

Simulationstechnik

Jiaqi Lan 13. September 2025

Inhaltsverzeichnis

1	Einführung zur Simulationstechnik	2
1.1	Algebraisches Gleichungssystem (AE)	2
1.2	Differentiales Gleichungssystem (ODE)	2
1.3	Differential-algebraisches Gleichungssystem (DAE)	3
1.4	Lösungsmethode	3
2	Numerische Verfahren von gewöhnlichen Differentialgleichungen	4
2.1	Modellanalyse von gewöhnlichen Differentialgleichungen	6
2.1.1	Stationäre Lösung (Lösen eines Systems algebraischer Gleichungen) . . .	6
2.1.2	Linearisierung am stationären Punkt	8
2.1.3	Analytische Lösung eines Systems linearer Differentialgleichungen . . .	8
2.2	Numerische Verfahren	12
2.2.1	Numerische Fehleranalyse algebraischer Gleichungssysteme	12
2.2.2	Numerische Stabilität für Iteration	15
2.2.3	Numerische Lösung eines Systems gewöhnlicher Differentialgleichungen	16
2.2.4	Numerische Stabilität numerischer Integration linearer Differentialgleichungen	19
2.2.5	Numerische Lösung differential-algebraischer Gleichungen	21
3	Prüfungsvorbereitungsaufgaben	23

1 Einführung zur Simulationstechnik

Simulationstechnik kann wesentlich als eine Methodik bezeichnet werden, mit der man das Problem des **dynamischen Systems** mithilfe unterschiedlicher **numerischer Verfahren** lösen und analysieren kann.

In Mathematik und Physik bezeichnet ein dynamisches System ein Modell, das beschreibt, wie sich ein System im Laufe der Zeit entwickelt. Es besteht aus **Zustandsvariablen** und einer **Entwicklungsvorschrift** (oft in Form von Gleichungen). Formal unterscheiden sie sich von **kontinuierlichem dynamischen System** und **diskretem dynamischen System**:

Inzwischen enthält ein dynamisches System 4 zentrale Elemente:

- **Zustandsraum** (state space): die Menge aller möglichen Zustände
- **Entwicklungsgesetz**: gegeben durch **Differentialgleichungssysteme** oder Iterationsabbildungen
- **Bahnen/Phasentrajektorien**: die Trajektorien, die das System ausgehend von einem Anfangszustand im Zeitverlauf beschreibt
- **Gleichgewichtspunkte, Grenzyklen, Attraktoren**

Numerische Verfahren sind Lösungsmethoden für die meisten komplizierten Differentialgleichungssysteme. Vor allem nutzt man **MATLAB** und **Simulink**, um das Modell zu analysieren.

1.1 Algebraisches Gleichungssystem (AE)

In algebraischen Gleichungen treten die Unbekannten nur durch algebraische Operationen (Addition, Subtraktion, Multiplikation, Division, Potenzierung) auf und beinhalten weder Ableitungen noch Integrale.

1.2 Differenciales Gleichungssystem (ODE)

In Differentialgleichungen ist die Unbekannte eine Funktion, und in der Gleichung treten die Ableitungen dieser Funktion auf. In Differentialgleichungen entwickeln sich alle Variablen durch Ableitungen. Differentialgleichungssysteme umfassen 4 zentrale Kennzeichen:

- **Anzahl der unabhängigen Variablen:**
Gewöhnliche Differentialgleichung (ODE, ordinary differential equation, einer unabhängigen Variablen)
Partielle Differentialgleichung (PDE, partial differential equation, mehrere unabhängige Variablen)

- **Linearität:**
Lineare Differentialgleichung (die unbekannte Funktion und ihre Ableitungen treten **nur in der ersten Potenz** auf und werden nicht miteinander multipliziert)
Nichtlineare Differentialgleichung
- **Ordnung** (die Ordnung der höchsten Ableitung)
- **Homogenität:**
Homogen (wenn die rechte Seite null ist, das heißt, keine unabhängigen Terme)
Inhomogen

1.3 Differential-algebraisches Gleichungssystem (DAE)

In differential-algebraischen Gleichungen entwickeln sich einige Variablen durch Ableitungen, während andere Variablen algebraischen Beschränkungen unterliegen.

1.4 Lösungsmethode

Lösungsverfahren umfassen 2 Verfahren, jeweils **analytische Verfahren** (exakte Darstellung der Lösung mit Formeln, elementaren oder speziellen Funktionen) und **numerische Verfahren** (Näherungslösungen durch Computer oder iterative Algorithmen). In der Simulationstechnik werden numerische Verfahren hauptsächlich untersucht.

Numerische Verfahren (Iterationsverfahren) umfassen 3 Verfahren:

- **Einschrittverfahren** (Runge-Kutta-Verfahren):
Euler-Vorwärts
Euler-Rückwärts
Mittelwert
Reun
- **Mehrschrittverfahren**
- **Extrapolationsverfahren**

2 Numerische Verfahren von gewöhnlichen Differentialgleichungen

Gewöhnliche Differentialgleichungen (ODE) können durch die unten genannte Gleichung mit Anfangsbedingungen definiert werden:

$$\dot{\mathbf{x}} = F(\mathbf{x}, t) \quad \text{mit} \quad \mathbf{x}_0 = \mathbf{x}(t_0) \in \mathbb{R}^n, t \in \mathbb{R}^+ \quad (1)$$

Wenn diese Gleichungen **lineare gewöhnliche Differentialgleichungen** sind, können sie in folgender Form dargestellt werden:

$$\dot{\mathbf{x}} = A\mathbf{x} + Bu(t) \quad \text{mit} \quad \mathbf{x}_0 = \mathbf{x}(t_0) \in \mathbb{R}^n, t \in \mathbb{R}^+ \quad (2)$$

Es ist in der Regel notwendig, Systeme höherer linearer Differentialgleichungen in Systeme erster linearer Differentialgleichungen zu reduzieren:

Dgln. in **n**. Ordnung + **n**. ABu. \leftrightarrow **n**. Dgln. in **1**. Ordnung + **n**. ABu.

