

Serie 8

Abgabedatum: 24.4/25.4, in den Übungsgruppen

Koordinatoren: Luc Grosheintz, HG G 46, luc.grosheintz@sam.math.ethz.ch

Webpage: <http://metaphor.ethz.ch/x/2017/fs/401-1662-10L>

1. Gram-Schmidt-Verfahren und Householder Transformation

In der Vorlesung haben wir die Householder Transformation verwendet um die **QR-Zerlegung** einer Matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ zu bestimmen. Ein weiteres, sehr intuitives Verfahren, das sukzessive die Spalten $\underline{a}_1, \dots, \underline{a}_n$ mit $\underline{a}_i \in \mathbb{R}^m$, von \mathbf{A} orthogonalisiert, ist das Gram-Schmidt-Verfahren. Das Gram-Schmidt-Verfahren ist ein Standardwerkzeug in Beweisen der Linearen Algebra. Die folgenden Algorithmen (in Pseudo-Code) liefern eine **QR-Zerlegung** nach dem *Gram-Schmidt-Verfahren* und dem *modifizierten Gram-Schmidt-Verfahren*:

Gram-Schmidt:

```
for j = 1, ..., n do
  v_j = A_{:j}
  for i = 1, ..., j - 1 do
    R_{ij} = q_i^T a_j
    v_j = v_j - R_{ij} q_i
  end for
  R_{jj} = ||v_j||_2
  Q_{:j} = v_j / R_{jj}
end for
```

Modifiziertes Gram-Schmidt:

```
for i = 1, ..., n do
  V_{:i} = A_{:i}
end for
for i = 1, ..., n do
  R_{ii} = ||v_i||_2
  q_i = v_i / R_{ii}
  for j = i + 1, ..., n do
    R_{ij} = q_i^T v_j
    v_j = v_j - R_{ij} q_i
  end for
end for
```

- a) Implementieren Sie die beiden Gram-Schmidt-Verfahren in `1.ortho.py` und verwenden Sie beide Verfahren, um die **QR-Zerlegung** der Matrix $\mathbf{Z} \in \mathbb{R}^{50 \times 50}$ mit den Einträgen:

$$\mathbf{Z}_{ij} = 1 + \min(i, j), \quad 0 \leq i, j < 50$$

zu bestimmen.

- b) Vergleichen Sie die Güte der beiden Gram-Schmidt-Verfahren in Bezug auf die Orthogonalität der Spalten von \mathbf{Q} .
- c) Warum sind die Gram-Schmidt-Verfahren im Gegensatz zur Householder-Transformation (siehe `1.ortho.py`) ungeeignete numerische Methoden zur Berechnung von **QR-Zerlegungen**?

Bitte wenden!

2. Radioaktiver Zerfall

In einem Gefäß befinden sich n verschiedene Elemente Z_1, \dots, Z_n . Zum Zeitpunkt t sei $M_k(t)$ die Menge von Element Z_k . Die Elemente seien radioaktiv und die Zerfallsprodukte zerfallen selbst nicht weiter. Die Zerfallskonstanten $\lambda_1, \dots, \lambda_n$ sind gegeben. Zu m ($m \geq n$) Zeiten t_j erfolgt eine Messung der Aktivität $G(t_j)$.

Folgende physikalische Gesetze werden angenommen:

1. Zerfallsgesetz: $M_i(t) = M_i(0) \exp(-\lambda_i t)$, $t \geq 0$
2. Gesamtaktivität: $G(t) = \sum_{i=1}^n G_i(t) = \sum_{i=1}^n \lambda_i M_i(t)$

Formulieren Sie ein Ausgleichsproblem zur Bestimmung von $M_1(0), \dots, M_n(0)$.

Wählen Sie verschiedene n , Stoffmengen $M_k(0) \in [100, 500]$ und Zerfallsraten $\lambda_k \in [10^{-2}, 10^{-1}]$. Berechnen Sie die exakte Gesamtaktivität für verschiedene Zeitpunkte t_i . Erstellen Sie künstliche Messdaten, indem Sie $G(t_i)$ mit einem Messfehler versehen, auch hier sollten Sie verschieden starke Messfehler ausprobieren. Lösen Sie das Ausgleichsproblem für die jeweils gewählten Parameter. Was beobachten Sie?

