Modellbildung und Simulation

Zeitkontinuierliche Modelle mit konzentrierten Parametern

INSTITUT FÜR TECHNISCHE MECHANIK / BEREICH DYNAMIK UND MECHATRONIK

Umfang und Ablauf einer Simulationsstudie

- Ablauf einer Simulationsstudie

Problemspezifikation	Aufgabenstellung, Qualitätskriterien Ergebnisse, Zeit- und Kostenrahmen
Modellbildung	Idealisierungen, Naturgesetze
Mathematisches Modell	algebraische Gleichungen, Dgln.
Modellanalyse	z.B. Lösungen, Eigenfrequenzen,…
Simulationsverfahren	Algorithmen
Simulator	Hardware + Software
Simulationsergebnis, Beurteilung	z.B. Zeitverläufe, Animationen

Institut für Technische Mechanik
Abteilung Dynamik / Mechatronik

Lernziele der heutigen Vorlesung

- Die Studierenden
  - kennen numerische Integrationsverfahren und deren Eigenschaften,
  - kennen die bei der numerischen Lösung gewöhnlicher Differentialgleichungen auftretenden Fehlerarten,
  - können einfache numerische Integrationsverfahren herleiten,
  - können geeignete numerische Integrationsverfahren zur Simulation eines Systems mit konzentrierten Parametern auswählen.

Numerische Integration gewöhnlicher Differentialgleichungen

1. Überblick: Integrationsverfahren für dynamische Systeme
2. Prädiktor-Korrektor-Verfahren
3. Fehleranalyse
4. Stabilität der Integrationsverfahren
5. Schrittweitensteuerung
6. Integrationsverfahren in Matlab

1. Überblick

- Modelltausch: kontinuierliches Modell (gew. Dgln) ⇨ diskretes Modell

- Aufgabe: Wahl eines geeigneten Integrationsverfahrens und Anpassung der Schrittweite

- Klassifizierung:

  Nach der Integrationsart:
  - Explizite Verfahren
  - Implizite Verfahren

  Nach Anzahl der verwendeten Stützstellen:
  - Einschrittverfahren
  - Mehrschrittverfahren

1. Überblick

Problem: Integration von

$$\dot{y}(t) = f(t, y(t)),\ \text{AB:}\, y(t_0) = y_0,\ t \geq t_0$$

Schritt t_n -> t_{n+1}:

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))\,\mathrm{d}\,t$$

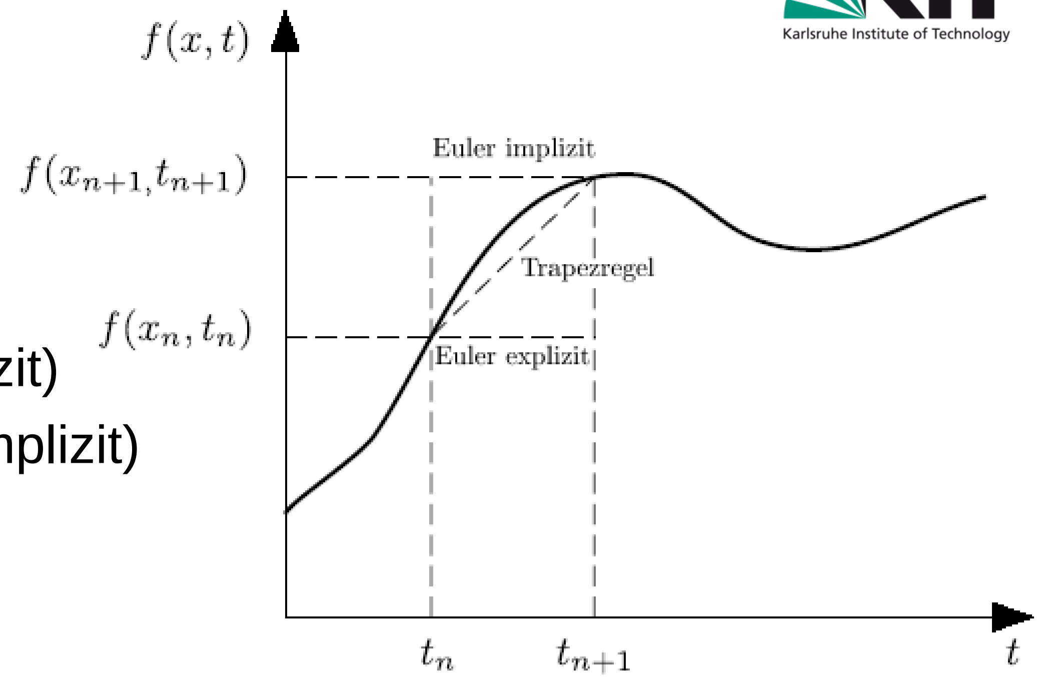
Das Intervall $[t_n, t_{n+1}]$ heißt Zeitschritt.

