparse structure of the matrices).

d) Implement an iterative solver for linear equation systems of the form

$$(M_m + aK_m)b = c$$

which uses the matrix-vector-product routine from part b). Possible choices are for instance the CG-method or the Jacobi method.

e) Write the final routine which solves the system (1). The input is given as $T \in \mathbb{R}$, $m, N \in \mathbb{N}$, $\underline{y}^0 = [\underline{y}^0_i]_{i=0}^m \in \mathbb{R}^{m+1}$, $f = [f^n_i]_{n=1,i=0}^{N,m-1} \in \mathbb{R}^{N\times m}$, and $l = [l^n]_{n=1}^N$, $r = [r^n]_{n=1}^N \in \mathbb{R}^N$. For $n = 1, \dots, N$, one obtains $\underline{y}^n \in \mathbb{R}^{m+1}$ sequentially as stated in (1). The output is given by $y = [y^n_i]_{n=0,i=0}^{N,m+1} \in \mathbb{R}^{(N+1)\times(m+1)}$.

f) Test your implementation with the following input data: $T = 10$, $m = 100$, $N = 1000$, $\underline{y}^0 \equiv 0 \in \mathbb{R}^{m+1}$, $l \equiv r \equiv -0.1 \in \mathbb{R}^N$, and $f = [f^n_i]_{n=1,i=0}^{N,m-1} \in \mathbb{R}^{N\times m}$ defined via

$$f^n_i = \begin{cases} 1 & i = 50\,, \\ 0 & \text{else.} \end{cases}$$

Visualize the solution in a comprehensive manner, e.g. a GIF-animation.

g) One can easily show that the solution y to the original PDE satisfies

$$\int_0^1 y(x,0)\,\mathrm{d}x = \int_0^1 y(x,t)\,\mathrm{d}x \quad \forall t \in [0,T]\,.$$

Does the computed approximation show an analogous behaviour? Why/Why not?

(30 points)

The programming exercise should be handed in either before/after the exercise class on 14.6.18 (bring your own laptop!) or in the HRZ-CIP-Pool, after making an appoint- ment at 'angelina.steffens@uni-bonn.de'. All group members need to attend the presentati- on of your solution. Closing date for the programming exercise is the 14.6.2018. You can choose the programming language yourself.