Die Mathematik ist doch die angenehmste Wissenschaft; sie und die Astronomie vertreten bei mir Tanzgesellschaften, Konzerte und andere derartige Belustigungen, die ich nur dem Namen nach kenne.
Friedrich Wilhelm Bessel (1784-1846), deutscher Astronom und Mathematiker

UNIVERSITÄT DES SAARLANDES
FR Mathematik
Prof. Dr. S. Rjasanow
T. Keßler, M. Sc.

Numerik partieller Differentialgleichungen
Sommersemester 2020
Blatt 2

Aufgabe 1 (3 Punkte)
Seien $a, b \in \mathbb{R}$ mit $a < b$. Wir wollen zeigen, dass für alle $\varepsilon > 0$ $\eta \in C_0^\infty(\Omega)$ existiert, so dass η nur Werte in $[0,1]$ annimmt, $\eta = 1$ auf $[a,b]$ und $\operatorname{supp}\eta \subseteq [a-\varepsilon, b+\varepsilon]$. Dazu gehen wie folgt vor: Es sei $\psi : \mathbb{R} \to \mathbb{R}$ die Funktion

$$\psi(x) = \begin{cases} \exp\left(-\dfrac{1}{1-x^2}\right), & |x| < 1, \\ 0, & \text{sonst}, \end{cases} \qquad x \in \mathbb{R}.$$

Aus der Vorlesung ist bekannt, dass $\psi \in C_0^\infty(\mathbb{R})$ mit $\operatorname{supp}\psi = [-1,1]$. Zeigen Sie, dass für $a < b \in \mathbb{R}$ und $\varepsilon > 0$ die Funktion

$$\psi_{a,b,\varepsilon}(x) = \frac{1}{\varepsilon c} \int\limits_a^b \psi\left(\frac{x-y}{\varepsilon}\right) \mathrm{d}y, \quad x \in \mathbb{R},$$

unendlich of differenzierbar ist mit Träger $[a-\varepsilon, b+\varepsilon]$ und dass für $x \in [a+\varepsilon, b-\varepsilon]$ $\psi_{a,b,\varepsilon}(x) = 1$. Dabei ist $c = \int_{\mathbb{R}} \psi(y)\,\mathrm{d}y$. Konstruieren daraus die geforderte Funktion η.

Aufgabe 2 (6 Punkte)
Sei $\Omega \subseteq \mathbb{R}$ offen. Zeigen Sie für $f \in C(\Omega)$:

$$f = 0 \iff \forall \psi \in C_0^\infty(\Omega) : \int\limits_\Omega f(x)\psi(x)\,\mathrm{d}x = 0.$$

Hinweis: Bilden Sie für die Implikation von rechts nach links die Kontraposition und nutzen Sie Aufgabe 1.

Aufgabe 3 (5 Punkte)
Zeigen Sie, dass das σ-Schema aus der Vorlesung für $\sigma = 1/2$ bei hinreichender Glattheit der Lösung u und der rechten Seite f die Approximationsordnung $O(\tau^2 + h^2)$ liefert.

Allgemeines zu den Programmieraufgaben
Sie dürfen für den praktischen Teil in `python` oder `C` programmieren. Zur Abgabe schicken Sie Ihren Quelltext donnerstags vor der Vorlesung an `kessler@num.uni-sb.de`. Hilfreiche Pakete für `python` sind `numpy` und `scipy`. Für `C` empfiehlt sich die GNU Scientific Library, welche zahlreiche grundlegende Datentypen und Algortihmen umfasst.

Aufgabe 4 (6 + 4* Punkte) — Praktische Aufgabe
In dieser Aufgabe implimentieren wir das σ-Schema für die Wärmeleitungsgleichung in einer Raumdimension,

$$\begin{aligned} u_t &= u_{xx} + f, && t > 0,\ x \in (0,1),\\ u(x,0) &= u_0(x), && x \in (0,1),\\ u(0,t) &= \mu_0(t), && t > 0,\\ u(1,t) &= \mu_1(t), && t > 0. \end{aligned}$$

Wir unterteilen $[0,1]$ in ein äquidistantes Gitter der Maschenweite $h > 0$ mit $N+1$ Knoten,

$$x_i = ih, \quad i = 0, \dots, N$$

Zum Zeitpunkt $t^j = j\tau$ mit $\tau > 0$ schreiben wir u_i^j für den Wert der Approximation im Knoten x_i. Das σ-Schema lautet damit

$$\frac{u_i^{j+1} - u_i^j}{\tau} = \sigma \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{h^2} + (1-\sigma)\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2} + \varphi_i^j,$$

für $i = 1, \dots, N-1$ and $j \in \mathbb{N}$. Weiterhin ist $u_0^j = \mu_0(t^j)$ und $u_N^j) = \mu_1(t^j)$. Zeigen Sie, dass sich mit $u^j = (u_1^j, \dots, u_{N-1}^j)^\top$ und $\varphi^j = (\varphi_1^j, \dots, \varphi_{N-1}^j)$ das System in Matrixschreibweise bringen lässt:

$$\begin{aligned} (I + \sigma A)u^{j+1} =&(I + (\sigma - 1)A)u^j\\ &+ \frac{\tau}{h^2}\Big(\sigma\mu_0(t^{j+1} + (1-\sigma)\mu_0(t^j))\Big)e_1\\ &+ \frac{\tau}{h^2}\Big(\sigma\mu_1(t^{j+1} + (1-\sigma)\mu_1(t^j))\Big)e_{N-1}\\ &+ \tau\varphi^j, \end{aligned}$$

wobei e_1, e_{N_1} die entsprechenden Standardbasisvektoren sind, $I \in \mathbb{R}^{(N-1)\times(N-1)}$ die Einheitsmatrix ist und

$$A = \frac{\tau}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

eine dünnbesetzte Tridiagonalmatrix ist. implimentieren Sie das explizitete Verfahren $\sigma = 0$ und testen Sie es mit $f = 0$, $\mu_0 = 0$, $\mu_1 = 1$ sowie verschiedene Anfangsbedingungen. Was beobachten Sie, wenn $\tau > h^2/2$ oder wenn sie u^0 durch Zufallszahlen initialisieren?

Zusatz (4 Punkte): Ändern Sie Ihren Code so ab, dass Sie die Wärmeleitungsgleichung mit dem Crank-Nicolson-Verfahren ($\sigma = 1/2$) lösen. Dabei hilft Ihnen die Funktion `spsolve` *aus* `scipy`*, die* `C`*-Bibliothek* `umfpack` *oder das cg-Verfahren.*