Beispiel: Lineare Differentialgleichungen reduzieren (1)

Originale Differentialgleichung:

$$\ddot{y} = -2\delta\omega_0\dot{y} - \omega_0^2 y + K\omega_0^2 u \quad \text{mit} \quad \text{ABu} : y_0 = y(t_0), y_1 = \dot{y}(t_0)$$

Zustandsvariablen definieren:

$$\begin{aligned} x_1 &= y \quad \text{mit} \quad \text{ABu} : x_1(t_0) = y_0 \\ x_2 &= \dot{y} \quad \text{mit} \quad \text{ABu} : x_2(t_0) = y_1 \end{aligned}$$

Umrechnen zur Simulationsgleichung:

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, u, t) : \quad \dot{x}_1 = \dot{y} = x_2 \\ \dot{x}_2 &= f_2(x_1, x_2, u, t) : \quad \dot{x}_2 = \ddot{y} = -2\delta\omega_0\dot{y} - \omega_0^2 y + K\omega_0^2 u = -2\delta\omega_0 x_2 - \omega_0^2 x_1 + K\omega_0^2 u \end{aligned}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\delta\omega_0 \end{bmatrix}}_A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ K\omega_0^2 \end{bmatrix}}_B u$$

Darstellung von der Matrix A :

$$A = J_f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (3)$$

Inzwischen definiert man die Funktion f_i mit $i \in 1, 2, \dots, n$ in folgender Form:

$$\dot{x}_i = f_i(x_1, x_2, \dots, x_n, u(t), t) \quad \text{bzw.} \quad \dot{x}_i = f_i(\mathbf{x}, t) \quad (4)$$

Beispiel: Lineare Differentialgleichungen reduzieren (2)

Originale Differentialgleichung:

$$\ddot{y} = a\dot{y} + by + c\ddot{u} + d\dot{u} + eu \quad \text{mit} \quad \text{ABu} : y_0 = y(t_0), y_1 = \dot{y}(t_0) \quad u_0 = u(t_0), u_1 = \dot{u}(t_0)$$

Sukzessive Integration zur 1. Ordnung u :

$$\begin{aligned} \int \ddot{y} \, dx &= \dot{y} = c\dot{u} + ay + du + \int (by + eu) \, dx \\ \int \dot{y} \, dx &= y = cu + \int (ay + du + \int (by + eu) \, dx) \, dx \end{aligned}$$

Zustandsvariablen definieren:

$$\begin{aligned} x_1 &= \int (by + eu) \, dx \quad \text{mit} \quad \text{ABu} : x_1(t_0) = y_1 - cu_1 - ay_0 - du_0 \\ x_2 &= \int (ay + du + x_1) \, dx \quad \text{mit} \quad \text{ABu} : x_2(t_0) = y_0 - cu_0 \\ y &= cu + x_2 \end{aligned}$$

Umrechnen zur Simulationsgleichung:

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, u, t) : \quad \dot{x}_1 = by + eu = bx_2 + (bc + e)u \\ \dot{x}_2 &= f_2(x_1, x_2, u, t) : \quad \dot{x}_2 = ay + du + x_1 = x_1 + ax_2 + (ac + d)u \end{aligned}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & b \\ 1 & a \end{bmatrix}}_A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} bc + e \\ ac + d \end{bmatrix}}_B u$$

Voraussetzungen: Existenz und Eindeutigkeit der Lösung $\mathbf{x}(t), t > 0$

- **Existenz:** $f(x)$ stetig
- **Eindeutigkeit:** $\|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\|$ mit $L < \infty$

Falls $f(x)$ stetig differenzierbar, dann sind beide Bedingungen erfüllt.

2.1 Modellanalyse von gewöhnlichen Differentialgleichungen

2.1.1 Stationäre Lösung (Lösen eines Systems algebraischer Gleichungen)

Definition zur stationären Lösung \mathbf{x}_s :

$$\dot{\mathbf{x}}_s = F(\mathbf{x}_s, t_s) = 0 \quad (5)$$

Obige Gleichung ist **algebraische Gleichung**. Numerische Bestimmung von \mathbf{x}_s durch (i) **exakte Elimination** oder (ii) **näherungsweise Lösung**. Wenn $f(\mathbf{x}_s, t_s) = 0$ nichtlineare algebraische Gleichungen vorliegen, kann die numerische Bestimmung \mathbf{x}_s durch **Newton-Verfahren** erfolgen. Wenn $f(\mathbf{x}_s, t_s) = 0$ lineare algebraische Gleichungen vorliegen, können sie in der folgenden Form dargestellt werden.

$$A\mathbf{x} = \mathbf{b} \quad \text{bzw.} \quad \mathbf{x} = A^{-1}\mathbf{b} \quad (6)$$

Stationäre Lösung für lineare algebraische Gleichungen

(i) Exakte Elimination: Gauß-Elimination

- Dreieckszerlegung von $A = LR$:

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 1 & & & 0 \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix}}_L \underbrace{\begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ 0 & & & r_{nn} \end{bmatrix}}_{\mathbf{z}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{\mathbf{b}}$$

Multiplikationszahlen: $\sum_{i=1}^{n-1} i^2 \simeq n^3/3$

- Vorwärtssubstitution ($\mathbf{z} = L^{-1}\mathbf{b}$):

$$\underbrace{\begin{bmatrix} 1 & & & 0 \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix}}_L \underbrace{\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}}_{\mathbf{z}} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{\mathbf{b}}$$

Multiplikationszahlen: $\sum_{i=1}^{n-1} i \simeq n^2/2$

- Rückwärtssubstitution ($\mathbf{x} = R^{-1}\mathbf{z}$):

$$\underbrace{\begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ 0 & & & r_{nn} \end{bmatrix}}_R \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}}_{\mathbf{z}}$$

Multiplikationszahlen: $\sum_{i=1}^{n-1} i \simeq n^2/2$

Gesamtmultiplikationszahlen: $n^2 + n^3/3 < (n+1)^n n!$ (Cramer-Regel)

(ii) Näherungsweise Lösung: Iterationsverfahren

$$\mathbf{x}_{i+1} = \Phi(\mathbf{x}_i) \quad \text{mit} \quad i = 0, 1, \dots, m \quad \mathbf{x}_0 = \text{Startwert} \quad (7)$$

Zum Beispiel die Fixpunktsgleichung (Richardson-Verfahren):

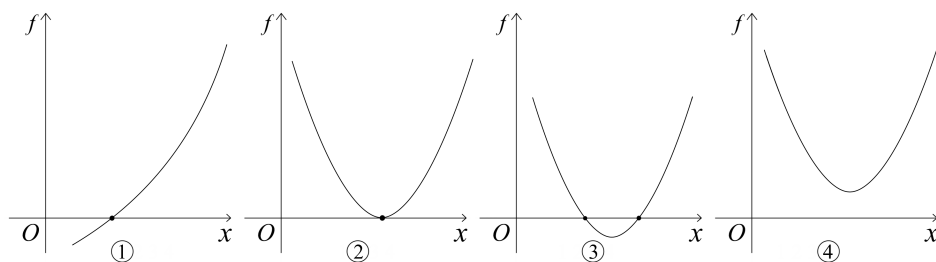
$$A\mathbf{x} = \mathbf{b} \Leftrightarrow \mathbf{x} = (\mathbf{I}_n - A)\mathbf{x} + \mathbf{b} \Leftrightarrow \mathbf{x}_{n+1} = (\mathbf{I}_n - A)\mathbf{x}_n + \mathbf{b} \quad (8)$$

Eine detailliertere Analyse befindet sich in Abschnitt 2.2.1.

