Serie 13

Abgabedatum: Abgabe mit TAs besprechen

Forum: https://forum.math.ethz.ch/c/spring-23/numerische-methoden-phys/

Webpage: https://metaphor.ethz.ch/x/2023/fs/401-2664-00L/

1. *Lineare ODEs mit Krylov-Verfahren*

Gegeben sei das lineare System gewöhnlicher Differentialgleichungen:

$$\begin{aligned}\dot{\underline{y}}(t) &= -i\mathbf{A}\underline{y}(t)\\ \underline{y}(0) &= \underline{y}_0\end{aligned}$$

mit der Hermite-symmetrischen Matrix $\mathbf{A} \in \mathbb{C}^{d\times d}$.

a) Zeigen Sie, dass die $\|\cdot\|_2$-Norm der Lösung durch die Evolution erhalten bleibt:

$$\|\underline{y}(t)\|_2 = \|\underline{y}(0)\|_2\,.$$

Hierbei ist:

$$\|\underline{u}\|_2^2 = \langle \underline{u}, \underline{u}\rangle = \sum_{j=1}^{d} \bar{u}_j u_j\,.$$

b) Verwenden Sie die Diagonalisierung von $\mathbf{A}$, um eine formale Lösung des Differentialgleichungssystems zu finden.

Hinweis: Siehe Beispiel 4.1.3 im Skript.

c) Verwenden Sie ein Krylov-Verfahren um eine numerische Lösung des Differentialgleichungssystems zu finden. Suchen Sie dafür eine Lösung:

$$\begin{aligned}\underline{u}_m(t) &\in \mathcal{K}_m(\mathbf{A}, \underline{y}_0)\\ \underline{u}_m(0) &= \underline{y}_0\,,\end{aligned}$$

so dass das Residuum $\dot{\underline{u}}_m(t) + i\mathbf{A}\underline{u}_m(t)$ orthogonal auf dem Krylov-Raum $\mathcal{K}_m(\mathbf{A}, \underline{y}_0)$ steht.

d) Wir wollen nun unsere Methoden für die Matrizen $\mathbf{A}$

1. `sqrt`
2. `minij`
3. `dvr`

welche im Template `krylov.py` implementiert sind, anwenden. Die Lösung soll mit

$$\underline{y}_0 = \frac{1}{\sqrt{d}}[1, 1, \ldots, 1]^{\mathrm{T}}$$

bis zur Zeit $t = 10^{-2}$ numerisch berechnet werden. Für das Krylov-Verfahren verwenden Sie sowohl das *Arnoldi*- als auch das *Lanczos*-Verfahren. Benutzen Sie Als Referenzlösung das Ergebnis welches `expm` aus `scipy.linalg` liefert. Geben Sie die Rechenzeiten und die Fehler bezüglich der Referenzlösung aus. Welchen Einfluss hat der Parameter m (Dimension des Krylov-Raums) auf die Lösung?

2. *Exponentielles Euler-Verfahren (Prüfungsaufgabe FS14)*

Betrachten Sie das *exponentielle Euler-Verfahren* mit konstanter Schrittweite:

$$\underline{y}_{k+1} = \underline{y}_k + h\varphi(h\mathbf{J_f})f(\underline{y}_k), \quad k = 0, \ldots, N \tag{1}$$

wobei:

$$\mathbf{J_f} := \mathrm{D}\, f(\underline{y}_k), \quad \varphi(z) = \frac{e^z - 1}{z}$$

a) Leiten Sie die Stabilitätsfunktion $S(z)$ von (1) her.

b) Schreiben Sie eine `Python`-Funktion `expEV` die das nichtlineare Anfangswertproblem

$$\dot{\underline{y}} = \begin{bmatrix} \dot{y_1} \\ \dot{y_2} \end{bmatrix} = \begin{bmatrix} -\frac{y_1^2}{y_2} + y_2 \log(y_2) \\ -y_1 \end{bmatrix}, \quad \underline{y}(0) = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

mit dem exponentiellen Eulerverfahren (1) mit konstanter Schrittweite löst.

Hinweis: Verwenden Sie das Template `exp_euler.py`.

Hinweis: Die Aufgabe wird viel leichter wenn Sie beachten, dass $Df \in \mathbb{R}^{2\times 2}$ klein ist. Sie können also hier einfach `expm` verwenden.

c) Bestimmen Sie empirisch die Konvergenzordnung des Verfahrens. Betrachten Sie das Zeitintervall $[0, 6]$ und berechnen Sie den Fehler bezüglich der exakten Lösung:

$$\underline{y}(t) = \begin{bmatrix} -\cos(t)\exp(\sin(t)) \\ \exp(\sin(t)) \end{bmatrix}$$

für verschiedene Anzahl von Zeitschritten $N = 24,\ 48,\ 96,\ 192,\ 384$.

