

# Einführung in Computational Engineering



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## Grundlagen der Modellierung und Simulation

### 9. Vorlesung: Zeitkontinuierliche Modellierung und Simulation

9. Dezember 2013

Prof. Dr. Jan Peters

MOODLE  
TEST IN  
DREI  
FOLIEN

produziert vom

**HRZ**  
Hochschulrechenzentrum

# Meisenantworten

- „Sag‘ uns wie alt Du bist...“ *Ihr dürft raten: <https://moodle.informatik.tu-darmstadt.de/mod/choice/view.php?id=19466>*
- Bitte keine Poloshirts sondern Hawai-Hemden! *Wenn Ihr den TU Präsidenten überredet, mir, Chris und Herke einen Weihnachtsturlaub in Hawai zu spendieren? Gerne!*
- Überblicksfolie vor jeder Vorlesung. *Ab diesem Mal dabei. Nachgebesserte Folien sind auch auf Moodle...*

## Stimmabgaben

- *Vorlesung und Übung werden ab nächster Woche getauscht...*

Abstimmoptionen	Teilnehmerzahl	Prozent der Teilnehmer/innen	Grafische Darstellung
Bitte tauschen!	71	62,3%	
Bitte so behalten!	24	21,1%	
Mir doch egal!	19	16,7%	

# Meisenantworten

- Beibehaltung von Pausen:

Stimmabgaben			
Abstimmoptionen	Teilnehmerzahl	Prozent der Teilnehmer/innen	Grafische Darstellung
Ich	63	56,8%	
Ich nicht, ich möchte früher nach Hause.	32	28,8%	
Ich nicht, ich möchte lieber mehr lernen.	16	14,4%	

- Nutzung von früherem Vorlesungsende:

Stimmabgaben			
Abstimmoptionen	Teilnehmerzahl	Prozent der Teilnehmer/innen	Grafische Darstellung
... das schwerere Themen wiederholt werden?	33	33,3%	
... das Ihr früher nach Hause könnt?	26	26,3%	
... ein Beispiel vorgerechnet wird.	40	40,4%	

Folgende Schreibweise hat bei mir unterschiedlich für welche Veränderung gesorgt:

$$\bullet \dot{x} = A \cdot x + B \cdot u \quad (\text{z.B. 4. Vorlesung, Folie 26})$$

$$\bullet \Delta \dot{x} = A \cdot \Delta x + B \cdot \Delta u \quad (\text{z.B. 5. Vorlesung, Folie 47})$$

In der ersten Gleichung ist  $A = f(x)$ , in der zweiten,

jedoch  $A = f_0(x) \neq f(x)$  (dieser Unterschied wird erst

(durch sorgfältigeren Vergleichen) über mehrere Vorlesungsfolien hinweg deutlich und kann den Einstieg des Studenten in das mathematisch relativ

angenehme Themen doch erheblich erschweren.

Daher habe ich die andere Wahl der Bezeichnungen dieser Matrizen oder einen Konvent späterhens in Folie 48 (4. Vorlesung) als sinnvoll!

$$\dot{x} = f(x, u)$$

$$= Ax + Bu \leftarrow \text{lineare 1}$$



$$\Delta \dot{x} = \frac{\partial f}{\partial x} \Big|_{x=x_0} \cdot \Delta x$$

$$\Delta \dot{x} = (\dot{x} - \dot{x}_0) \quad (x - x_0)$$

# Überblick der Vorlesungsinhalte



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

1. Einführung
2. Diskrete Modellierung und Simulation
3. Zeitkontinuierliche Modellierung und Simulation
4. Teilschritte einer Simulationsstudie
5. Interpretation und Validierung
6. Modulare und objektorientierte Modellierung und Simulation
7. Parameteridentifikation von Modellen

Bitte jetzt auf Moodle eine Frage  
beantworten!



---

Grundlagen der Modellierung und Simulation

## **3. ZEITKONTINUIERLICHE MODELLIERUNG UND SIMULATION**



## Heutige Lernziele: Kernfragen

---

- Was ist ein symplektischer Euler? Wie verhalten sich impliziter, expliziter und symplektischer Euler am periodischen System?
- Was sind Prädikator-Korrektor Verfahren?
- Was ist das Verfahren von Heun? Wie kann man es intuitiv interpretieren?  
Warum ist es ein Prädikator-Korrektor Verfahren?
- Was ist das Runge-Kutta Verfahren? Welche Ordnung hat es?
- Was sind Konvergenz- und Konsistenzordnung?
- Wie funktioniert hier Schrittweitensteuerung? Welche Varianten gibt es?
- Wie kann man mit Differentialgleichungen mit Unstetigkeiten umgehen?