Stationäre Lösung für nichtlineare algebraische Gleichungen

Lösungsarten:

- 1 Einfache reelle Lösung ($\det(\frac{\partial F}{\partial \mathbf{x}}|_{\mathbf{x}_s}) \neq 0$)
- 2 k-fache reelle Lösung ($\frac{\partial F}{\partial \mathbf{x}}|_{\mathbf{x}_s} = \begin{cases} 0, i=1,2,\dots,k-1 \\ \neq 0, i=k \end{cases}$)
- 3 Zwei reelle Lösungen (von Startwert und von Parameter abhängig)
- 4 Keine reelle Lösung



(ii) Näherungsweise Lösung: Newton-Verfahren

Lösung der Aufgabe mittels Taylor-Reihe in der Umgebung von \mathbf{x}_0 :

$$F(\mathbf{x}) = F(\mathbf{x}_0) + \frac{\partial F}{\partial \mathbf{x}} \bigg|_{\mathbf{x}_0} (\mathbf{x} - \mathbf{x}_0) = 0 \quad (9)$$

$$\mathbf{x}_1 = \mathbf{x}_0 - \left(\frac{\partial F}{\partial \mathbf{x}} \bigg|_{\mathbf{x}_0} \right)^{-1} f(\mathbf{x}_0) \quad (10)$$

$$\text{Iterationsvorschrift : } \mathbf{x}_{i+1} = \mathbf{x}_i - \left(\frac{\partial F}{\partial \mathbf{x}} \bigg|_{\mathbf{x}_i} \right)^{-1} f(\mathbf{x}_i) \quad \text{mit} \quad i = 0, 1, \dots, m \quad (11)$$

Eine detailliertere Analyse befindet sich in Abschnitt 2.2.1.

2.1.2 Linearisierung am stationären Punkt

Wenn man nichtlineare gewöhnliche Differentialgleichungen analysiert, wird sie an dem stationären Punkt linearisiert werden. Betrachtung in der Umgebung der stationären Lösung:

$$\mathbf{x} = \mathbf{x}_s + \Delta \mathbf{x} \quad (12)$$

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_s + \Delta \dot{\mathbf{x}} = 0 + \Delta \dot{\mathbf{x}} \quad (13)$$

$$\dot{\mathbf{x}} = F(\mathbf{x}_s + \Delta \mathbf{x}) \stackrel{1. \text{ Taylor am } \mathbf{x}_s}{=} F(\mathbf{x}_s) + \underbrace{\frac{\partial F}{\partial \mathbf{x}} \bigg|_{\mathbf{x}=\mathbf{x}_s}}_A \Delta \mathbf{x} + O_2(\Delta \mathbf{x}) \quad (14)$$

$$\Delta \dot{\mathbf{x}} = A(\mathbf{x}_s) \Delta \mathbf{x} \quad \text{mit} \quad \Delta \mathbf{x}_0 = \mathbf{x}_0 - \mathbf{x}_s \quad (15)$$

Eigenschaften von Jacobimatrix $A(\mathbf{x}_s)$: dense (mehrere Element $\neq 0$) und dünn (seltene Element $\neq 0$) besetzte Matrizen.

2.1.3 Analytische Lösung eines Systems linearer Differentialgleichungen

Gl. 15 ist das homogene System linearer Differentialgleichungen. Allgemeine Lösung kann durch die folgende Gleichung dargestellt werden:

$$\Delta \mathbf{x}(t) = e^{At} \Delta \mathbf{x}_0, \quad e^{At} = \sum_{k=0}^{\infty} \frac{(At)^k}{k!} \quad (16)$$

wobei $A \in \mathbb{R}^{n \times n}$ eine konstante Matrix ist. Da A^k sehr schwer zu berechnen ist, wird es zuerst diagonalisiert.

Darstellung mit Eigenwerten und Eigenvektoren: Angenommen, A ist diagonalisierbar:

$$A = V \Lambda V^{-1}, \quad A^k = V \Lambda^k V^{-1} \quad (17)$$

wobei $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ die Eigenwerte enthält, $V = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$ die Eigenvektorenmatrix ist. Eigenwerte λ_i können durch $\det(A - \lambda \mathbf{I}) = 0$ berechnet werden. Eigenvektorenmatrix \mathbf{v}_i können durch $(A - \lambda_i \mathbf{I}) \mathbf{v}_i = 0$ berechnet werden.

Dann gilt:

$$e^{At} = V e^{\Lambda t} V^{-1}, \quad e^{\Lambda t} = \text{diag}(e^{\lambda_1 t}, e^{\lambda_2 t}, \dots, e^{\lambda_n t}) \quad (18)$$

$$\Delta \mathbf{x}(t) = \sum_{i=1}^n c_i e^{\lambda_i t} \mathbf{v}_i \quad (19)$$

$$\mathbf{c} = V^{-1} \Delta \mathbf{x}_0 \quad (20)$$

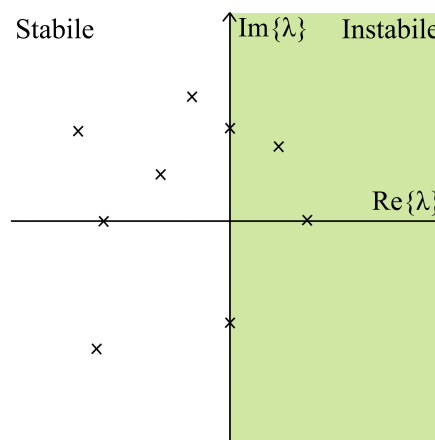
wobei die Koeffizienten c_i durch die Anfangsbedingung $\Delta \mathbf{x}(0) = \Delta \mathbf{x}_0$ bestimmt werden wie Gl. 20. Besondere Fälle:

- Mehrfacher Eigenwert: Man benötigt verallgemeinerte Eigenvektoren, und die Lösung kann Terme wie $te^{\lambda t}$ enthalten.
- Komplexe Eigenwerte: Die Lösung enthält Sinus- und Kosinusfunktionen:

$$e^{(\alpha \pm i\beta)t} = e^{\alpha t}(\cos \beta t \pm i \sin \beta t) \quad (21)$$

Eigenwerte analysieren

In einem komplexen Koordinatensystem werden die Eigenwerte markiert. Wenn $\lambda_i < 0$ ist, handelt es sich um eine stabile Lösung. Enthalten die Eigenwerte einen imaginären Teil, handelt es sich um eine periodische Lösung. Wie in der folgenden Abbildung dargestellt.



