D-MATH
Dr. Vasile Gradinaru
Dr. Adrian Montgomery Ruf

Serie 9

Best before: Di. 05.05. / Mi. 06.05, in den Übungsgruppen

Koordinatoren: Adrian Montgomery Ruf, HG G 54.1, `adrian.ruf@sam.math.ethz.ch`

Webpage: http://metaphor.ethz.ch/x/2020/fs/401-1662-10L/#exercises

1. 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 $y(t)_{\texttt{ROW}}$ sowie die Fehler $y(t)_{\texttt{ROW}} - y(t)$ beider Methoden gegen die Zeit. Wie gross kann λ sein, bevor der Fehler der `ROW-2` Methode einen maximalen Wert von 0.05 überschreitet? Verwenden Sie zudem die Methode `solve_ivp` aus `scipy.integrate` und vergleichen Sie Ihre numerische Lösung und deren Approximationsfehler mit dem von `solve_ivp`. Benutzen Sie für `solve_ivp` die Methoden 'RK45' (default), 'Radau', 'BDF' und 'LSODA'.

c) Messen Sie die Konvergenzordnung beider Methoden. Benutzen Sie hierfür obige Gleichung und Anfangswerte mit $\lambda = 10$. Wählen Sie $N = [2^4, \ldots, 2^{12}]$ 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:

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

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 ($h_{j+1} = 1.1 h_j$) an.

Bitte wenden!

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 $y(t)_{\texttt{ADA}}$ sowie die Fehler $y(t)_{\texttt{ADA}} - y(t)$ 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 $y(t)_{\texttt{ADA}}$ 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?

2. *Broyden Verfahren*

Schreiben Sie Code, der die Plots aus dem Beispiel 3.9.2 (Broyden-Quasi-Newton-Verfahren: Konvergenz) und Beispiel 3.9.5 (Broyden-Verfahren für ein grosses nicht-lineares System), reproduziert.
Implementieren Sie zusätzlich auch die teurere Version des Broyden-Verfahren und plotten Sie diese ebenfalls.

Hinweis: Diese Aufgabe besitzt kein Template.