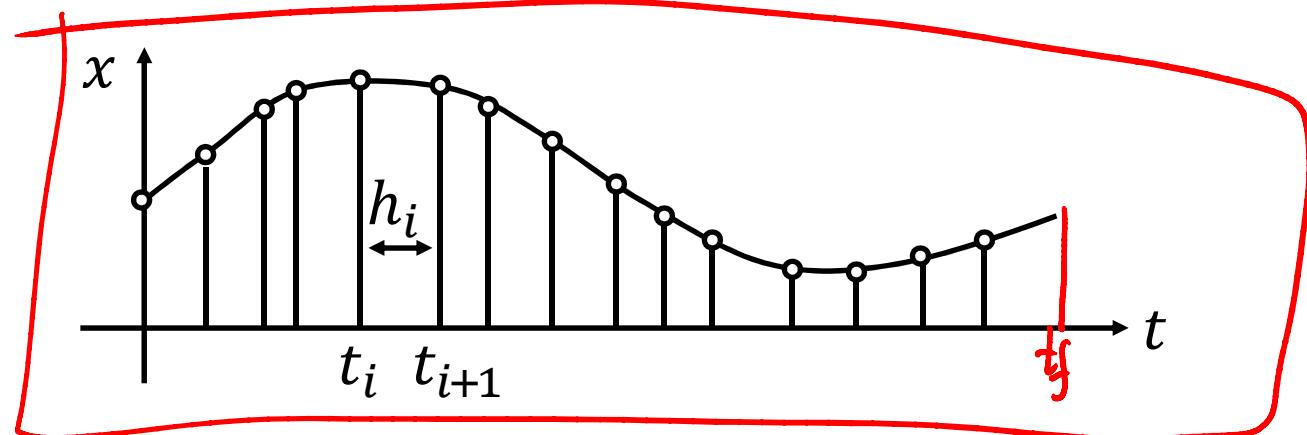
# Wiederholung

- Man betrachte die skalare (Zustands-)DGL:

$$\dot{x} = f(x), \quad x(0) = 0, \quad 0 \leq t \leq t_f$$

mit "final time"  $t_f$

- Aufstellen der Lösung zu diskreten Zeitpunkten  $t_{i+1} = t_i + h_i$



# Wiederholung



- Unterscheidung in der Art der Approximation der Fläche unter der Funktion  $f$  und/oder des Gradienten dieser
  - Einschrittverfahren (ESV), z.B.  $\int_{t_i}^{t_{i+1}} f(x(\tau)) d\tau \approx h f(x(t_i))$
  - Mehrschrittverfahren (MSV) allgemein als
$$\int_{t_i}^{t_{i+1}} f(x(\tau)) d\tau = \sum_{k=0}^n a_k \int_{t_{i-k-1}}^{t_{i-k}} f(x(\tau)) d\tau + h \sum_{k=0}^m b_k f(x(t_{i-k}))$$
  - Extrapolationsverfahren
- Diese Verfahrensklassen können weiter unterteilt werden:
  - Explizites Verfahren  $\int_{t_i}^{t_{i+1}} f(x(\tau)) d\tau \approx h f(x(t_i))$
  - Implizites Verfahren  $\int_{t_i}^{t_{i+1}} f(x(\tau)) d\tau \approx h f(x(t_{i+1}))$