Bestimmung von Simulationszeit

- Zeitkonstante oder Periode:

$$T_i = \min \left\{ \frac{1}{|\operatorname{Re}\{\lambda_i\}|}, \frac{2\pi}{|\operatorname{Im}\{\lambda_i\}|} \right\} \quad (22)$$

$$T_{\max} = \max_{i=1,2,\dots,n} \{T_i\} \quad (23)$$

- Steifigkeitsmaß:

$$\gamma = \frac{T_{\max}}{T_{\min}} > 10^3 \sim 10^7 \quad (24)$$

- Simulationszeit:

$$t_{\text{sim}} = 5T_{\max} \quad (25)$$

- Zeitschrittweite:

$$h_{\text{Numerik}} = \min \left\{ \frac{T_{\min}}{10}, \frac{t_{\text{sim}}}{200} \right\} \quad (26)$$

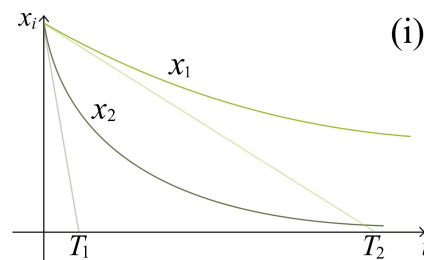
$$h_{\text{Graphik}} = \frac{t_{\text{sim}}}{200} \quad (27)$$

- Rechenzeit:

$$\frac{t_{\text{sim}}}{h_{\text{Numerik}}} = \frac{5T_{\max}}{T_{\min}/10} = 50\gamma > 50 \cdot 10^3 \quad (28)$$

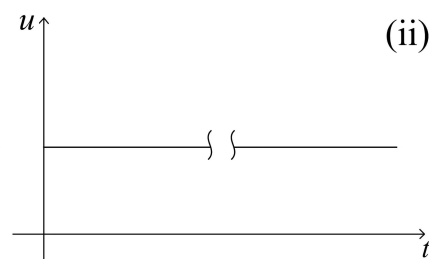
Problem (i): Steife Differentialgleichungen:

Eigenschaft: weit liegende Zeitkonstanten T_i bzw. sehr großer Unterschied zwischen T_{\max} und T_{\min} , $\gamma \gg 1$. Infolgedessen ist die Rechenzeit zu groß.



Abhilfe: Variable Zeitschrittweite h (graduellerweise die Zeitschrittweite).

Problem (ii): Unstetige $u(t)$ oder $f(\mathbf{x})$:



Abhilfe: Erkennen der Unstetigkeitsstelle und Versuchen, stückweise numerische Lösung mit neuen Anfangsbedingungen.

Beispiel: Analytische Lösung eines 2×2 Systems linearer Differentialgleichungen

Betrachten wir das System:

$$\dot{\mathbf{x}} = A\mathbf{x}, \quad A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{x}(0) = \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

Eigenwerte berechnen

Die Eigenwerte λ erfüllen:

$$\begin{aligned} \det(A - \lambda I) &= 0 \\ \det \begin{bmatrix} 2 - \lambda & 1 \\ 1 & 2 - \lambda \end{bmatrix} &= (2 - \lambda)^2 - 1 = 0 \\ \Rightarrow \lambda_1 &= 1, \quad \lambda_2 = 3 \end{aligned}$$

Eigenvektoren bestimmen

- Für $\lambda_1 = 1$:

$$(A - I)\mathbf{v} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \mathbf{v} = 0 \Rightarrow \mathbf{v}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

- Für $\lambda_2 = 3$:

$$(A - 3I)\mathbf{v} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \mathbf{v} = 0 \Rightarrow \mathbf{v}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Eigenvektormatrix

$$V = [\mathbf{v}_1, \mathbf{v}_2] = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

Koeffizienten c_i berechnen

Aus der Anfangsbedingung $\mathbf{x}_0 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ folgt: $\mathbf{c} = V^{-1}\mathbf{x}_0$

$$\det(V) = 1 \cdot 1 - (-1) \cdot 1 = 2, \quad V^{-1} = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

$$\mathbf{c} = V^{-1}\mathbf{x}_0 = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Analytische Lösung

$$\mathbf{x}(t) = c_1 e^{\lambda_1 t} \mathbf{v}_1 + c_2 e^{\lambda_2 t} \mathbf{v}_2 = 1 \cdot e^t \begin{bmatrix} 1 \\ -1 \end{bmatrix} + 2 \cdot e^{3t} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

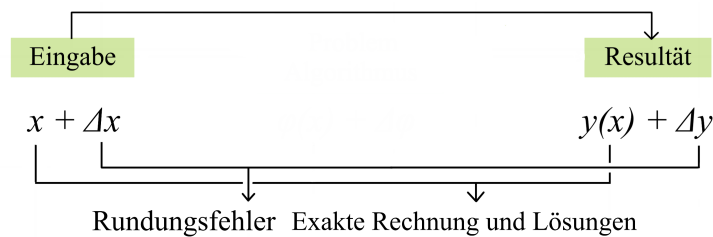
$$\mathbf{x}(t) = \begin{bmatrix} e^t + 2e^{3t} \\ -e^t + 2e^{3t} \end{bmatrix}$$

2.2 Numerische Verfahren

Das Wesen der numerischen Methoden besteht darin, durch **Approximation** und **Diskretisierung** kontinuierliche und möglicherweise nicht exakt lösbare Probleme in Näherungsrechnungen mit endlichen Schritten zu überführen und ihre Zuverlässigkeit durch **Fehleranalyse** sicherzustellen.

2.2.1 Numerische Fehleranalyse algebraischer Gleichungssysteme

Rundungsfehler

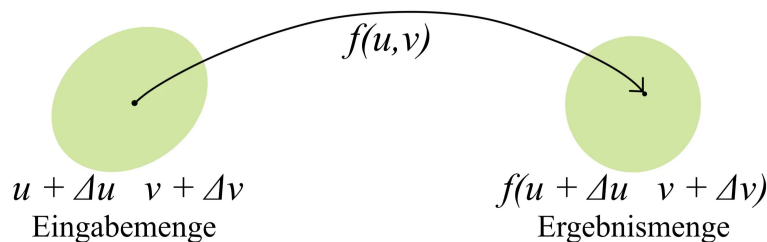


Dabei wird die numerische Lösung mit Rundungsfehlern dargestellt als:

$$x^* = \text{rd}\{x\} \quad f^* = \text{rd}\{f(x)\} \quad (29)$$

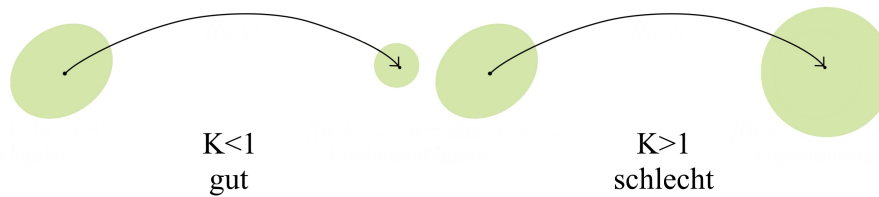
Relativer Fehler ist:

$$\delta_x = \left| \frac{x - x^*}{x} \right| \quad \delta_f = \left| \frac{f - f^*}{f} \right| \leq \text{eps (Genauigkeit)} \quad (30)$$



$$f(u + \Delta u, v + \Delta v) \stackrel{1. \text{ Taylor am } (u, v)}{=} f(u, v) + \underbrace{\left. \frac{\partial f}{\partial u} \right|_{(u, v)} \Delta u + \left. \frac{\partial f}{\partial v} \right|_{(u, v)} \Delta v}_{\text{Absolute Fehler}} \quad (31)$$

$$\delta_f = \frac{f(u + \Delta u, v + \Delta v) - f(u, v)}{f(u, v)} = u \frac{f_u}{f} \frac{\Delta u}{u} + v \frac{f_v}{f} \frac{\Delta v}{v} = \underbrace{u \frac{f_u}{f}}_{K_u} \delta_u + \underbrace{v \frac{f_v}{f}}_{K_v} \delta_v \quad (32)$$