3. ⋆ **Kernaufgabe: Zeitabhängige Schrödinger-Gleichung** ⋆

Wir betrachten die zeitabhängige Schrödinger-Gleichung mit Potential V:

$$\begin{cases} i\varepsilon\frac{\partial u}{\partial t} &= -\frac{1}{2}\varepsilon^2\frac{\partial^2 u}{\partial x^2} + V(x)u, \\ u(x,0) &= g(x), \qquad x \in [-1, 1] \end{cases} \tag{2}$$

wobei $\varepsilon = 0.01$ die Rolle von $\hbar$ übernimmt. Sei $N = 2^l$. Wir wählen eine Lösung:

$$u_N(x,t) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} c_k(t) e^{2\pi i k x}, \qquad x \in [-1,1],$$

so dass (2) in $\underline{x} = [x_j] = [\frac{2j}{N}]$ mit $j = -\frac{N}{2}, \ldots, \frac{N}{2} - 1$ erfüllt ist. Sei $\underline{c} = [c_0, \ldots, c_{\frac{N}{2}-1}, c_{-\frac{N}{2}}, \ldots, c_{-1}]$ der Vektor mit den Fourier-Koeffizienten. Da $\underline{c} = \mathcal{F}_N[u_N(\underline{x})](\underline{k})$ erhalten wir:

$$i\dot{c}_k = \frac{1}{2}\varepsilon k^2 c_k + \left(\mathcal{F}_N V_N \mathcal{F}_N^{-1} \underline{c}\right)_k$$

mit:

$$V_N = \frac{1}{\varepsilon}\operatorname{diag}(V(\underline{x})),$$

oder:

$$i\dot{\underline{c}} = \frac{1}{2}\varepsilon D_N^2 \underline{c} + \mathcal{F}_N V_N \mathcal{F}_N^{-1} \underline{c},$$

mit:

$$D_N = \operatorname{diag}\left(0, \ldots, \frac{N}{2} - 1, -\frac{N}{2}, \ldots, -1\right).$$

Deshalb gilt nun:

$$\boxed{\dot{\underline{c}} = -\frac{1}{2} i\varepsilon D_N^2 \underline{c} - i\mathcal{F}_N V_N \mathcal{F}_N^{-1} \underline{c}} \tag{3}$$

und wir haben:

$$e^{-ih\frac{\varepsilon}{2}D_N^2}\underline{c} = \left[e^{-\frac{1}{2}i\varepsilon h k^2} c_k\right], \quad \text{mit} \quad k = 0, \ldots, \frac{N}{2} - 1, -\frac{N}{2}, \ldots, -1 \tag{4}$$

$$e^{-ihV_N} u_N(\underline{x}) = \left[e^{-ihV(x_j)} u_N(\underline{x})_j\right]_{j=-\frac{N}{2}}^{\frac{N}{2}-1}. \tag{5}$$

a) Verwenden Sie (4) und (5) um das Strang-Splitting für (2) via der formalen Lösung von Gleichung (3) zu implementieren.

Hinweis: benutzen Sie das Template `tdse_Template.py`. Alle benötigten Parameter sind dort zu finden.

b) Bleiben die Norm und die Energie erhalten? Testen Sie mit folgenden Potentialen:

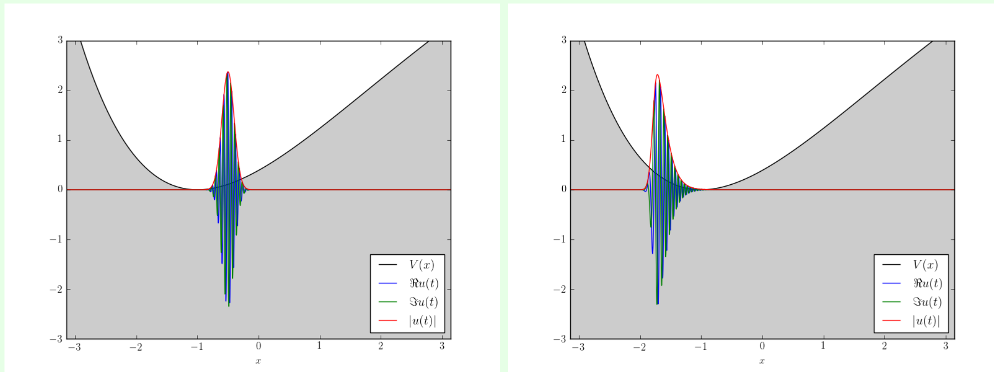
- Harmonisch: $V(x) = \frac{1}{2}x^2$
- Morse Potential: $V(x) = V_0\left(1 + e^{-2\beta(x+1)} - 2e^{-\beta(x+1)}\right)$ mit den Parametern $V_0 = 8$ und $\beta = \frac{1}{4}$.

und Anfangswerten ($\varepsilon = 10^{-2}$):

$$g_0(x) = \left(\frac{1}{\pi\varepsilon}\right)^{\frac{1}{4}} e^{-\frac{1}{2\epsilon}(x+\frac{1}{2})^2} e^{-i\frac{x}{\varepsilon}},$$

$$g_1(x) = \left(\frac{1}{\pi\varepsilon}\right)^{\frac{1}{4}} e^{-\frac{1}{2\epsilon}x^2}.$$

Plotten Sie das Potential und die Lösungen.



(a) Wellenpaket φ am Anfang.

(b) Wellenpaket φ im Zeitschritt 273.

Abbildung 1: Zeitentwicklung eines Wellenpakets