S. Mishra
L. Scarabosio
J. Šukys

Spring Term 2014

Numerical Methods for Partial Differential Equations

ETH Zürich
D-MATH

Homework Problem Sheet 5

Introduction. The first problem is the implementation of Crank-Nicolson time stepping scheme coupled with linear finite element discretization of the heat equation. You will reuse most of the FEM code that you have written for the Problem 1 of the Exercise sheet 3. The second problem is the "introductory" problem for the *hyperbolic* partial differential equations, the method of characteristics and the finite volume method using upwinding, which will be introduced in the lectures on May 5th-6th.

Problem 5.1 Parabolic Timestepping with Crank-Nicolson (Core problem)

Let $\Omega := (0,1)^2$ and consider the problem

$$\begin{aligned} \frac{\partial u}{\partial t} - \Delta u &= f && \text{in } (0,T] \times \Omega, \\ u &= g && \text{on } (0,T] \times \partial\Omega, \\ u &= u_0 && \text{on } \{0\} \times \Omega, \end{aligned}$$

where f, g and u_0 are given such $u(t,\mathbf{x}) = \cos(2\pi x_1)\sin(t\pi x_2)$ is the exact solution.

(5.1a) Derive the variational formulation for this parabolic problem.

HINT: Fix $t \in (0,T)$ and integrate by parts in $\mathbf{x}$ to obtain conditions on $u(t) \in H^1(\Omega)$.

(5.1b) Show that the initial value problem arising from a spatial discretization of the variational formulation using piecewise linear finite elements with basis functions $\{b_N^i\}_i \in V_N$ is given by

$$\begin{aligned} \boldsymbol{M}\frac{\mathrm{d}}{\mathrm{d}t}\vec{\boldsymbol{\mu}}(t) + \boldsymbol{A}\vec{\boldsymbol{\mu}}(t) &= \boldsymbol{F}(t), \\ \boldsymbol{M}\vec{\boldsymbol{\mu}}(0) &= \vec{\boldsymbol{\mu}}_0, \end{aligned} \tag{5.1.1}$$

where $\vec{\boldsymbol{\mu}}(t)$ is the finite element coefficient vector of $u_N(t)$, $(\vec{\boldsymbol{\mu}}_0)_i = (u_0, b_N^i)$ is the finite element coefficient vector of the projection of u_0 onto V_N, $\boldsymbol{F}$ is the time-dependent load vector

$$F_i(t) = \int_\Omega f(t,\mathbf{x}) b_N^i(\mathbf{x})\,\mathrm{d}\mathbf{x},$$

and $\boldsymbol{M}$ and $\boldsymbol{A}$ are the mass- and Galerkin matrices respectively

$$M_{ji} = \int_\Omega b_N^i(\mathbf{x}) b_N^j(\mathbf{x})\,\mathrm{d}\mathbf{x}, \qquad A_{ji} = \int_\Omega \operatorname{grad} b_N^i(\mathbf{x}) \cdot \operatorname{grad} b_N^j(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

(5.1c) For an initial value problem

$$\frac{\partial}{\partial t}\boldsymbol{y} = \boldsymbol{h}(t, \boldsymbol{y}), \quad \boldsymbol{y}(0) = \boldsymbol{y}_0,$$

let the time-stepping scheme be given for $m = 0, \ldots, K$ by the *Crank-Nicolson scheme*

$$\boldsymbol{y}^{(m+1)} = \boldsymbol{y}^{(m)} + \frac{1}{2}\Delta t\big(\boldsymbol{h}(t_m, \boldsymbol{y}^{(m)}) + \boldsymbol{h}(t_{m+1}, \boldsymbol{y}^{(m+1)})\big),$$

with initial value $\boldsymbol{y}^{(0)} = \boldsymbol{y}_0$, time step $\Delta t := T/K$ and time points $t_m := m\Delta t$.

Show that the Crank-Nicolson scheme applied to (5.1.1) gives the following linear system:

$$\left(\boldsymbol{M} + \frac{1}{2}\Delta t\boldsymbol{A}\right)\vec{\boldsymbol{\mu}}^{(m+1)} = \left(\boldsymbol{M} - \frac{1}{2}\Delta t\boldsymbol{A}\right)\vec{\boldsymbol{\mu}}^{(m)} + \frac{1}{2}\Delta t(\boldsymbol{F}_{m+1} + \boldsymbol{F}_m), \tag{5.1.2}$$

where $\boldsymbol{F}_m = \boldsymbol{F}(t_m)$.

(5.1d) Next, we aim to implement the linear FEM with Crank-Nicolson time stepping (5.1c). For this purpose, we will reuse the routines from the Problem 1 of the Exercise sheet 3. The Galerkin matrix assembly routines

```
Aloc = STIMA_Heat_LFE(Vertices, QuadRule, FHandle)
  A = assemMat_LFE(Coordinates,EHandle,varargin)
```

can be reused without any modifications.