Beurteilung: Problem $\begin{cases} \text{gut} \\ \text{schlecht} \end{cases}$ konditioniert $\Leftrightarrow |K_u \text{ oder } v| \begin{cases} < 1 \\ > 1 \end{cases}$ oder $|f_u \text{ oder } v| \begin{cases} \text{klein} \\ \text{gro\ss} \end{cases}$

Beispiel: Rundungsfehleranalyse linearer Differentialgleichung

Linearer Differentialgleichung:

$$\dot{y} = (\lambda + \Delta\lambda)y \quad \text{mit} \quad t > 0 \quad y(0) = y_0 + \Delta y_0$$

Lösung:

$$y(t) = (y_0 + \Delta y_0)e^{(\lambda + \Delta\lambda)t} = f(y_0 + \Delta y_0, \lambda + \Delta\lambda)$$

Entwickeln die obige Gleichung an y_0, λ in eine Taylor-Reihe erster Ordnung:

$$f(y_0 + \Delta y_0, \lambda + \Delta\lambda) = y_0 e^{\lambda t} + \underbrace{e^{\lambda t} \Delta y_0 + t y_0 e^{\lambda t} \Delta\lambda}_{\Delta f = \Delta y}$$

Kondition linearer Differentialgleichungen analysieren:

$$\lim_{t \rightarrow \infty} |f_{y_0} \text{ oder } f_\lambda| = \begin{cases} 0 & , \operatorname{Re}\{\lambda\} < 0 \quad \text{gut} \\ \infty & , \operatorname{Re}\{\lambda\} \geq 0 \quad \text{schlecht} \end{cases}$$

Verfahrensfehler

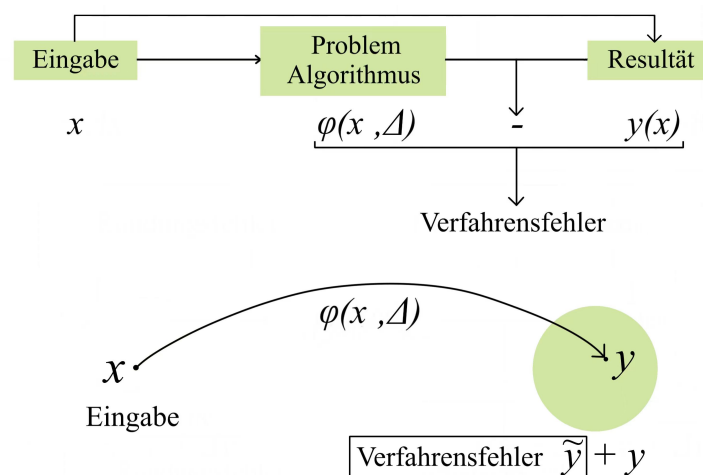
Der **Verfahrensfehler** ist der von der numerischen Methode selbst eingeführte Fehler und umfasst in der Regel den **Trunkierungsfehler** sowie mögliche andere methodenbedingte Fehler, wie zum Beispiel Fehler, die durch nicht konvergente Iterationen entstehen.

$$\varphi(x, \Delta) = \bar{y} \tag{33}$$

$$\tilde{y} = \varphi(x, \Delta) - f(x) \simeq O^P(\Delta) \simeq \Delta^P \tag{34}$$

wobei P die **Verfahrensordnung** ist und Δ numerische Parameter wie Zeitschrittweite ist. P kann auch die **Konsistenzordnung** von Verfahren dargestellt werden. Die Konsistenz stellt sicher, dass die Formel der numerischen Methode selbst eine korrekte Näherung der ursprünglichen Gleichung darstellt:

$$\lim_{\Delta \rightarrow 0} \varphi(x, \Delta) - f(x) = 0 \Leftrightarrow P \geq 1 \tag{35}$$



Beispiel: Verfahrensfehleranalyse Vorwärts-Eulen-Verfahren

Differentialgleichung:

$$\dot{y} = f(y) = y \quad \text{mit} \quad t > 0 \quad y(0) = y_0$$

Numerische Lösung Vorwärts-Eulen-Verfahren am $i = 0$ (Zeitschrittweite h):

$$\bar{y}_1 = \varphi(y_0, h) = y_0 + h f(y_0) = y_0(1 + h)$$

Exakte Lösung am $i = 0$:

$$y_1 = f(y_0) = y_0 e^h$$

Lokale Verfahrensfehler am $i = 0$:

$$\tilde{y}_1 = \bar{y}_1 - y_1 = y_0(1 + h - e^h)$$

Entwickeln das obige Exponentialglied in eine Taylor-Reihe:

$$\tilde{y}_1 = -y_0 \left(\frac{h^2}{2} + \frac{h^3}{6} + \dots \right) \simeq O^2(h) \simeq h^2$$

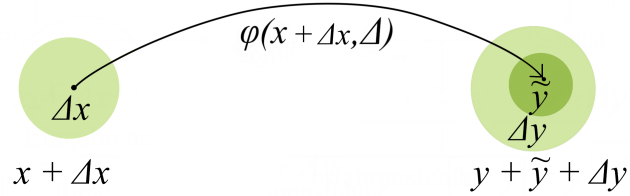
Lokale Verfahrensfehler (**Amplitudenfehler**) ist in $O^2(h)$. **Steigungsfehler:** $\tilde{f}_1 = \tilde{y}_1/h = O^2(h)/h = O^1(h)$

Globale Verfahrensfehler:

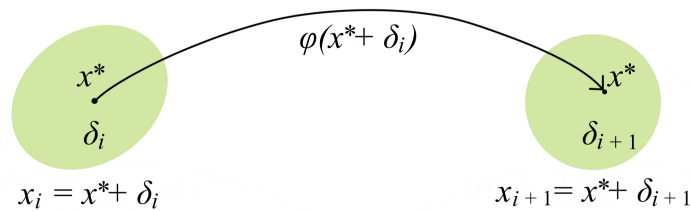
$$\tilde{y} = \sum_{k=1}^{\frac{t_{\text{sim}}}{h}} |\tilde{y}_k| = \sum_{k=1}^{\frac{t_{\text{sim}}}{h}} O^2(h) = O^1(h) \quad P = 1$$

Verfahrensordnung (Konsistenzordnung oder Fehler der Steigungsfehler) ist $P = 1$. Lokale Verfahrensordnung (Fehler der Amplitudenfehler) ist $P + 1 = 2$.