# Beispiel: Feder-Masse System

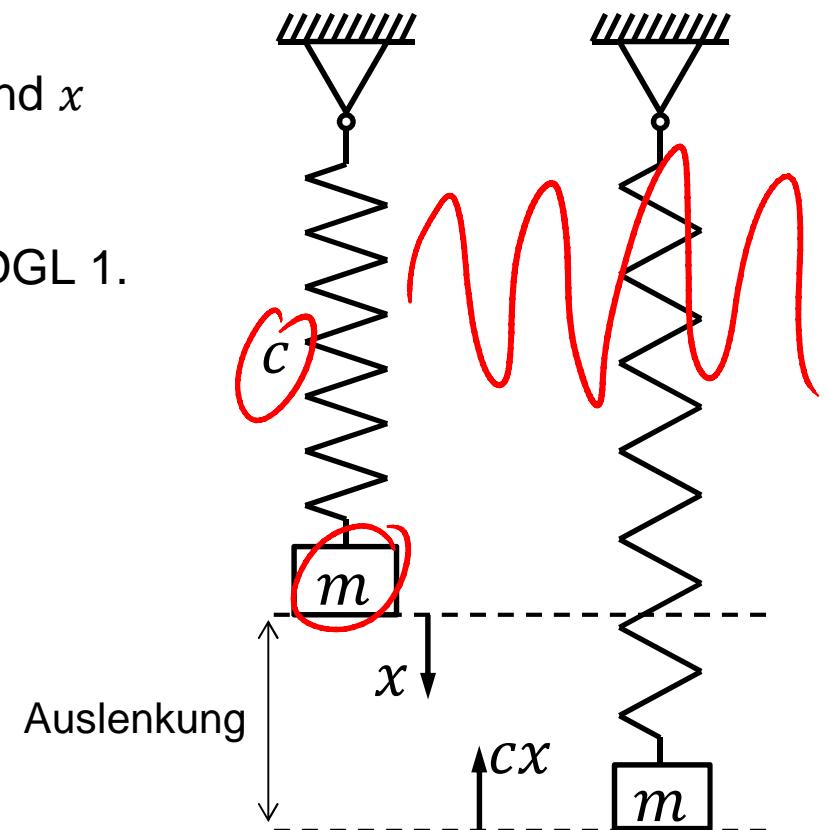


- Schwingung einer Masse an einer Feder
  - Masse  $m$ , Federkonstante  $c$  und Systemzustand  $x$  zum Zeitpunkt  $t$  (Längenveränderung)
  - DGL 2. Ordnung  $\ddot{x}(t) = -\frac{c}{m}x(t)$  kann in eine DGL 1. Ordnung

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{c}{m} & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = A \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

transformiert werden

- Wie können wir dieses System numerisch aufintegrieren?



# Beispiel: Feder-Masse System



- Standard Möglichkeiten

- Impliziter Euler:  $x(t_{i+1}) = x(t_i) + \int_{t_i}^{t_{i+1}} f(x(\tau)) d\tau$

$$\approx x(t_i) + h f(x(t_i))$$

$$= x(t_i) + h A x(t_i) = (I + hA)x(t_i)$$

- Expliziter Euler:  $x(t_{i+1}) = x(t_i) + \int_{t_i}^{t_{i+1}} f(x(\tau)) d\tau$

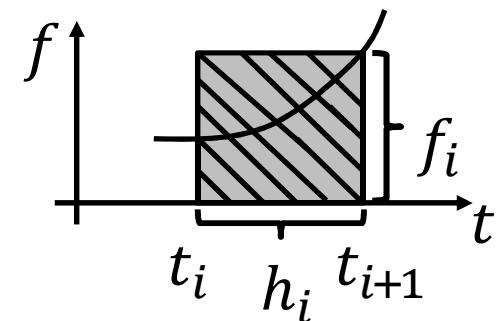
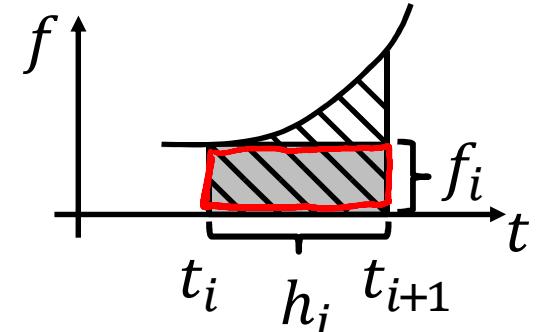
$$\approx x(t_i) + h f(x(t_{i+1}))$$