Es gibt verschiedene Möglichkeiten, y_{n+1} zu bestimmen:

1. Integrationsformeln
2. Polynominterpolation
3. Numerische Differentiation

1. Überblick:

1. Integrationsformeln



z.B. Rechteckregel

- $y_{n+1}=y_n + f(t_n,y_n)\,h$ (Euler explizit)
- $y_{n+1}=y_n + f(t_{n+1},y_{n+1})\,h$ (Euler implizit)

Trapezregel:

$y_{n+1}=y_n + (f(t_{n+1},y_{n+1})+f(t_n,y_n))\,h/2$

-> liefert offenbar ausschließlich Einschrittverfahren

Institut für Technische Mechanik
Abteilung Dynamik / Mechatronik

1. Überblick

2. Näherung von f(t,y(t)) durch Interpolationspolynom p(t) und Lösen des Integrals $\int\limits_{t_n}^{t_{n+1}} p(t)\,\mathrm{d}\,t$

a) Lagrange-Interpolation $p(t) = \sum\limits_{j=0}^{k-1} f(t_{n-j}, y_{n-j}) \frac{\prod_{\substack{i=0\\i\neq j}}^{k-1}(t - t_{n-i})}{\prod_{\substack{i=0\\i\neq j}}^{k-1}(t_{n-j} - t_{n-i})}$

Adams-Bashforth-Verfahren

offenbar explizit

k-Schrittverfahren

für k=1: Euler explizit

Nachteil: Interpolation erfolgt außerhalb des Integrationsintervalls

b) Lagrange-Interpolation $p(t) = \sum\limits_{j=-1}^{k-1} f(t_{n-j}, y_{n-j}) \frac{\prod_{\substack{i=-1\\i\neq j}}^{k-1}(t - t_{n-i})}{\prod_{\substack{i=-1\\i\neq j}}^{k-1}(t_{n-j} - t_{n-i})}$

Adams-Bashforth-Moulton-Verfahren

offenbar implizit, k-Schrittverfahren

(außer k=0, Einschrittverfahren)

für k=1: Euler implizit

für k=2: Trapezregel

1. Überblick

3. Numerische Differentiation

Ersetze $\dot{y}(t)$ durch $\sum_{i=0}^{k} \alpha_i \, y_{n+1-i}$

2. Prädiktor-Korrektor-Verfahren

Nachteil impliziter Verfahren: in jedem Schritt ist ein nichtlineares Gleichungssystem zu lösen!

Ausweg: 2stufiges Verfahren

1. Näherung für y_{n+1} mit explizitem Verfahren
2. einsetzen in implizites Verfahren

Beispiel: Kombination aus explizitem Euler-Verfahren und Trapezregel (Verfahren von Heun)

$$y_{n+1}^{(1)}=y_n+h\,f(t_n,y_n)$$
$$y_{n+1}\quad=y_n+h/2\,(f(t_{n+1},y_{n+1}^{(1)})+f(t_n,y_n))$$

Notation:

$$y_{n+1}=y_n+1/2\,(k_n^{(1)}+k_n^{(2)})$$
$$k_n^{(1)}=h\,f(t_n,y_n)$$
$$k_n^{(2)}=h\,f(t_n+h,y_n+k_n^{(1)})$$