Kombination von Rundungsfehler und Verfahrensfehler ist wie in der folgenden Abbildung dargestellt:



2.2.2 Numerische Stabilität für Iteration



Iterationsfehler (Fixpunktsfehler):

$$\delta_i = x_i - x^* \quad \text{mit} \quad \varphi(x^*) = x^* \quad (36)$$

$$\varphi(x^* + \delta_i) = x^* + \delta_{i+1} \quad (37)$$

Konvergenz der Iteration. Entwickeln Gl. 37 in eine Taylor-Reihe am x^* :

$$\varphi(x^* + \delta_i) \stackrel{i \rightarrow \infty}{=} \varphi(x^*) + \left. \frac{d\varphi}{dx} \right|_{x^*} \delta_i + O^2(\delta_i) \quad \text{bzw.} \quad \delta_{i+1} \stackrel{i \rightarrow \infty}{=} O^1(\delta_i) \quad (38)$$

Fehlerverstärkung der Iteration:

$$\left| \frac{\delta_{i+1}}{\delta_i} \right| = \left| \left. \frac{d\varphi}{dx} \right|_{x^*} \right| = \begin{cases} < 1 & \text{stabil} \\ > 1 & \text{instabil} \\ = 1 & \text{grenzstabil} \end{cases} \quad (39)$$

Fehlerverstärkung für Newton-Verfahren:

$$x_{i+1} = \varphi(x_i) = x_i - \left(\left. \frac{df}{dx} \right|_{x_i} \right)^{-1} f(x_i) \quad \text{mit} \quad f(x^*) = 0 \quad (40)$$

$$\left| \left. \frac{d\varphi}{dx} \right|_{x^*} \right| = \left| \frac{f(x^*)f''(x^*)}{(f'(x^*))^2} \right| = 0 < 1 \quad (41)$$

Konvergenz für Newton-Verfahren:

$$\left. \frac{d^2 \varphi}{dx^2} \right|_{x^*} = \frac{f''(x^*)}{f'(x^*)} + \frac{f(x^*) f^{(3)}(x^*)}{(f'(x^*))^2} - \frac{2f(x^*) (f''(x^*))^2}{(f'(x^*))^3} = \frac{f''(x^*)}{f'(x^*)} \quad (42)$$

$$\delta_{i+1} \stackrel{i \rightarrow \infty}{\approx} O^2(\delta_i) \quad (43)$$

Beispiel: Quadratwurzel berechnen mit Picard-Iteration

Aufgabe:

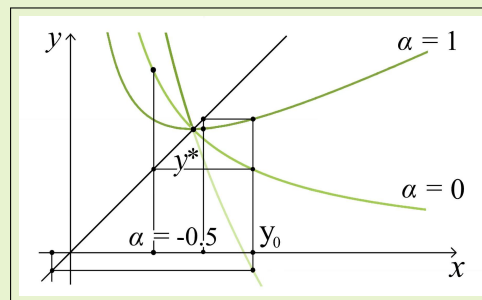
$$y^2 = 1 \quad y^* = \pm 1$$

Iterationsgleichung:

$$y_{i+1} = \varphi(y_i) = \frac{1}{1 + \alpha} \left(\alpha y_i + \frac{1}{y_i} \right)$$

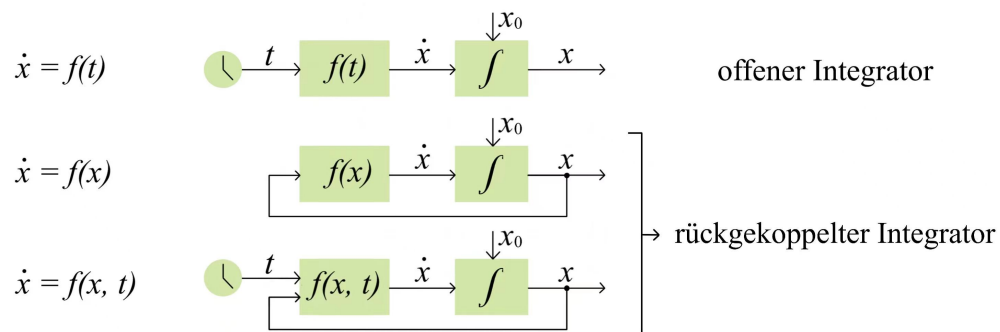
Fehlersverstärkung:

$$\left| \frac{d\varphi}{dx} \right|_{x^*} = \left| \frac{\alpha - 1}{\alpha + 1} \right| \quad \text{bedingt stabil mit} \quad \begin{cases} \alpha < 0, & \text{instabil} \\ \alpha = 0, & \text{granzstabil} \\ \alpha > 0, & \text{stabil} \end{cases}$$



2.2.3 Numerische Lösung eines Systems gewöhnlicher Differentialgleichungen

Integratorenformen (zeitkritisch oder nicht und zustandsvariablenabhängig oder nicht)



Einschritt-digitaler Integrator (Einschritt-numerische Integration)

Zeitdiskretisierung:

$$t_k = kh \quad \text{mit} \quad k = 0, 1, 2, \dots, \frac{t_{\text{sim}}}{h} \quad x(t_k) = x_k \quad (44)$$

Analytische Integration zu numerischer Integration:

$$x_{k+1} = x_k + \int_{t_k}^{t_{k+1}} f(\tau) d\tau \simeq x_k + h\varphi(\mathbf{s}_k) \quad (45)$$

Einschritt-numerische Integration (nur von t_k und t_{k+1} abhängig):

$$x_{k+1} = x_k + h \sum_{i=1}^m w_i s_i \quad k = 0, 1, 2, \dots, \frac{t_{\text{sim}}}{h} \quad \text{mit} \quad \sum_i^m w_i = 1 \quad (46)$$

$$s_i = f\left(x_k + h \sum_{j=1}^m a_{ij} s_j, t_k + c_i h\right) \quad i = 1, 2, \dots, m \quad \text{mit} \quad c_i = \sum_j^m a_{ij} \quad (47)$$

wobei m die Ordnung der RUKU-Verfahren ist.

Butcher-Koeffizientenschema:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1m} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_m & a_{m1} & a_{m2} & \cdots & a_{mm} \\ \hline & w_1 & w_2 & \cdots & w_m \end{array} \quad \begin{array}{c|ccc} & \ddots & & A_R \\ \mathbf{c} & & A_D & \\ & A_L & & \ddots \\ \hline & & \mathbf{w}^T & \end{array}$$

Fallunterscheidungen für $A = \{a_{ij}\} = A_L + A_D + A_R$

- Expliziter RUKU (Einschritt-numerischer Integrator): $A_R = A_D = 0$
- Semi-impliziter RUKU: $A_R = 0$
- Impliziter RUKU: sonst.

