Advanced Discretization Methods (WS 19/20)
Homework 11
(P. Knabner, L. Wester)

Deadline for submission (theory): January 21st, 2019, 12:15
Deadline for submission (programming): January 21st, 2019, 12:15

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

Exercise 20: Stability of the one-step θ method (10)

Consider the scalar model problem

$$\begin{aligned} \xi' + \lambda\xi &= 0 \qquad r \in (0,T) \\ \xi(0) &= \xi_0 \end{aligned}$$

for some $T > 0$ and $\lambda \in \mathbb{C}$ with $\mathrm{Re}(\lambda) > 0$. Applying the one-step θ method to this model yields a scheme of the form

$$\xi^{n+1} = R_\theta(-\lambda\Delta t)\xi^n$$

for a certain function $R_\theta : \mathbb{C} \to \mathbb{C}$.
Determine the domains of stability, defined as

$$S_{R_\theta} := \{z \in \mathbb{C} \ : \ |R_\theta(z)| < 1\}$$

for $\theta \in \{0, \frac{1}{2}, 1\}$. Further show that only the method for $\theta = 1$ is L-stable, i.e. it satisfies

$$R_\theta(z) \to 0 \text{ for } \mathrm{Re}(z) \to -\infty$$

Exercise 21: Fully discrete estimate (5)

Derive a fully-discrete equivalent to the estimate proven in Exercise 19a) when discretizing in time with the Implicit Euler method with varying time step size $\tau_n := t_{n+1} - t_n$.

Programming exercise 10: A fractional θ-scheme (10)

Given a problem of the form: Find $u \in V$, s.t.

$$(\partial_t u, v) + a(u,v) = (f,v) \qquad \forall v \in V$$

with appropriate boundary & initial data, we can define a fractional θ stepping method that is solved in 3 steps:

1. $$\frac{U^{n+\theta} - U^n}{\theta\Delta t} + \alpha a(U^{n+\theta}, v) + (1-\alpha)a(U^n, v) = (f^n, v)$$
2. $$\frac{U^{n+1-\theta} - U^{n+\theta}}{\theta'\Delta t} + (1-\alpha)a(U^{n+1-\theta}, v) + \alpha a(U^{n+\theta}, v) = (f^{n+1-\theta}, v)$$
3. $$\frac{U^{n+1} - U^{n+1-\theta}}{\theta\Delta t} + \alpha a(U^{n+1}, v) + (1-\alpha)a(U^{n+1-\theta}, v) = (f^{n+1-\theta}, v)$$

For some $\theta \in (0, \frac{1}{2})$, $\alpha \in (0,1)$ and $\theta' := 1 - 2\theta$. Expand your code from Programming exercise 9 by a function

```
uh = FractionalTheta(...,DT,T,theta,alpha)
```

which solves the familiar diffusion-reaction problem with the fractional θ method. Test your code with the same solution as before and for different choices of α, θ.