



Modellreduktion mit Proper Orthogonal Decomposition

2. Programmierblatt – Abgabe: Dienstag, 08.01.2013, 8:15 Uhr per Email

Ziel ist es, die lineare Wärmeleitungsgleichung mit Dirichlet-Nullrandbedingung mittels Reduced Order Modelling (ROM) zu lösen.

1. Modifikationen an `SolverPde`:

- Der Anfangswert `data.input.z0` soll in $\mathbb{R}^{NxNy \times 1}$ liegen statt in $\mathbb{R}^{Nx \times Ny}$.
- Der Löser soll auch auf inhomogene Probleme anwendbar sein, d.h. mit einer rechten Seite `data.input.f` aus $\mathbb{R}^{NxNy \times Nt}$ umgehen können.
- Die Funktion soll neben Ψ auch die Identität $\Phi \in \mathbb{R}^{NxNy \times NxNy}$ als `data.output.Phi` zurückgeben.
- Φ und Ψ sollen nur berechnet werden, wenn das Feld `data.input.assem` mit dem Character `TRUE` belegt ist.
- Die Verfahren zum Lösen des resultierenden Systems gewöhnlicher Differenzialgleichungen in t sind so zu modifizieren, dass `data.output.z` in $\mathbb{R}^{NxNy \times Nt}$ eine diskrete Lösung von

$$\Phi \dot{z}(t) + \Psi z(t) = f(t), \quad \Phi z(0) = z_0$$

ist (wobei Φ nicht die Identitätsmatrix sein muss!)

2. Funktion `AssemPod` zur Berechnung einer POD-Basis:

- Funktionsaufruf: `data = AssemPod(data)`.
- Eingabe- und Ausgabeparameter:

```
data.input.x ..... (Nx+2)x1 discretization of [ax,bx]
data.input.y ..... (Ny+2)x1 discretization of [ay,by]
data.input.t ..... Ntx1 discretization of [0,T]
data.input.Snap ..... (NxNy)xNt snapshots
data.input.rank ..... 1x1 rank l of pod basis

data.output.SingVal ..... 1x1 singular values
data.output.Basis ..... (NxNy)x1 pod basis
```

- Die Funktion soll die Trapezregel-Gewichtungsmatrizen `TimeWeights` $\in \mathbb{R}^{Nt \times Nt}$ und `SpaceWeights` $\in \mathbb{R}^{NxNy \times NxNy}$ bestimmen (vgl. Aufgabe 7) und dann aus den Snapshots mittels Eigenwertzerlegung (`eigs`, nicht `svd`, vgl. Thm. 1.3.2) eine POD-Basis vom Rang ℓ berechnen.
- Beachten Sie: `sqr(A)` berechnet die Wurzeln der Einträge von A . Zur Berechnung von \sqrt{A} steht der Befehl `sqr(A)` zur Verfügung, der allerdings nur nicht zu große Matrizen vom Format `full` verarbeiten kann. Bekannterweise liefern die beiden Verfahren bei

Diagonalmatrizen das gleiche Ergebnis – warum ist das so? Wie bestimmen Sie die Wurzel einer Nicht-Diagonalmatrizen A ohne die MATLAB-Funktion `sqrtn`?

3. Script `program02`:

a) Deklaration der Daten: $\Omega = [0, 1]^2$, $N_x = N_y = N_t = 200$, $\Theta = (0, 1)$, $f(t; x, y) = tx - y^2$, $z_0(x, y) = \chi_{[0.25, 0.75]^2}(x, y)$, $\sigma = 0.05$.

b) Berechnen Sie mittels `SolverPde` eine Lösung $z \in \mathbb{R}^{NxNy \times Nt}$ von

$$\begin{aligned} \dot{z}(t; x, y) - \sigma \Delta z(t; x, y) &= f(t; x, y) && \text{in } \Theta \times \Omega, \\ z(t; x, y) &= 0 && \text{in } \Theta \times \partial\Omega, \\ z(0; x, y) &= z_0(x, y) && \text{in } \Omega. \end{aligned}$$

Verwenden Sie dabei Rannacher-Smoothing (`RS`).

c) Bestimmen Sie mittels `AssemPod` eine POD-Basis $(\psi_1, \dots, \psi_\ell)$ vom Rang $\ell = 1, \dots, 25$, $\psi_i \in \mathbb{R}^{NxNy \times 1}$.

d) Berechnen Sie die Daten des reduzierten Modells $\Psi^\ell, \Phi^\ell \in \mathbb{R}^{\ell \times \ell}$, $z_0^\ell \in \mathbb{R}^{\ell \times 1}$ und $f^\ell \in \mathbb{R}^{\ell \times Nt}$.

e) Lösen Sie das reduzierte Anfangswertproblem (vergessen Sie nicht, `assem` auf `FALSE` zu setzen, sonst überschreiben Sie Φ^ℓ, Ψ^ℓ !).

f) Transformieren Sie die reduzierte Lösung zurück in einen Vektor $z_{\text{ROM}}^\ell \in \mathbb{R}^{NxNy \times Nt}$.

g) Bestimmen Sie die Approximationsfehler $\text{MaxErr}(\ell)$ und $\text{IntErr}(\ell)$,

$$\begin{aligned} \text{MaxErr}(\ell) &= \max_{(x,y) \in \Omega} \max_{t \in \Theta} |z_{\text{ROM}}^\ell(t; x, y) - z(t; x, y)|, \\ \text{IntErr}(\ell) &= \left(\int_{\Omega} \int_{\Theta} |z_{\text{ROM}}^\ell(t; x, y) - z(t; x, y)|^2 dt d(x, y) \right)^{\frac{1}{2}}, \end{aligned}$$

wobei die Integrale wie üblich mittels Trapezregel berechnet werden.

h) Stellen Sie die beiden Approximationsfehler und die berechneten Eigenwerte $\sigma_1, \dots, \sigma_\ell$ in jeweils einem Schaubild dar (logarithmische Skalen `semilogy` könnten nützlich sein). Kommentieren Sie die Zusammenhänge und Unterschiede.

i) Generieren Sie für die FDM-Lösung ein Video, vgl. Programmieraufgabe 1.