Technische Universität Berlin SS 2017
Institut für Mathematik
Prof. Dr. G. Bärwolff
Sekr. MA 4-5 29.06.2017

3. Exercise sheet - solutions

FV/FD-Methods for the solution of pde's

1) Exercise
Solve the equation

$$u_t = u_{xx} + u_{yy} \; .$$

numerically on

$$\Omega =]0, \pi[\times]0, \pi[$$

with homogeneous boundary conditions, and the initial condition

$$u(x, y, 0) = \begin{cases} 1 & \text{for } |x - \frac{\pi}{2}| < \frac{1}{2} \text{ and } |y - \frac{\pi}{2}| < \frac{3}{2} \\ 1 & \text{for } |x - \frac{\pi}{2}| < \frac{3}{2} \text{ and } |y - \frac{\pi}{2}| < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \; .$$

Use an appropriate finite difference discretization of Ω and test implicit and explicit time discretizations. Follow the solution to a steady state ($\frac{\partial u}{\partial t} \approx 0$).

Solution:
With the matrices

$$B = \begin{pmatrix} 2 & -1 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ \vdots & & & & & \\ 0 & \dots & 0 & -1 & 2 & -1 \\ 0 & \dots & 0 & 0 & -1 & 2 \end{pmatrix} \in \mathbb{R}^{N \times N} \, ,$$

the identity $I_N = eye(N)$ and the discretization parameter $h = \frac{\pi}{N}$ we create the stiffness matrix

$$A = B \otimes I_N + I_N \otimes B + \frac{h^2}{\tau}(I_N \otimes I_N) \in \mathbb{R}^{N^2 \times N^2}$$

and the r.h.s

$$F = \frac{h^2}{\tau}(I_N \otimes I_N)\hat{U} \; ,$$

where $\hat{U} \in \mathbb{R}^{N \times N}$ is the vector consisting of the unknown values $\hat{u}_{ij}$ at the gridpoints $(i * h, j * h)$ at the time level $t = k\tau$. The solution U at the time level $t = (k+1)\tau$ we get with the Euler implicit method by the solution of

$$A\,U = F \; .$$

We start this process at $t = 0$ with the projection of $u(x, y, 0)$ onto the grid $\Omega_h = \{(i\,h, j\,h) | i = 1, \dots, N, j = 1, \dots, N\}$.

2) Exercise
We consider the inital boundary value problem

$$\begin{aligned} u_t = u_{xx} \quad & \text{in } \Omega =]0,1[\\ u(0,t) = u(1,t) &= 0 \\ u(x,0) &= \sin(\pi x) \end{aligned}$$

with the exakt solution

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

Use the horizontal method of lines to solve the problem numerically and compare the numerical solutions to the exact solution. Use an equidistant finite difference discretization of u_{xx} and Ω.
Test different ode-solver including Euler-explicit and Euler-imlicit and MATLAB ode45 or an equivalent Octave-ode-solver.

Solution:
Solution:
At the gridpoints $x_i = i\,h$, $i = 1,\ldots,N, h = \frac{1}{N+1}$ we get the ode-system

$$\frac{\partial u_i(t)}{\partial t} = \frac{2u_i(t) - u_{i+1}(t) - u_{i-1}(t)}{h^2} \ , \quad i = 1,\ldots,N,$$

$(u_0(t) = u_{N+1}(t) = 0)$ with the initial values

$$u_i(0) = \sin(i\,h\pi) \ , \quad i = 1,\ldots,N \ .$$

This initial value problem is now to solve with the named ode-solver.

3) Exercise
The heat conduction number and the temperature conduction number of a potato are

$$\lambda = 0,16\frac{W}{m\,K} \ , \qquad a = 5,6\cdot 10^4\frac{m^2}{3600\,s}$$

and the convective heat transfer coefficient is in some sense uncertain, we suppose it as

$$\alpha = 30\frac{W}{m^2\,K} \ .$$

The potato is supposed as a ball Ω with the radius $R = 5\,cm$. We consider such a potato, which has after cooking the homogeneous temperature of $373\,K$. The potato was cooled in a big can with water of a constant temperature of $u_\infty = 291\,K$. The heat conduction equation is

$$u_t = a(u_{xx} + u_{yy} + u_{zz}) \ ,$$

the initial condition is

$$u(x,y,z,0) = 373\,K \ .$$

On the boundary $\Gamma = \partial\Omega$ we consider a bc of third kind

$$-\lambda\frac{\partial u}{\partial\vec{n}} = \alpha(u - u_\infty) \ ,$$

where $\frac{\partial}{\partial\vec{n}}$ is the derivative in the outer normal direction (r-direction). Determine the minimal time when the maximal temperatur of the potato is less or equal to $333\,K$.

Because of the radial symmetry it is useful to understand this problem as a 2,5d problem (2d in polar coordinates).