$$x(t_{i+1}) = x(t_i) + h A x(t_{i+1})$$

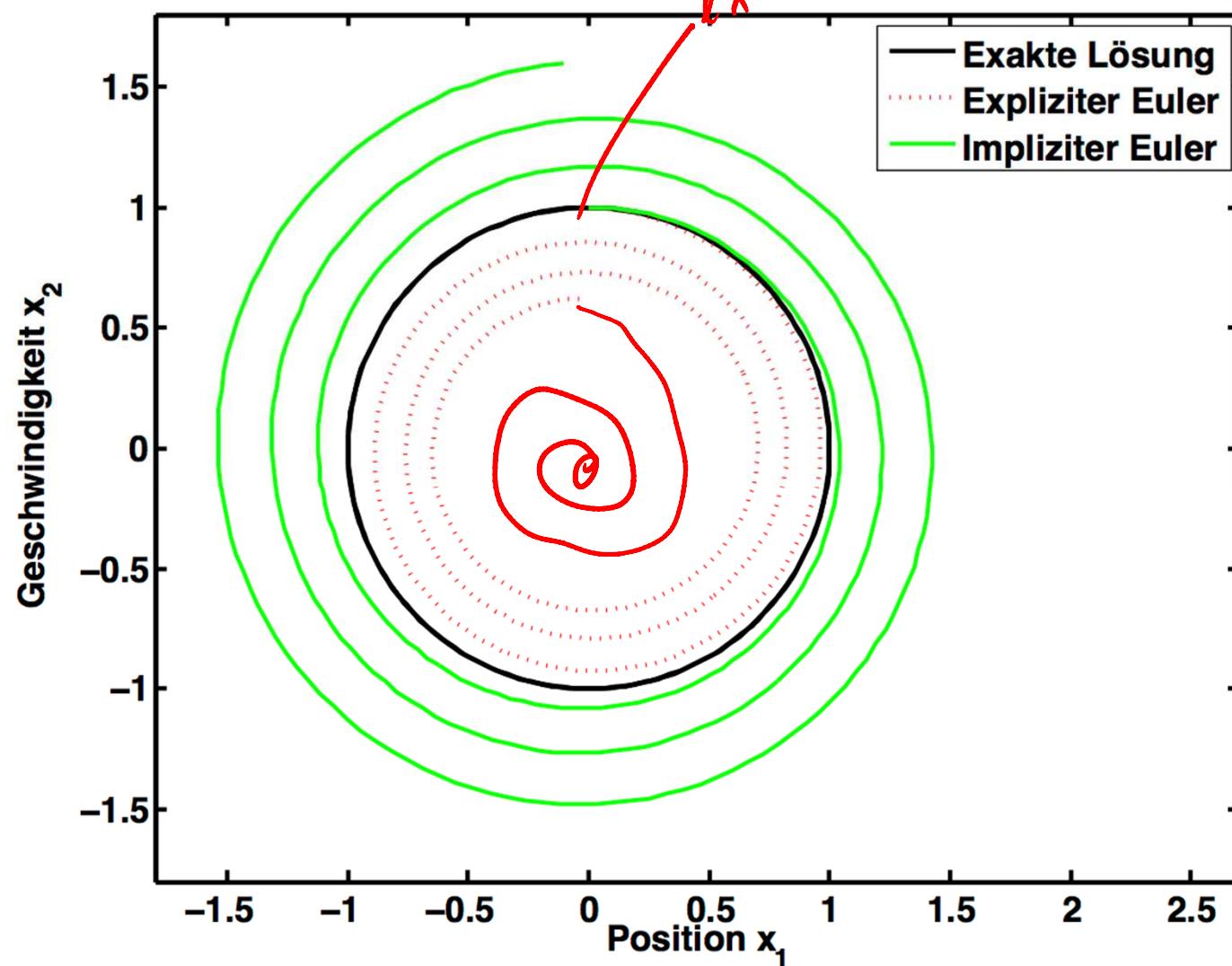
$$\text{Umformen: } x(t_{i+1}) = x(t_i) + h A x(t_{i+1})$$

$$\Rightarrow x(t_{i+1}) = (I - hA)^{-1} x(t_i)$$

*Matrixinversion  
TEUER!*



# Beispiel: Feder-Masse System



# Beispiel: Feder-Masse System

- **Symplektischer-Euler:** Speziell nur für DGLs 2. Ordnung wie  $\ddot{x}(t) = -\frac{c}{m}x(t)$ .