Beispiel: Butcher-Koeffizientenschema verschiedener Integratoren

RUKE 4. Ordnung (explizit):

$s_1 = f(x_k, t_k)$	0	0	0	0	0
$s_2 = f(x_k + h/2s_1, t_k + h/2)$	1/2	1/2	0	0	0
$s_3 = f(x_k + h/2s_2, t_k + h/2)$	1/2	0	1/2	0	0
$s_4 = f(x_k + hs_3, t_k + h)$	1	0	0	1	0
$x_{k+1} = x_k + h/6(s_1 + 2s_2 + 2s_3 + s_4)$		1/6	1/3	1/3	1/6

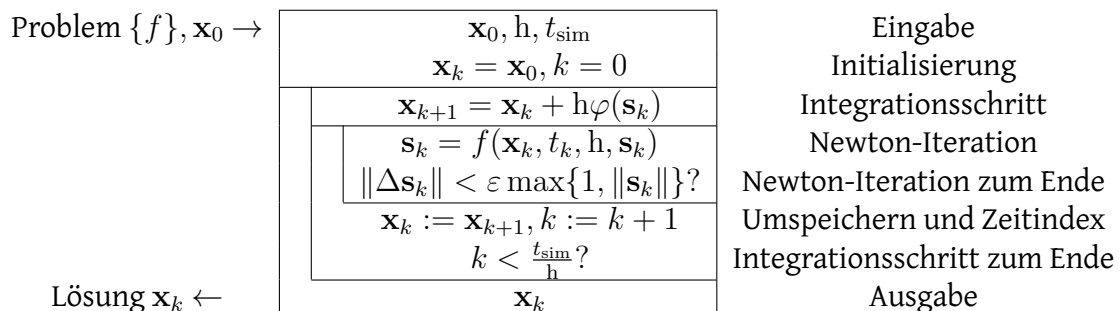
Trapez-Regel (semi-implizit):

$s_1 = f(x_k, t_k)$	0	0	0
$s_2 = f(x_k + h/2(s_1 + s_2), t_k + h)$	1	1/2	1/2
$x_{k+1} = x_k + h/2(s_1 + s_2)$		1/2	1/2

Euler-Rückwärts (implizit):

$s_1 = f(x_k + hs_1, t_k + h)$	1	1
$x_{k+1} = x_k + hs_1$		1

Programmstruktur für Einschritt-numerischen Integrator



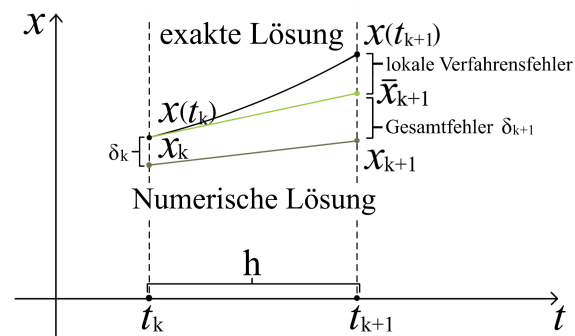
Newton-Iteration in der Programmstruktur:

$$g(s_k) := s_k - f(s_k) \quad \text{mit} \quad g(s_k^*) = 0 \quad (48)$$

$$s_k^{i+1} = s_k^i - \frac{\partial g}{\partial s_k} g(s_k^i) = s_k^i - \left(1 - \frac{\partial f}{\partial s_k}\right) (s_k^i - f(s_k^i)) \quad \text{mit} \quad i = 0, 1, 2, \dots, m \quad (49)$$

Mehrschritt-digitaler Integrator (Mehrschritt-numerische Integration)

2.2.4 Numerische Stabilität numerischer Integration linearer Differentialgleichungen

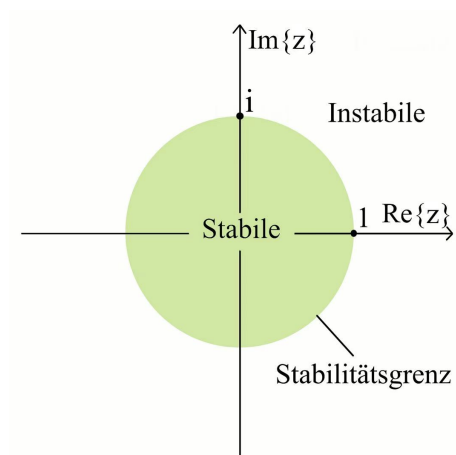


Fortpflanzung von Gesamtfehler δ_k in t_k nach t_{k+1} :

$$\delta_{k+1} = z \delta_k \quad (50)$$

wobei z Fehlerverstärkung ist, die konjugiert komplex sein kann. Verfahren arbeitet numerisch stabil durch:

$$\lim_{h \rightarrow 0} \delta_k = 0 \quad \Leftrightarrow \quad |z| = \left| \frac{\delta_{k+1}}{\delta_k} \right| < 1 \quad \forall k \quad (51)$$



Beispiel: Numerische Stabilität der RUKU-Verfahren

Aufgabe:

$$\dot{x} = \lambda x \quad \text{mit} \quad \lambda = \alpha + w i$$

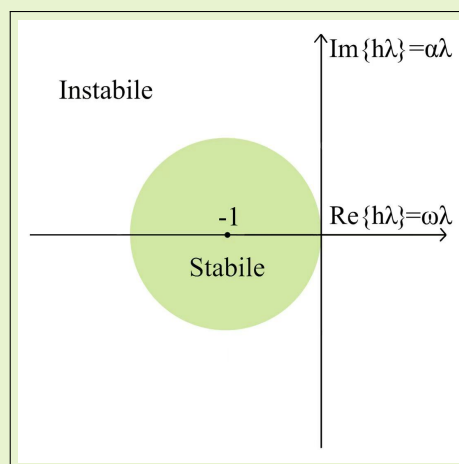
Fehlerverstärkung Euler-Vorwärts:

$$x_{k+1} = x_k + h f(x_k) = x_k + h \lambda x_k$$

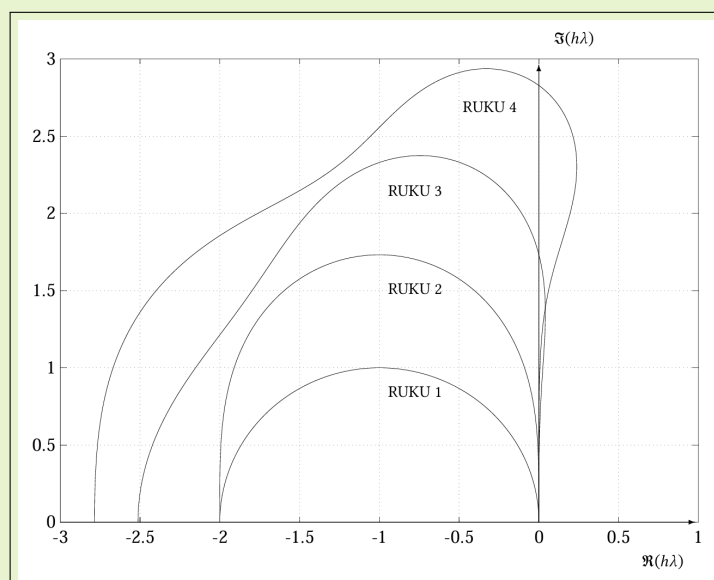
$$x_{k+1} = (1 + h \lambda) x_k, \quad \bar{x}_{k+1} = (1 + h \lambda) \bar{x}_k$$