2. Prädiktor-Korrektor-Verfahren

prominentester Vertreter: Runge-Kutta-Verfahren

$$y_{n+1}=y_n+ 1/6\ (k_n^{(1)}+2\ k_n^{(2)}+2\ k_n^{(3)}+k_n^{(4)})$$

$$k_n^{(1)}=h\ f(t_n,y_n)$$

$$k_n^{(2)}=h\ f(t_n+h/2,\ y_n+k_n^{(1)}/2)$$

$$k_n^{(3)}=h\ f(t_n+h/2,\ y_n+k_n^{(2)}/2)$$

$$k_n^{(4)}=h\ f(t_n+h,\ y_n+k_n^{(3)})$$

3. Fehleranalyse

Fehlerarten:

- **lokaler Diskretisierungsfehler l(h):**
  Differenz zwischen Differenzenquotienten und genähertem Differenzenquotienten für die Lösung des Anfangswertproblems, d.h., das, was in jedem Zeitschritt neu an Fehler entsteht.

- Ist $l(h)\,/h^p$ beschränkt für $h \to 0$, so wird das Diskretisierungsschema **konsistent zur Ordnung p** genannt.

- **Konsistenzordnungen:**
  - Euler-Verfahren: p=1
  - Trapezregel: p=2
  - Verfahren von Heun: p=2
  - Runge-Kutta-Verfahren: p=4

3. Fehleranalyse

- **globaler Diskretisierungsfehler e(h):**
  maximaler Fehler zwischen genäherter und exakter Lösung

- Gilt $e(h) \rightarrow 0$ für $h \rightarrow 0$, so wird das Diskretisierungsschema **konvergent** genannt.

- Konsistenz allein garantiert noch keine Konvergenz!

4. Stabilität der Integrationsverfahren

- Modellproblem zur Stabilitätsuntersuchung:

$$\dot{y}(t) = \lambda y(t), \lambda \in C, y(t_0 = 0) = y_0$$

- Exakte Lösung: $y(t) = \exp(\lambda t) y_0$
  stabil für $\mathrm{Re}(\lambda) \leq 0$, sonst instabil.

- Verfahren $y_{n+1}=F(\lambda h)\, y_n$ stabil, wenn $|F(\lambda h)| < 1$

- Beispiele:

explizites Euler-Verfahren:

