Universität Konstanz
Fachbereich Mathematik und Statistik

Prof. Dr. Stefan Volkwein
Martin Gubisch

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`:
   a) Der Anfangswert `data.input.z0` soll in $\mathbb{R}^{NxNy\times 1}$ liegen statt in $\mathbb{R}^{Nx\times Ny}$.
   b) Der Löser soll auch auf inhomogene Probleme anwendbar sein, d.h. mit einer rechten Seite `data.input.f` aus $\mathbb{R}^{NxNy\times N_t}$ umgehen können.
   c) Die Funktion soll neben Ψ auch die Identität $\Phi \in \mathbb{R}^{NxNy\times NxNy}$ als `data.output.Phi` zurückgeben.
   d) Φ und Ψ sollen nur berechnet werden, wenn das Feld `data.input.assem` mit dem Character `TRUE` belegt ist.
   e) 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
   $$\Phi\dot{z}(t) + \Psi z(t) = f(t), \qquad \Phi z(0) = z_0$$
   ist (wobei Φ nicht die Identitätsmatrix sein muss!)

2. Funktion `AssemPod` zur Berechnung einer POD-Basis:
   a) Funktionsaufruf: `data = AssemPod(data)`.
   b) 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 ........ lx1 singular values
data.output.Basis .......... (NxNy)xl pod basis
```

   c) 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. Kap. 4.1, S. 22) eine POD-Basis vom Rang ℓ berechnen.
   d) Beachten Sie: `sqrt(A)` berechnet die Wurzeln der Einträge von A. Zur Berechnung von $\sqrt{A}$ steht der Befehl `sqrtm(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 `sqrtm`?

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 $\mathrm{z} \in \mathbb{R}^{NxNy \times Nt}$ von

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

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

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

   d) Berechnen Sie die Daten des reduzierten Modells $\Psi^\ell, \Phi^\ell \in \mathbb{R}^{\ell \times \ell}$, $\mathrm{z}_0^\ell \in \mathbb{R}^{\ell \times 1}$ und $\mathrm{f}^\ell \in \mathbb{R}^{\ell \times N_t}$.

   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 $\mathrm{z}_{\mathrm{ROM}}^\ell \in \mathbb{R}^{NxNy \times Nt}$.

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

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

   wobei die Integrale wie üblich mittels Trapezregel berechnet werden.

   h) Stellen Sie die beiden Approximationsfehler und die berechneten Eigenwerte $\sigma_1, ..., \sigma_\ell$ 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.