- Impliziter Euler für die Position

$$x_1(t_{i+1}) = x_1(t_i) + h f_1(x(t_{i+1}))$$

- Expliziter Euler für die Geschwindigkeit

$$x_2(t_{i+1}) = x_2(t_i) + h f_2(x(t_i))$$

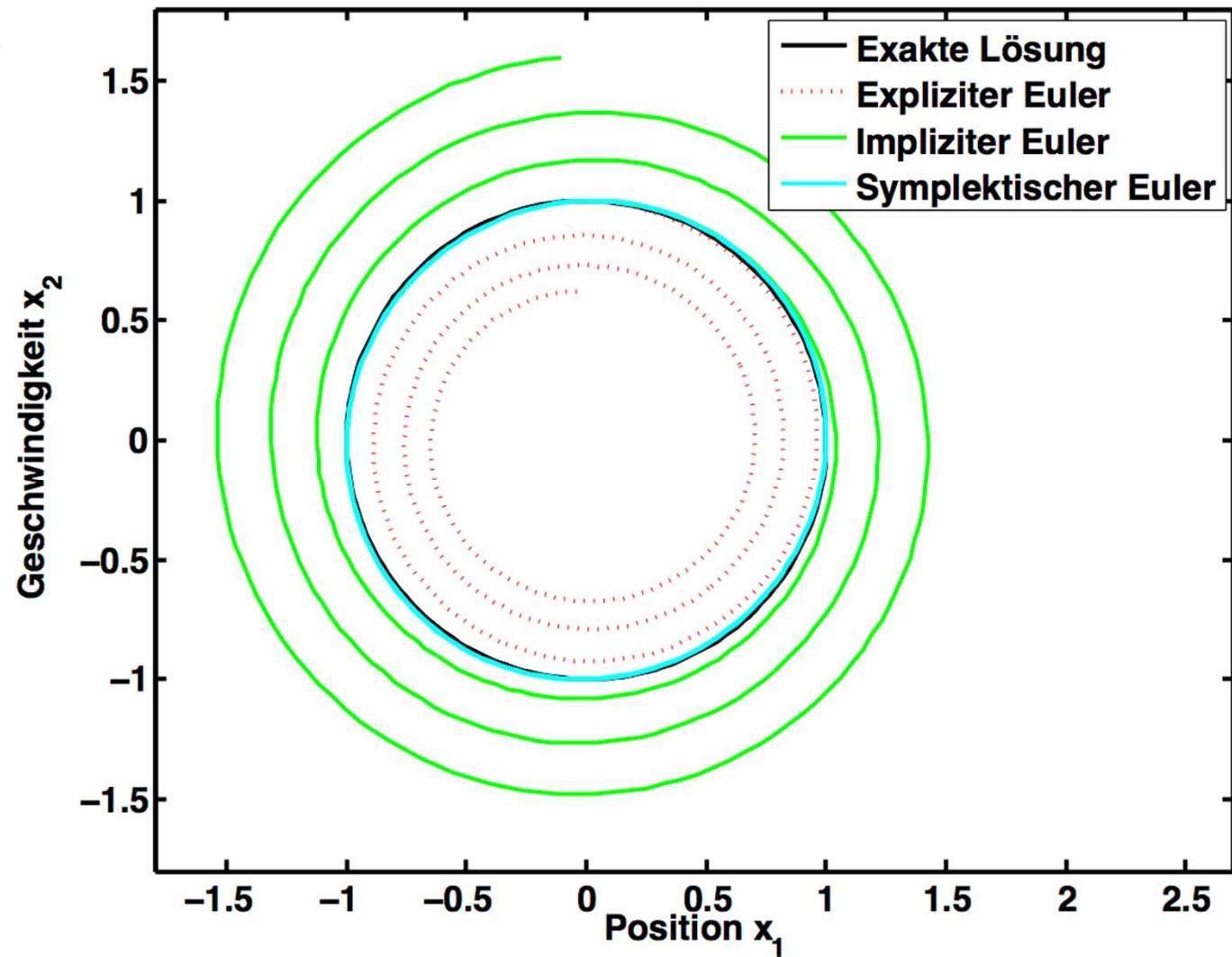
- In unserem Beispiel

$$\begin{aligned} x_2(t_{i+1}) &= x_2(t_i) - h \frac{c}{m} x_1(t_i) \\ x_1(t_{i+1}) &= x_1(t_i) + h x_2(t_{i+1}) \\ &= x_1(t_i) + h(x_2(t_i) + h f_2(x(t_i))) \\ &= x_1(t_i) + h \left( x_2(t_i) - h \frac{c}{m} x_1(t_i) \right) \end{aligned}$$

Höhere Genauigkeit bei gleicher Zahl von Funktionsevaluationen.

# Beispiel: Feder-Masse System

Symplectic Euler  
funktioniert deutlich  
besser bei gleichem  
Aufwand geht aber  
(fast) nur bei  
mechanischen  
Systemen.



## 3.6.2 Auswahl von numerischen Integrationsverfahren

- Zur Auswahl und zur Bewertung von numerischen Integrationsverfahren betrachtet man:
  - Rechenaufwand (Berechnungseffizienz)
  - Genauigkeit (Approximationsfehler)
  - Eignung für steife Systeme (Stabilität)
  - Implementierungsaufwand (Eigenprogrammierung oder Verwendung einer Bibliothek)

### 3.6.2 Einschrittverfahren

- Ordnung eines ESV
  - Numerische Integrationsverfahren sind an eine Verfahrensordnung geknüpft (z.B. das Euler-Verfahren ist erster Ordnung, mehr dazu bei Konsistenzordnung)
- Prädiktor-Korrektor Verfahren
  - An die Ordnung eines Verfahrens gekoppelt
  - Der Prädiktor beschreibt den jeweilig vorherigen Schritt, den man macht
  - Der Korrektor beschreibt die Gewichtung des jeweilig momentanen Schrittes, den man macht