$$\delta_{k+1} = x_{k+1} - \bar{x}_{k+1} = (1 + h \lambda)(x_k - \bar{x}_k) = (1 + h \lambda) \delta_k$$

$$z = 1 + h \lambda \quad \text{bzw} \quad |1 + h \lambda| < 1$$



RUKU-Verfahren in verschiedener Ordnung:



2.2.5 Numerische Lösung differential-algebraischer Gleichungen

Differential-algebraische Gleichungen stellen ein dynamisches System mit algebraischen Nebenbedingungen dar (semi-expliziter Fall).

$$\dot{\mathbf{x}} = F(\mathbf{x}, \mathbf{z}) \quad \text{mit} \quad t > 0, \mathbf{x}(t_0) = \mathbf{x}_0, \mathbf{x} \in \mathbb{R}^n \quad (52)$$

$$0 = g(\mathbf{x}, \mathbf{z}), \mathbf{z} \in \mathbb{R}^n \quad (53)$$

Semi-explizite Form kann auch wie folgt dargestellt werden:

$$\begin{bmatrix} \mathbf{I} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}, \mathbf{z}) \\ g(\mathbf{x}, \mathbf{z}) \end{bmatrix} \quad (54)$$

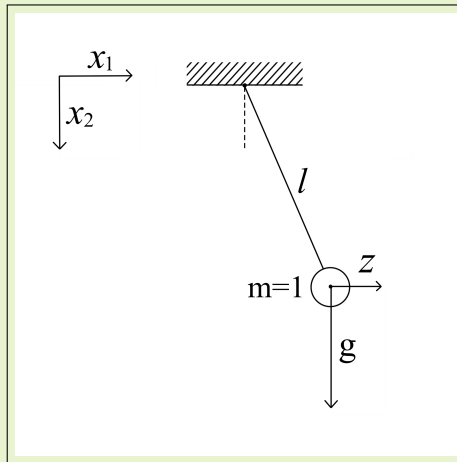
MATLAB arbeitet mit dieser Struktur „odefile“ mit dem Integrator ode45s (Index $\alpha = 1$).

Implizite Darstellung von DAE-System:

$$F(\mathbf{x}, \dot{\mathbf{x}}) = 0 \quad (55)$$

Das DAE-System besitzt den Index α , unten $\alpha \in \mathbb{N}$ die minimale Anzahl an der Differentiation erreicht ist, so dass das System $F(\mathbf{x}, \dot{\mathbf{x}}) = 0, \frac{dF}{dt} = 0, \dots, \frac{d^\alpha F}{dt^\alpha} = 0$ in einem ODE-System (explizit) $\dot{\mathbf{x}} = \Phi(\mathbf{x})$ ausgelöst werden kann.

Beispiel: Mathematisches Pendel



Aufgabe:

$$\begin{cases} \ddot{x}_1 = -zx_1 \\ \ddot{x}_2 = -zx_2 - g \\ 0 = x_1^2 + x_2^2 - l^2 = g(\mathbf{x}) \end{cases}$$

$$\frac{dg}{dt} = 2x_1\dot{x}_1 + 2x_2\dot{x}_2 = 0$$

$$\frac{d^2g}{dt^2} = 2\dot{x}_1^2 + 2x_1\ddot{x}_1 + 2\dot{x}_2^2 + 2x_2\ddot{x}_2 = 0$$

bzw.

$$2\dot{x}_1^2 + 2x_1^2z + 2\dot{x}_2^2 + 2x_2^2z - 2x_2g = 0$$

$$\frac{d^3g}{dt^3} = \dots + a(\mathbf{x})\dot{z} = 0 \quad \text{mit} \quad \text{Index } \alpha = 3$$

3 Prüfungsvorbereitungsaufgaben

Jiaqi Lan

Aufgabe 1 (Explizites Mehrschritt-Verfahren). Das Adams-Bashforth-Verfahren zweiter Ordnung für die Lösung von gewöhnlichen Differenzialgleichungen der Form $\dot{\mathbf{x}} = F(t, \mathbf{x})$ ist gegeben durch

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{h}{24}(55F_k - 59F_{k-1} + 37F_{k-2} - 9F_{k-3}) \quad (56)$$

Hierbei gilt $F_i = F(\mathbf{x}_i, t_i)$. Lösen Sie die folgenden Teilaufgaben.

a) Verifizieren Sie die Ordnung des Verfahrens im Fall $\dim \mathbf{x} = 1$ mit einer skalaren Funktion $f(t, x)$.

Lösung

Verfahrensordnung ist P und lokaler Amplitudenfehler ist $P + 1$. Lokaler Amplitudenfehler kann mit folgender Gleichung berechnet werden:

$$\tilde{x}_{k+1} = \bar{x}_{k+1} - x(t_{k+1})$$

$$\text{wobei } \bar{x}_{k+1} = x(t_k) + \frac{h}{24}(55f(t_k) - 59f(t_{k-1}) + 37f(t_{k-2}) - 9f(t_{k-3}))$$

Eliminieren durch Taylor-Reihe:

$$x(t_{k+1}) \stackrel{\text{Taylor am } x(t_k)}{=} x(t_k) + hf(t_k) + \frac{h^2}{2}f'(t_k) + \frac{h^3}{6}f''(t_k) + O^4(h)$$

$$f(t_{k-1}) \stackrel{\text{Taylor am } f(t_k)}{=} f(t_k) - hf'(t_k) - \frac{h^2}{2}f''(t_k) - O^3(h)$$

$$f(t_{k-2}) \stackrel{\text{Taylor am } f(t_k)}{=} f(t_k) - 2hf'(t_k) - \frac{4h^2}{2}f''(t_k) - O^3(h)$$

$$f(t_{k-3}) \stackrel{\text{Taylor am } f(t_k)}{=} f(t_k) - 3hf'(t_k) - \frac{9h^2}{2}f''(t_k) - O^3(h)$$

In die ursprüngliche Gleichung einsetzen:

$$\tilde{x}_{k+1} = -\frac{1}{3}h^3f''(t_k) + O^4(h) = O^3(h)$$

$$\text{bzw. } P + 1 = 3, P = 2$$

Verfahrensordnung $P = 2$.