$$y_{n+1}=y_n+h\lambda y_n$$
$$y_{n+1}= (1+h\lambda)\, y_n$$

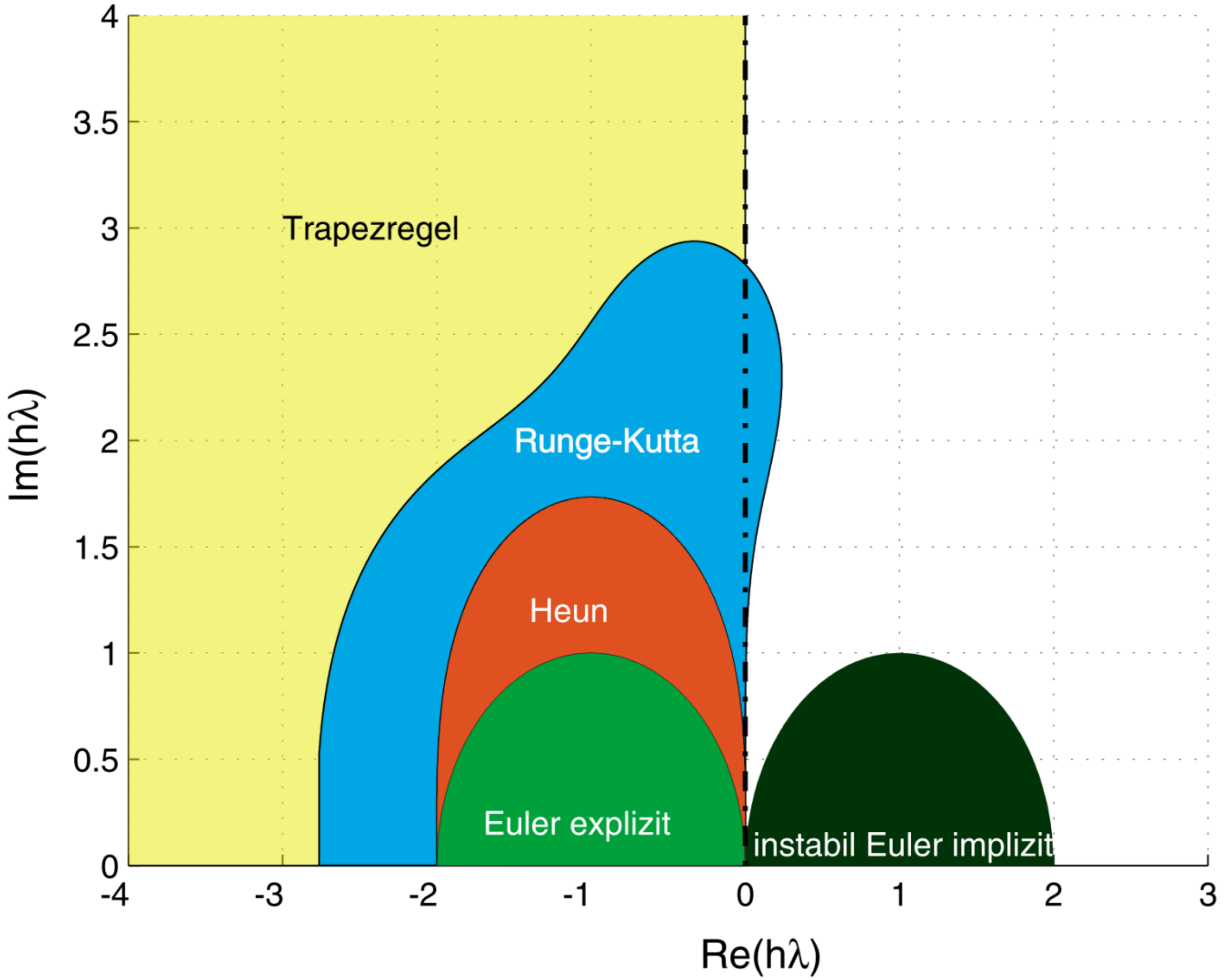
implizites Euler-Verfahren:

$$y_{n+1}=y_n +h\lambda\, y_{n+1}$$
$$y_{n+1} = y_n/(1-h\lambda)$$

Institut für Technische Mechanik
Abteilung Dynamik / Mechatronik

4. Stabilität der Integrationsverfahren

Stabilitätskarte der behandelten Verfahren



4. Stabilität der Integrationsverfahren

Runge-Kutta- Verfahren der Ordnung $p = 4$

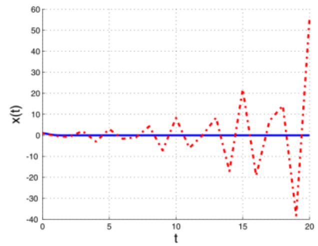
System 1.Ordnung

$$\dot{z}_1 = z_2$$

$$\dot{z}_2 = 2\sigma\, z_2 - (\sigma^2 + \omega^2)\, z_1;$$

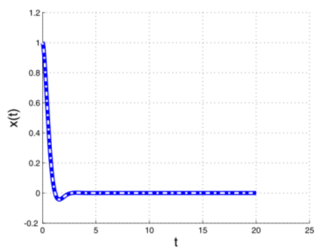
$$z_1(0) = 1,\ z_2(0) = 0$$

gedämpft $\lambda = -2 + 2i$

1


$h = 1$

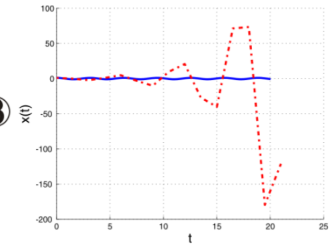
$\rightarrow h\lambda = -2 + 2\,i$

2


$h = 0.1$

$\rightarrow h\lambda = -0.2 + 0.2\,i$

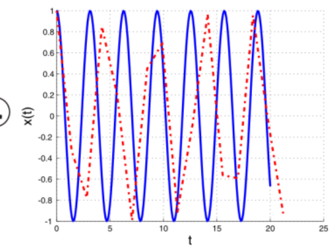
ungedämpft $\lambda = 2i$

3


$h = 1.5$

$\rightarrow h\lambda = 3.0\,i$

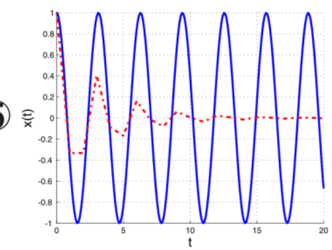
instabil

4


$h = 1.4142$

$\rightarrow h\lambda = 2.8284\,i$

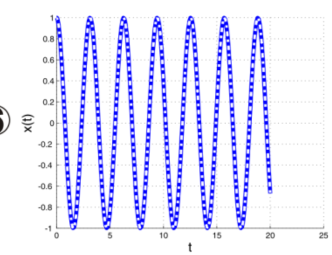
Schrittweite zu groß

5


$h = 1.5$

$\rightarrow h\lambda = 2\,i$

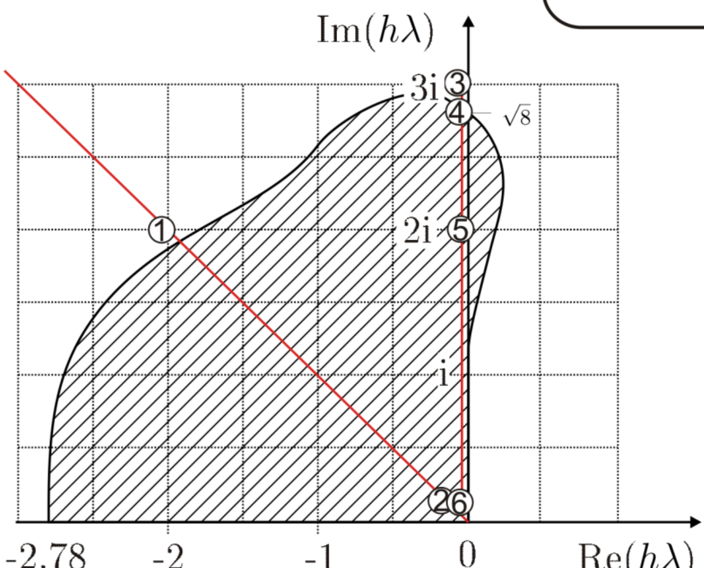
Schrittweite zu groß

6


$h = 0.1$

$\rightarrow h\lambda = 0.2\,i$

Stabile Integration ungedämpfter Systeme



5. Schrittweitensteuerung

- Schrittweitensteuerung
- Idee: lokalen Fehler messen und Schrittweite entsprechend anpassen
- Benötigt: Kontrollrechnung mit einem Verfahren der Ordnung p+1

$$|y_{n+1} - \tilde{y}_{n+1}| \leq Ch^p + \tilde{C}h^{p+1} \approx Ch^p$$

Kontrollrechnung sollte mit bereits vorgenommenen Funktionsauswertungen auskommen

(z.B. Runge-Kutta-Fehlberg-Verfahren)

Institut für Technische Mechanik
Abteilung Dynamik / Mechatronik

6. Integrationsverfahren in Matlab

Verfahren für nicht steife bis mäßig steife Dgln.

- **ode45: Explizites Runge-Kutta (4,5)-Einschrittverfahren**

Beste Wahl für den ersten Versuch.

- **ode23: Explizites Runge-Kutta (2,3)-Einschrittverfahren**

Bei geringeren Genauigkeitsanforderungen und mäßiger Steifheit evtl. günstiger als ode45.