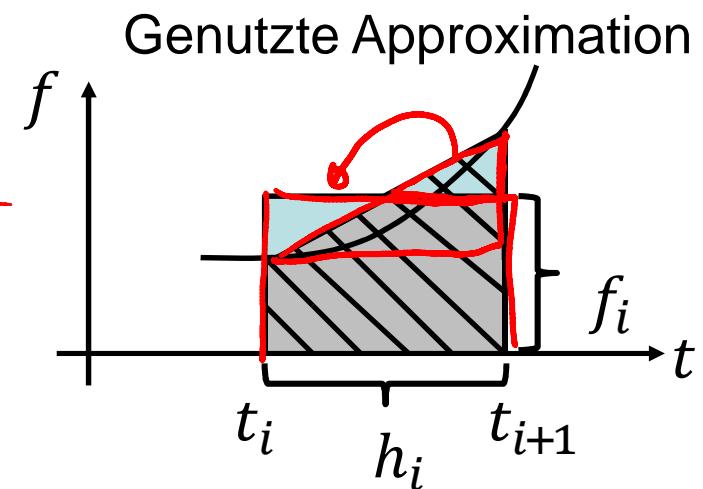
### 3.6.3 Heun-Verfahren

- Oft auch „verbessertes“ oder „modifiziertes“ Euler-Verfahren genannt

- Graphisch betrachtet findet hierbei eine Trapez Approximation statt, anstatt einer Rechteck Approximation

- Im Gegensatz zu Euler ein Prädiktor-Korrektor Verfahren

- Verfahren zweiter Ordnung



### 3.6.3 Heun-Verfahren



- Verfahrensvorschrift des ESV von Heun gegeben durch:

$$x(t_{i+1}) = x(t_i) + \frac{h_i}{2} \cdot (k_1 + k_2)$$



- $k_1$  ist der Korrektor-Schritt mit  
$$k_1 := f(t_i, x(t_i))$$
  - $k_2$  ist der Prädiktor-Schritt mit  
$$k_2 := f(t_{i+1}, x(t_i) + h_i \cdot k_1)$$
  - Approximationsfehler ist  $O(h^2)$
  - Heun-Verfahren ist ein Verfahren 2. Ordnung
- Approximation in 2. Ordnung*

### 3.6.3 Heun-Verfahren: Numerisches Beispiel



- Beispiel-DGL:  $\dot{x}(t) = -x(t)$ , mit  $x(0) = 1$
- Gesucht:  $x(t)$  mit  $0 \leq t \leq 5$
- Anwendung des Heun-Verfahrens ( $x(t_i) = x_i$ )

$$x(t_{i+1}) = x(t_i) + \frac{h_i}{2} \cdot (k_1 + k_2), \quad \text{mit } h_i = h = 0.5$$

$$x_0 = x(0) = 1$$

$$x_1 = x(0.5) = x_0 + 0.25(-1.5x_0) = 0.625$$

$$x_2 = x(1) = x_1 + 0.25(-1.5x_1) = 0.390$$

⋮

### 3.6.3 Heun-Verfahren: Feder-Masse Beispiel

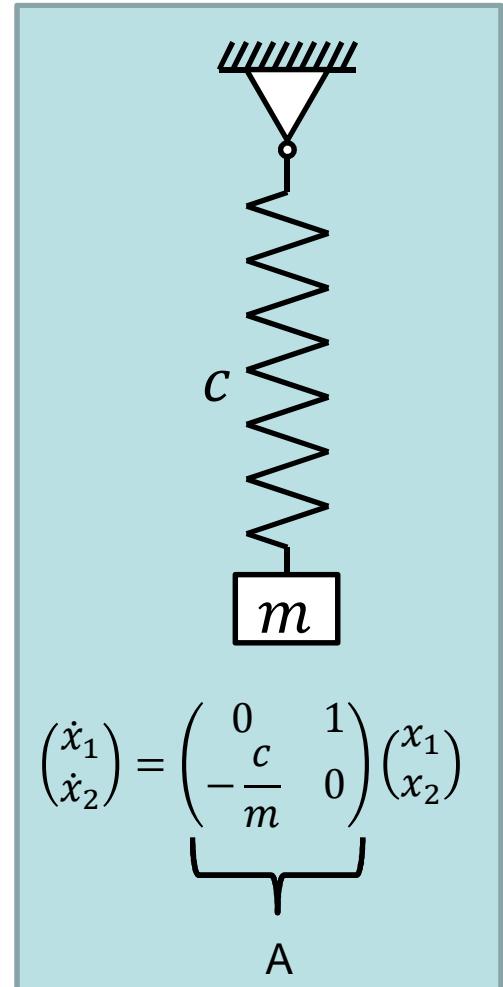
- Heun-Verfahren beim Feder-Masse Modell

$$\begin{aligned}
 x(t_{i+1}) &= x(t_i) + \frac{h_i}{2} \cdot (f(t_i, x(t_i)) + f(t_{i+1}, x(t_i) + h_i \cdot k_1)) \\
 &= x(t_i) + \frac{h_i}{2} \left( Ax(t_i) + A(x(t_i) + h_i Ax(t_i)) \right) \\
 &= x(t_i) + h_i Ax(t_i) + \frac{h_i^2 A^2}{2} x(t_i)
 \end{aligned}$$

