Advanced Discretization Methods (WS 19/20)
Homework 10

(P. Knabner, L. Wester)

Deadline for submission (theory):	January 14th, 2019, 12:15
Deadline for submission (programming):	January 14th, 2019, 12:15

Remark: When you apply theorems, whether they were found in Knabner/Angermann or another source, make sure to **cite them**!

Exercise 19: Estimates for parabolic equations (3+2+2+3)

Let Ω be a smooth and bounded domain, $V := H_0^1(\Omega)$ and $H := L^2(\Omega)$. We further denote by $a : V \times V \to \mathbb{R}$ the familiar bilinear form

$$a(u,v) := \int_\Omega \nabla u \cdot \nabla v \, dx$$

and set $u_0 \in V$, $f \in L^2(0,T;H)$, and $T > 0$. Show that the following a priori estimates hold for the solution $u \in H^1(0,T;V)$ of the following boundary value problem (with initial condition $u(0) = u_0$):

$$\int_\Omega \partial_t u(t) v \, \mathrm{d}x + a(u(t),v) = \int_\Omega f(t) v \, \mathrm{d}x \qquad \text{for all } v \in V,\ t \in (0,T).$$

a) $\|u(s)\|_0 \leq \|u_0\|_0 + \int_0^s \|f(t)\|_0 \, dt$,

b) $\|u(s)\|_0^2 + \int_0^s |u(t)|_1^2 \, dt \leq \|u_0\|_0^2 + C \int_0^s \|f(t)\|_0^2 \, dt$ for some $C > 0$, and

c) $|u(s)|_1^2 + \int_0^s \|\partial_t u(t)\|_0^2 \, dt \leq |u_0|_1^2 + \int_0^s \|f(t)\|_0^2 \, dt$.

Let $u \in H^1(0,T;V)$ be a weak solution of the problem

$$\begin{cases} \partial_t u - \Delta u = f & \text{in } (0,\infty) \times \Omega, \\ u = u_D & \text{on } (0,\infty) \times \partial\Omega, \\ u = u_0 & \text{in } \{0\} \times \Omega. \end{cases}$$

for $f, u_0, u_D \in C(\overline{\Omega})$.

d) Recall Grönwall's lemma to show that $u(t,x)$ converges for $t \to \infty$ to the weak solution of

$$\begin{cases} -\Delta v = f & \text{in } \Omega, \\ v = u_D & \text{on } \partial\Omega. \end{cases}$$

Programming exercise 9: Time-stepping with BDF (10)

Dust off your (or the official) solution to Programming exercise 2 and expand it with the function `BDF.m` given as:

```
uh = BDF(...,DT,T,steporder)
```

where `steporder` refers to the version of the BDF-stepping algorithm to be called upon. We want to consider `steporder` $\in \{1,2,3\}$, defined for a differential equation of the type

$$\partial_t u = F(t,u)$$

as

- BDF1 (Implicit Euler):
$$\frac{U^{n+1} - U^n}{h} = F^{n+1}$$
- BDF2:
$$\frac{U^{n+1} - \frac{4}{3}U^n + \frac{1}{3}U^{n-1}}{h} = \frac{2}{3}F^{n+1}$$
- BDF3:
$$\frac{U^{n+1} - \frac{18}{11}U^n + \frac{9}{11}U^{n-1} - \frac{2}{11}U^{n-2}}{h} = \frac{6}{11}F^{n+1}$$

Generate any additionally required initial values other than u_0 with the Crank-Nicolson method rather than passing exact data. Test your code with the example from Programming exercise 2.