- **ode113: Adams-Bashforth-Moulton Mehrschrittverfahren**
  - mit Schrittweiten- und Ordnungssteuerung (bis Ordnung 13),
  - Selbststartend
  - Bei hohen Genauigkeitsanforderungen und teurer rechter Seite (z.B. bei Mehrkörpersystemen) günstiger als ode45.

6. Integrationsverfahren in Matlab

Verfahren für steife Dgln.

- **ode15s: Implizites Mehrschrittverfahren**
  Basierend auf num. Differenzen (NDF) bzw. Rückwärtsdifferenzen (BDF, Gear)
  Erste Wahl, wenn ODE 23/45/113 wegen Steifheit zu langsam werden.

- **ode23s: Implizites Einschrittverfahren**

Bei geringeren Genauigkeitsanforderungen evtl. günstiger als ode15s.

- **ode23t: Implizite Trapezregel (Einschrittverfahren)**

Bei geringeren Genauigkeitsanforderungen evtl. günstiger als ode15s.
Keine numerische Dämpfung der Lösung.

- **ode23tb: Implizites Runge Kutta Einschrittverfahren**

Bei geringeren Genauigkeitsanforderungen evtl. günstiger als ode15s.

6. Integrationsverfahren in Matlab

Schrittweitensteuerung in Matlab:

Alle Verfahren besitzen Schrittweitensteuerung

Schranken für relativen und absoluten Fehler vorgeben:

|e(i)| < max(RelTol*abs(y(i)),AbsTol(i))

```
options=odeset('RelTol',1.e-7,'AbsTol',1.e-10)
sol=ode45(@my_ode,[0 Tmax],init_values,options);

function dy = my_ode(t,y)
…
```

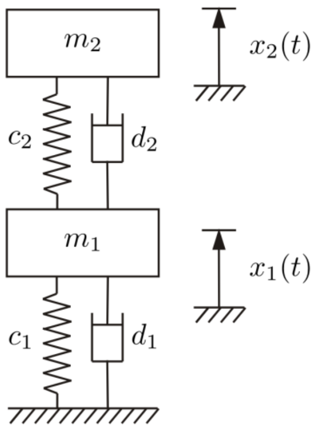
Ausblick

- Nächste Vorlesung:
  - Simulationsverfahren: Was ist bei der numerischen Integration gewöhnlicher Differentialgleichungen zu beachten, wenn Nebenbedingungen erfüllt werden müssen?
- Nächste Übung:
  - Numerische Integrationsverfahren für gewöhnliche Differentialgleichungen implementieren, Ergebnisse analysieren (Pool)

Problem der Woche

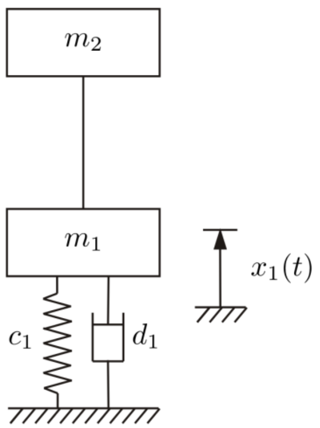
Zwei Varianten eines einfachen Fahrzeugmodells
(Viertelfahrzeug mit Nutzlast)

Modell 1



$$m_1 = 1, \quad m_2 = 2$$
$$c_1 = 1, \quad c_2 = 400$$
$$d_1 = 0, \quad d_2 = 0$$

Modell 2



$$m_1 = 1, \quad m_2 = 2$$
$$c_1 = 1$$
$$d_1 = 0$$

Wie sieht das Weg-Zeit-Verhalten von Masse 1 aus?

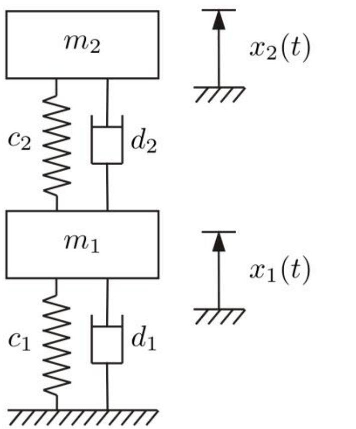
Welche Schwierigkeiten könnten bei der numerischen Integration auftreten?