3. Die Normalgleichungen sind schlecht konditioniert

Wir betrachten die Matrix:

$$\mathbf{A} = \begin{pmatrix} 1 + \varepsilon & 1 \\ 1 - \varepsilon & 1 \\ \varepsilon & \varepsilon \end{pmatrix}. \quad (1)$$

In exakter Arithmetik ist die Normalgleichung:

$$\mathbf{A}^T \mathbf{A} \underline{x} = \mathbf{A}^T \underline{b} \quad (2)$$

äquivalent zu

$$\mathbf{B}_\alpha \begin{pmatrix} r \\ \underline{x} \end{pmatrix} := \begin{pmatrix} -\alpha \mathbf{I} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} r \\ \underline{x} \end{pmatrix} = \begin{pmatrix} \underline{b} \\ \underline{0} \end{pmatrix}. \quad (3)$$

Schreiben Sie ein Python-Skript, das die Kondition von \mathbf{A} , $\mathbf{A}^T \mathbf{A}$, \mathbf{B}_1 und \mathbf{B}_α mit $\alpha = \varepsilon \|\mathbf{A}\|_2 / \sqrt{2}$ für $10^{-5} < \varepsilon < 1$ plottet. Das Python-Modul `numpy.linalg` hat eine Funktion `cond`.

Bitte wenden!

4. Kernaufgabe: Adaptive Methoden für steife Systeme

Aufgabenstellung

- a) Implementieren Sie die Rosenbrock-Wanner Methoden der Ordnung 2 und 3. Es sollen Funktionen `row_2_step(f, Jf, yi, h)` und `row_3_step(f, Jf, yi, h)` geschrieben werden, die ausgehend vom Wert $y_i(t_i)$ genau einen Zeitschritt h der entsprechenden Methode berechnen und die Propagierte $y_{i+1}(t_i + h)$ zurück geben.

Hinweis: Die Parameter sind im Template `stiff_row_Template.py` erklärt.

- b) Lösen Sie die logistische Differentialgleichung:

$$\dot{y}(t) = \lambda y(t)(1 - y(t))$$

mit dem Anfangswert $y(0) = c = 0.01$ und $\lambda = 25$ bis zum Zeitpunkt $T = 2$. Benutzen Sie $N = 100$ Zeitschritte. Plotten Sie die numerischen Lösungen $y(t)_{\text{ROW}}$ sowie die Fehler $y(t)_{\text{ROW}} - y(t)$ beider Methoden gegen die Zeit. Wie gross kann λ sein, bevor der Fehler der ROW-2 Methode einen maximalen Wert von 0.05 überschreitet?

- c) Messen Sie die Konvergenzordnung beider Methoden. Benutzen Sie hierfür obige Gleichung und Anfangswerte mit $\lambda = 10$. Wählen Sie $N = [2^4, \dots, 2^{12}]$ und berechnen Sie den Fehler zum Endzeitpunkt $T = 2$ gegenüber der exakten Lösung:

$$y(t) = \frac{ce^{\lambda t}}{1 - c + ce^{\lambda t}}$$

Plotten Sie den Fehler gegen die Anzahl Schritte doppelt logarithmisch.

- d) Implementieren Sie eine adaptive Strategie basierend auf den ROW-2 und ROW-3 Methoden. Verwenden Sie als Fehlerschätzer die Norm:

$$\varepsilon_i := \|y(t_i)_{\text{ROW-2}} - y(t_i)_{\text{ROW-3}}\|_2$$

Wählen Sie den initialen Zeitschritt als $h_0 = T/(100(\|f(y_0)\|_2 + 0.1))$ und passen Sie die Grösse des nächsten Zeitschritts durch Verkleinern ($h_{j+1} = \frac{h_j}{2}$) oder Vergrössern ($h_{j+1} = 1.1h_j$) an.

- e) Testen Sie die Implementation wiederum an der logistischen Differentialgleichung mit $\lambda = 50$. Wie viele Zeitschritte werden insgesamt zur Lösung benötigt? Plotten Sie die numerische Lösung $y(t)_{\text{ADA}}$ sowie die Fehler $y(t)_{\text{ADA}} - y(t)$ gegen die Zeit.

- f) Lösen Sie das folgende gekoppelte System:

$$\dot{y}_0(t) = -76 y_0(t) - 25\sqrt{3} y_1(t)$$

$$\dot{y}_1(t) = -25\sqrt{3} y_0(t) - 26 y_1(t)$$

mit Anfangswerten $y_0(0) = 1$ und $y_1(0) = 1$ bis zum Zeitpunkt $T = 1$ mit dem adaptiven Verfahren und einer Anfangsschrittweite von $h = 0.1$, plotten Sie $y(t)$.

- g) Lösen Sie die folgende sehr steife Gleichung:

$$\dot{y}(t) = \lambda y^2(t)(1 - y^2(t))$$

mit dem Anfangswert $y(0) = 0.01$ und $\lambda = 500$. Plotten Sie die numerische Lösung $y(t)_{\text{ADA}}$ sowie die Grösse der Zeitschritte gegen die Zeit. Wie viele Zeitschritte benötigt dieses Verfahren und was ist der kleinste Zeitschritt? Wie viele Zeitschritte dieser Grösse würde ein nicht-adapives Verfahren benötigen?