Because of the independence of the coordinates θ and φ the 1d problem

$$u_t = \frac{a}{r^2}\frac{\partial}{\partial r}[r^2 \frac{\partial u}{\partial r}]$$

is to solve for the function $u(r,t)$ on $]0, R[$ with the boundary condition

$$-\lambda \frac{\partial u}{\partial r} = \alpha(u - u_\infty) \; .$$

Because of the bc of thrid kind an analytical solution seems to be complicable. Thus a numerical solution should be found. With a Finite-Volume ansatz ($dV \implies r^2 \sin\theta \, d\theta d\varphi dr$)

$$\int_\omega u_t \, r^2 \sin\theta d\theta d\varphi dr = \int_\omega \frac{a}{r^2}\frac{\partial}{\partial r}[r^2 \frac{\partial u}{\partial r}] r^2 \sin\theta d\theta d\varphi dr$$

and the independance of θ and φ we get

$$\int_{r_{i-1/2}}^{r_{i+1/2}} u_t \, r^2 \, dr = \int_{r_{i-1/2}}^{r_{i+1/2}} \frac{\partial}{\partial r}[r^2 \frac{\partial u}{\partial r}] \, dr$$

and by an appropriate implicit approximation of the time derivative we get the Finite-Volume approximation

$$\frac{u_i - \hat{u}_i}{\tau} \frac{r^3_{i+1/2} - r^3_{i-1/2}}{3} = r^2_{i+1/2} \frac{u_{i+1} - u_i}{\Delta r} - r^2_{i-1/2} \frac{u_i - u_{i-1}}{\Delta r}$$

where $\hat{u}_i$ is the numerical solution at time $t_k = k\tau$ and u_i ($i = 1, \ldots, n$) is the numerical solution at the new time level $t_{k+1} = t_k + \tau$.[1] The bc is approximated by

$$-\lambda \frac{u_{n+1} - u_n}{\Delta r} = \alpha(u_{n+1} - u_\infty)$$

and is used to eliminate the unknown u_{n+1}. At the end we have to solve a linear equation system for the unknowns $U = (u_1 \; u_2 \; \ldots \; u_n)^T$ per time-step, starting with $k = 1$ and the values $\hat{u}_i$ at time $t = 0$ from the initial condition.

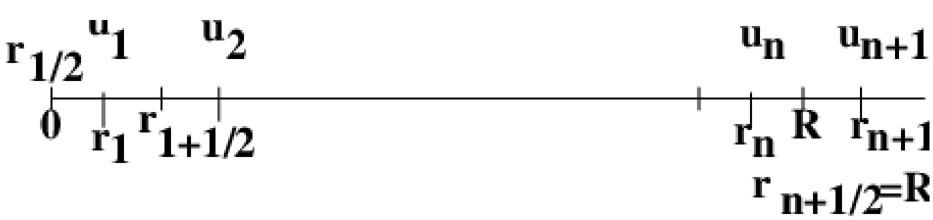

Abbildung 1: discretization of $\Omega = [0, R]$

Listing 1: source code

```
% hot potato problem – exercise 4.3
%
%
n = 25;
```

[1]Instead of the use of the factor $\frac{r^3_{i+1/2} - r^3_{i-1/2}}{3}$ it's also possible to use $r_i^2 * \Delta r$ as a result of the midpoint rule

```
R0 = 0.;
R1 = 1.0/20.;
drho = (R1-R0)/n;
Ubound = 333;
U0 = 373.;
Uu = 291.;
a = 5.6/3600*1.0e-4;
lambda = 0.16;
alpha = 30;
%
% Zentralpunkte der Finiten Zellen
rho = linspace(R0+drho/2,R1-drho/2,n);
% Randpunkte der Finiten Zellen
rhop = linspace(R0,R1,n+1);
% Matrixaufbau Ar
Ar = zeros(n,n);
u0 = ones(n,1)*U0;
%
for i=2:n-1
  Ar(i,i) = a*(rhop(i+1)^2 + rhop(i)^2)/drho/(rho(i)^2*drho);
  Ar(i,i-1) = (-a*rhop(i)^2/drho)/(rho(i)^2*drho);
  Ar(i,i+1) = (-a*rhop(i+1)^2/drho)/(rho(i)^2*drho);
end
  Ar(1,1) = (a*(rhop(1)^2+rhop(2)^2)/drho)/(rho(1)^2*drho);
  Ar(1,2) = (-a*rhop(2)^2/drho)/(rho(1)^2*drho);
  Ar(n,n) = (a*rhop(n)^2/drho + a*rhop(n+1)^2/drho)/(rho(n)^2*drho) ...
      - a*rhop(n+1)^2/drho*(lambda/drho)/(lambda/drho + alpha)/(rho(n)^2*drho);
  Ar(n,n-1) =  (-a*rhop(n)^2/drho)/(rho(n)^2*drho);
%
for i=1:n
  Ualt(i) = U0;
end
R = zeros(n,1);
% Zyklus
% rechte Seite (Beruecksichtigung der Randbedingung)
  for i=1:n
   if (i < n) R(i) =  0; end
   R(n) = ...
        a*rhop(n+1)^2*alpha/drho/(lambda/drho + alpha)*Uu/(rho(n)^2*drho);
  end
%
Time_min=80; %time in minutes
Time=Time_min*60; % time in seconds
%
odefun=@(t,x) -Ar*x+R; % function of right side
[T,U]=ode23s(@(t,x)odefun(t,x),[0,Time],u0); %ode23s works faster for stiff
%
% Plot
for i=1:length(T)
  plot(U(i,:))
end
% Plot
```