

Differentialgleichungen

Themen der Aufgabenstellung:

- Realisierung des Euler-Verfahrens (explisit/implizit)
- Realisierung des Runge-Kutta-Verfahren
- steife Differentialgleichungen (stiff equations)
- nichtlineare Differentialgleichungen (Van-der-Pol-DGL)
- chaotische Systeme (Lorenz-Attraktor)
- Einstieg in MATLAB/Simulink
- Einstieg in Jupyter Notebooks mit IPython

Aufgabenstellung:

1) Lösung "steifer Differentialgleichungen" mit Euler/Runge-Kutta (RK 2. Ordng.)

Es ist ein IPython-Programm zu schreiben, welches die folgende Differentialgleichung realisiert (unabh. Variable: x):

$$y' = 10 - 500 \cdot y + 5000 \cdot x \quad y(0) = 1$$

Versuchsvorbereitung:

- Geben Sie das Analogrechner-/Simulink-Schaltbild an.
- Geben Sie die Iterationsgleichungen für das **Euler**-Verfahren an.
- Geben Sie die Iterationsgleichungen für das **RK2**-Verfahren an.
- Geben Sie die Iterationsgleichungen für das **implizite Euler**-Verfahren an.
- Schreiben Sie ein IPython-Programm "*StiffDGL*" welches die DGL mit allen Verfahren löst und zusammen mit der analytischen Lösung (s.u.) in einem Plot anzeigt.

Die analytische Lösung der Differentialgleichung lautet:

$$y = 10 \cdot x + e^{-500 \cdot x}$$

Geben Sie in einem zweiten Plot die Abweichungen von der richtigen Lösung aus.

Versuchsdurchführung:

- Starten Sie das Programm mit $h=0.001$, $x_{\text{End}}=0.2$.
- Starten Sie das Programm mit $h=0.003$, $x_{\text{End}}=0.2$.
- Starten Sie das Programm mit $h=0.004$, $x_{\text{End}}=0.2$.
- Starten Sie das Programm mit $h=0.005$, $x_{\text{End}}=0.2$.

Wie ist das Ergebnis zu interpretieren?

Differentialgleichungen

2) Lösung einer (nichtlinearen) DGL 2. Ordnung (Van-der-Pol-DGL) mit RK 2

Es ist ein IPython-Programm zu schreiben, welches die folgende Differentialgleichung realisiert (unabh. Variable: t):

$$\ddot{y} = 6 \cdot (1 - y^2) \cdot \dot{y} - y \quad y(0) = 0, \quad \dot{y}(0) = 1$$

Versuchsvorbereitung:

- Geben Sie das Analogrechner-/Simulink-Schaltbild an.
- Geben Sie die DGL 2. Ordnung als 2 DGLn 1. Ordnung an.
- Geben Sie die Iterationsgleichungen für das **Euler**-Verfahren an.
- Geben Sie die Iterationsgleichungen für das **RK2**-Verfahren an.
- Schreiben Sie ein IPython-Programm "VanDerPol", welches die DGL mit beiden Verfahren löst und in einem Plot anzeigt.

Versuchsdurchführung:

- Starten Sie das Programm mit $h=0.001$, $t_{\text{End}}=31$.
- Starten Sie das Programm mit $h=0.02$, $t_{\text{End}}=31$.

3) Lösung eines Differentialgleichungssystems (Lorenz-Attraktor) mit RK 2

Es ist ein IPython-Programm zu schreiben, welches das folgende (chaotische) Differentialgleichungssystem realisiert (unabh. Variable: t):

$$\begin{aligned}\dot{x} &= -10 \cdot (x - y) & x(0) &= 0.01, & y(0) &= 0.01, & z(0) &= 0.0 \\ \dot{y} &= (40 - z) \cdot x - y \\ \dot{z} &= x \cdot y - 2.67 \cdot z\end{aligned}$$

Versuchsvorbereitung:

- Geben Sie die Iterationsgleichungen für das **RK2**-Verfahren an.
- Schreiben Sie ein IPython-Programm "Lorenz", welches das DGL-System löst.
Geben Sie im 1. Plot die Funktion $x(t)$ aus:
Geben Sie im 2. Plot $z(x)$ aus.
- Realisieren Sie das Differentialgleichungssystem mit **MATLAB/Simulink**
(Tipp: Funktionen mit Embedded-Matlab-Funktionsblöcken realisieren,
Ausgabe mit toWorkspace, Plotten in Matlab mit plot3(x,y,z)).

Differentialgleichungen

Versuchsdurchführung:

- a) Starten Sie das Programm mit $h=0.002$, $t_{\text{End}}=120$.
- b) Ändern Sie jetzt in der 2. Gleichung die 40 auf 40.000000001.
Vergleichen Sie $x(t)$ der beiden Simulationen, indem Sie das Ergebnis in einem Plot darstellen.

Vorzubereiten:

Analogrechner-Schaltbilder
Ableitung der Iterationsgleichungen.

Aufgabenbearbeitung:

Fertigzustellen und abzugeben sind

- Analogrechner/Simulink-Schaltbilder (nachvollziehbare Skizzen)
- Iterationsgleichungen (in nachvollziehbarer Form)
- IPython-Programme (in Jupyter-Notebooks)
- Nachvollziehbares Protokoll der Ergebnisse, mit
 - Zweck des Versuchs,
 - Versuchsrandbedingungen,
 - Funktionsausdrucken (kommentiert),
 - kurze Bewertung der Ergebnisse.

Anmerkung: Einrichten von Jupyter-Notebooks bei azure:

1. Folgende Link aufrufen: <https://notebooks.azure.com/>
2. Kostenfreien Account einrichten.
3. Folgende Link in einem zweiten Browserfenster aufrufen:
<https://notebooks.azure.com/LearnAndy/libraries/HAW-MDP>
4. Library clonen.
5. Beispielprogramme ansehen.
6. Eigene Notebooks (wie in Versuchsbeschreibung beschrieben) anlegen und realisieren.