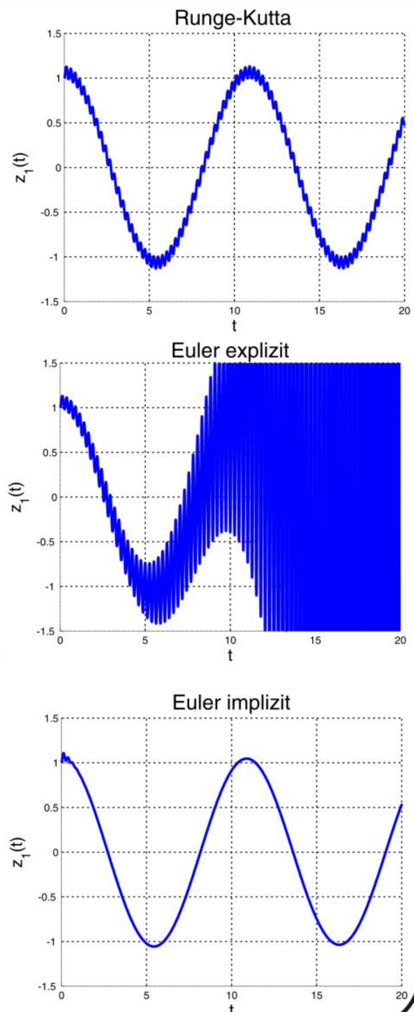
Problem der Woche

Steife Differentialgleichungen:

Größter EW-Betrag/Kleinster EW-Betrag > 10^3 bis 10^4

Modell 1

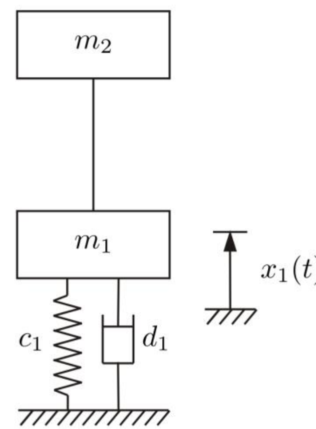


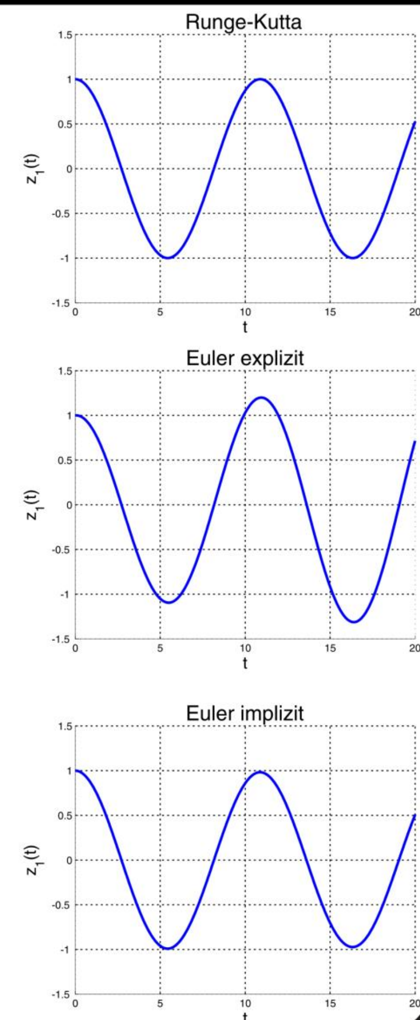


$$m_1 = 1, \quad m_2 = 2$$
$$c_1 = 1, \quad c_2 = 400$$
$$d_1 = 0, \quad d_2 = 0$$

$$\lambda_{1,2} = \pm 0.5770\, i$$
$$\lambda_{3,4} = \pm 24.5085\, i$$

Modell 2





$$m_1 = 1, \quad m_2 = 2$$
$$c_1 = 1$$
$$d_1 = 0$$

$$\lambda_{1,2} = \pm 0.5774\, i$$