Modify the load vector assembly routines

```
  Lloc = LOAD_LFE(Vertices, QuadRule, FHandle)
L = assemLoad_LFE(Coordinates,QuadRule,FHandle)
```

to compute the *time-dependent* load vector $\mathbf{F}$ in (5.1.1).

HINT: For the detailed description of the above functions, refer to the Problem 1 of the Exercise sheet 3.

(5.1e) Implement the local mass matrix routine

```
Aloc = MASS_LFE(Vertices, QuadRule, FHandle),
```

which will be used in the `assemMat_LFE` routine for the assembly of the global mass matrix $\mathbf{M}$ in (5.1.1).

HINT: Modify the existing routine `Aloc = STIMA_Heat_LFE`.

(5.1f) Implement a function

```
U = Crank_Nicolson_LFE(Mesh, K, T, G_HANDLE, F_HANDLE, U0_HANDLE)
```

to compute the FE solution (vector of coefficients `U`) using the `K` iterations of the Crank-Nicolson time stepping scheme (5.1.2) up to a specified time `T`.

HINT: For *each* iteration of the Crank-Nicolson time stepping, you will need to solve (numerically) the resulting linear system for the coefficients of `U`.

HINT: For efficiency, construct the matrices $\boldsymbol{M} + \frac{1}{2}\Delta t \boldsymbol{A}$ and $\boldsymbol{M} - \frac{1}{2}\Delta t \boldsymbol{A}$ only once.

HINT: Use the supplied function `P7O6` for any quadrature that you might need.

HINT: For the implementation of the Dirichlet boundary conditions, reuse the routine

```
[U,FreeDofs] = assemDir_LFE(Mesh,BdFlag,GHandle)
```

from the Problem 1 of the Exercise sheet 3 - you will need to modify it to accept an additional argument `t` indicating the time t.

(5.1g) Implement a function

```
U = plot_Crank_Nicolson_LFE()
```

which computes the solution using `Crank_Nicolson_LFE` for $T = 0.5, 1.0, 1.5, 2.0$ and plots it using the `plot_LFE` routine from the Problem 1 of the Exercise sheet 3 (you may modify the `plot_LFE` routine to indicate the time T in the title). Use the mesh `Square3.mat` and $K = 100$ time steps (for $T = 2$; use proportionally smaller K for other values of T). For each time $T = 0.5, 1.0, 1.5, 2.0$, you can recompute the solution from $t = 0$.

Does your implementation of the Crank-Nicolson time stepping scheme approximate the exact solution correctly?

(5.1h) Implement a function

```
U = conv_Crank_Nicolson_LFE()
```

which computes the solution for $T = 1$ using `Crank_Nicolson_LFE` on the series of meshes

```
Square1.mat  -  Square4.mat
```

and plots the $L^2(\Omega)$-error convergence. What type of convergence and what order do you observe?

Use the supplied function `P7O6` for any quadrature that you might need. For each level of mesh refinement, use the number of timesteps needed to balance the errors from time and space discretization.

HINT: Use the provided `L2Err_LFE` routine to compute the $L^2(\Omega)$-error of the solution.

Problem 5.2 Transport in One Dimension

Consider the one-dimensional linear transport equation:

$$\begin{aligned} U_t + (a(x)U)_x &= 0, \qquad \forall (x,t) \in \mathbb{R} \times \mathbb{R}_+, \\ U(x,0) &= U_0(x), \qquad \forall x \in \mathbb{R}, \end{aligned} \tag{5.2.1}$$

with coefficient $a(x) \in C^1(\mathbb{R})$.

(5.2a) Write down the equation for characteristics of (5.2.1). Use it to derive an expression for the exact solution.

HINT: Assume that a is an increasing function of x.

(5.2b) Let $U(x,t)$ be a smooth solution of (5.2.1), that decays to zero at infinity. Then show that U satisfies the energy bound

$$\int_{\mathbb{R}} U^2(x,T)dx \leq e^{CT} \int_{\mathbb{R}} U_0^2(x)dx, \tag{5.2.2}$$

for all $T > 0$, with constant C depending on $\|a\|_{C^1}$.

(5.2c) Consider the equation (5.2.1) on the domain $D = (0,1)$ with periodic boundary conditions and $a = -1$. Implement a stable numerical scheme to simulate (5.2.1). Plot the results at $T = 1$ and 200 mesh cells for the following initial conditions:

Smooth Solution

$$U_0(x) = \sin(2\pi x) \tag{5.2.3}$$

Non-smooth Solution

$$U_0(x) = \begin{cases} 1, & \text{if} \quad x < 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{5.2.4}$$

(5.2d) Plot $L^1(D)$ and $L^\infty(D)$ errors vs numbers of cells (for no. of cells 100, 200, 400, 800, 1600, 3200, 6400). Use exact solution derived in sub-problem (5.2a) to calculate the errors. Comment the observed results. In which case does the method converge? What is the convergence rate?

Published on April 30th.

To be submitted on May 12th.

Last modified on May 2, 2014