Expliziter  
Euler

Term 2. Ordnung in  
Taylor Approximation

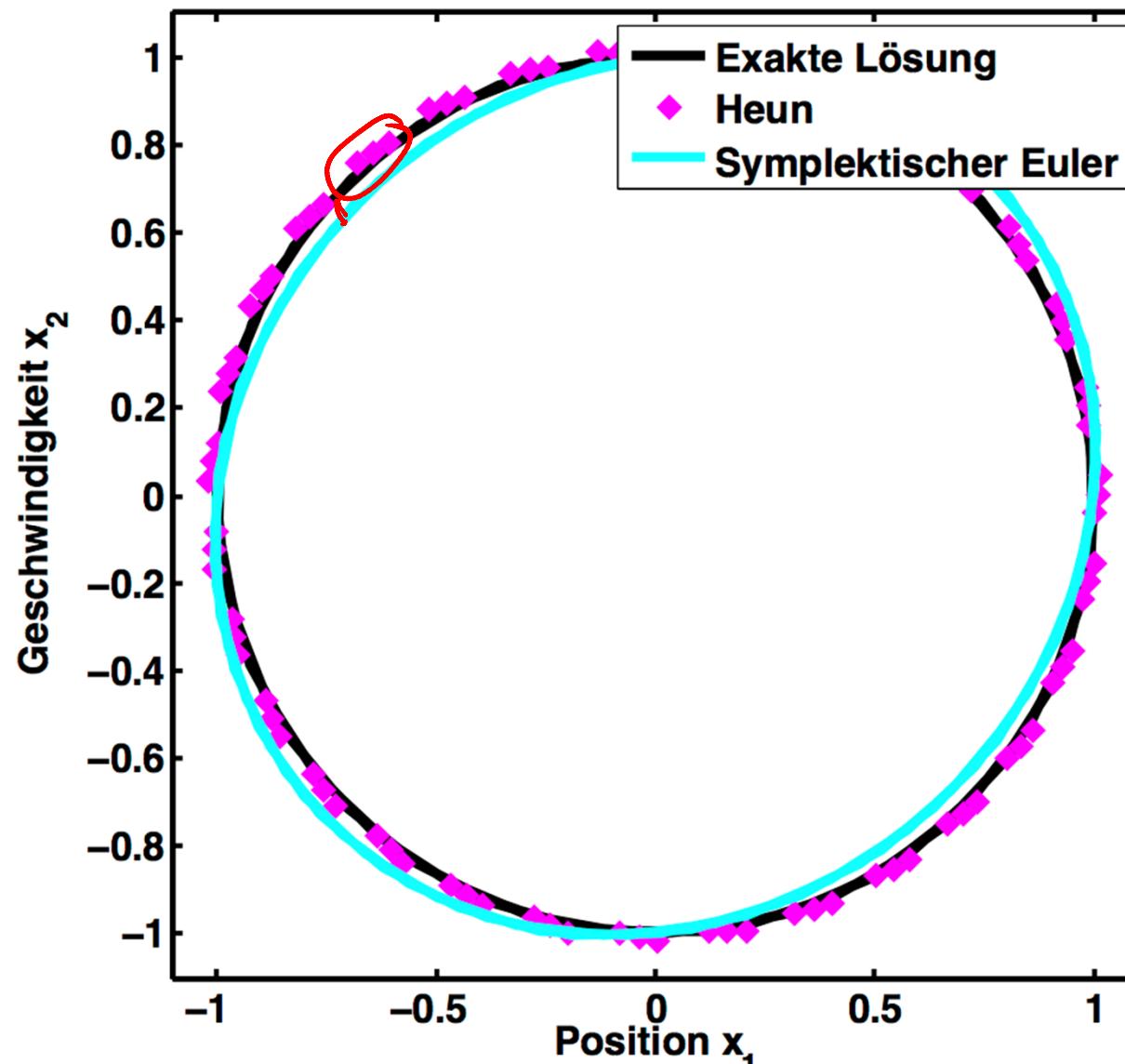
$\approx e^{At}$



### 3.6.3 Heun-Verfahren: Feder-Masse Beispiel



$h = 0,2$



### 3.6.4 Runge-Kutta- 4

- Verfahrensvorschrift des RK4 ist gegeben durch:

$$x(t_{i+1}) = x(t_i) + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$

- Mit den Zwischenstufen  $k_i$ :

$$\underline{k_1 = f(t_i, x(t_i))}$$

$$k_2 = f\left(t_i + \frac{h}{2}, x(t_i) + \frac{h}{2} k_1\right)$$

$$k_3 = f\left(t_i + \frac{h}{2}, x(t_i) + \frac{h}{2} k_2\right)$$

$$k_4 = f(t_{i+1}, x(t_i) + h \cdot k_3)$$

- Verfahren vierter Ordnung

- Approximationsfehler ist  $\underline{O(h^4)}$

- Falls  $f$  4-mal stetig differenzierbar ist innerhalb von  $]t_i, t_{i+1}[$

### 3.6.4 Runge-Kutta- 4: Numerisches Beispiel



- Beispiel-DGL:  $\dot{x}(t) = -x(t)$ , mit  $x(0) = 1$
- Gesucht:  $x(t)$  mit  $0 \leq t \leq 5$
- Anwendung des Runge-Kutta-4 ( $x(t_i) = x_i$ )

$$x(t_{i+1}) = x(t_i) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad \text{mit } h_i = h = 0.5$$

$$x_0 = \underline{\underline{x(0)}} = 1$$

$$x_1 = x(0.5) = x_0 + \underline{\underline{0.08(-4.72x_0)}} = 0.607$$

$$x_2 = x(1) = x_1 + \underline{\underline{0.08(-2.86)}} = 0.368$$

⋮

### 3.6.3 Runge-Kutta- 4: Feder-Masse Beispiel



- Runge-Kutta- 4 beim Feder-Masse Modell

$$x(t_{i+1}) = x(t_i) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

mit

$$k_1 = f(x(t_i)) = Ax(t_i)$$

$$k_2 = f\left(x(t_i) + \frac{h}{2}k_1\right) = Ax(t_i) + \frac{h}{2}A^2x(t_i)$$

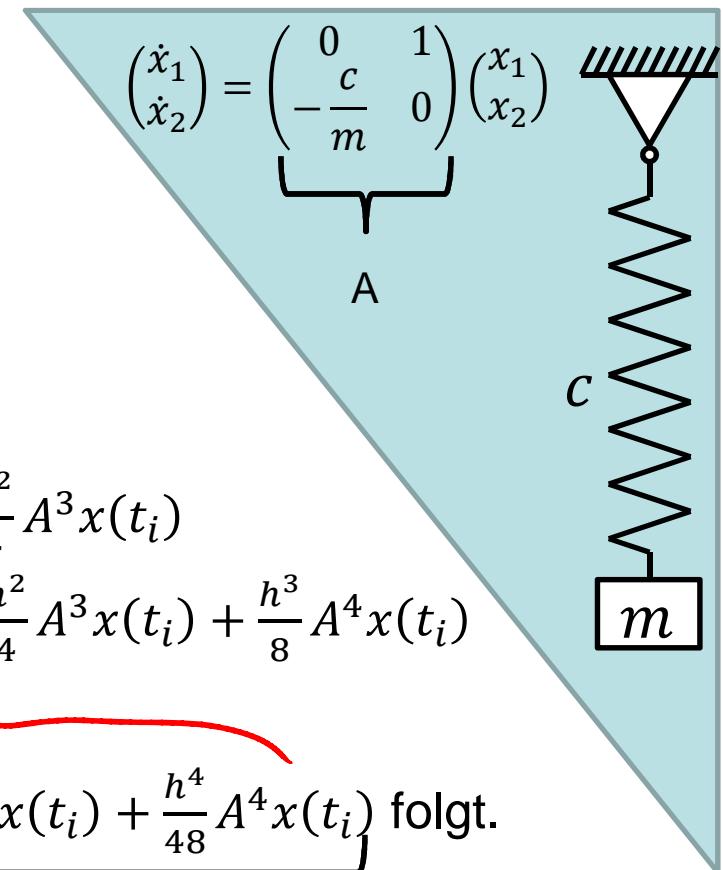
$$k_3 = f\left(x(t_i) + \frac{h}{2}k_2\right) = Ax(t_i) + \frac{h}{2}A^2x(t_i) + \frac{h^2}{4}A^3x(t_i)$$

$$k_4 = f(x(t_i) + h \cdot k_3) = Ax(t_i) + hAx(t_i) + \frac{h^2}{4}A^3x(t_i) + \frac{h^3}{8}A^4x(t_i)$$

woraus

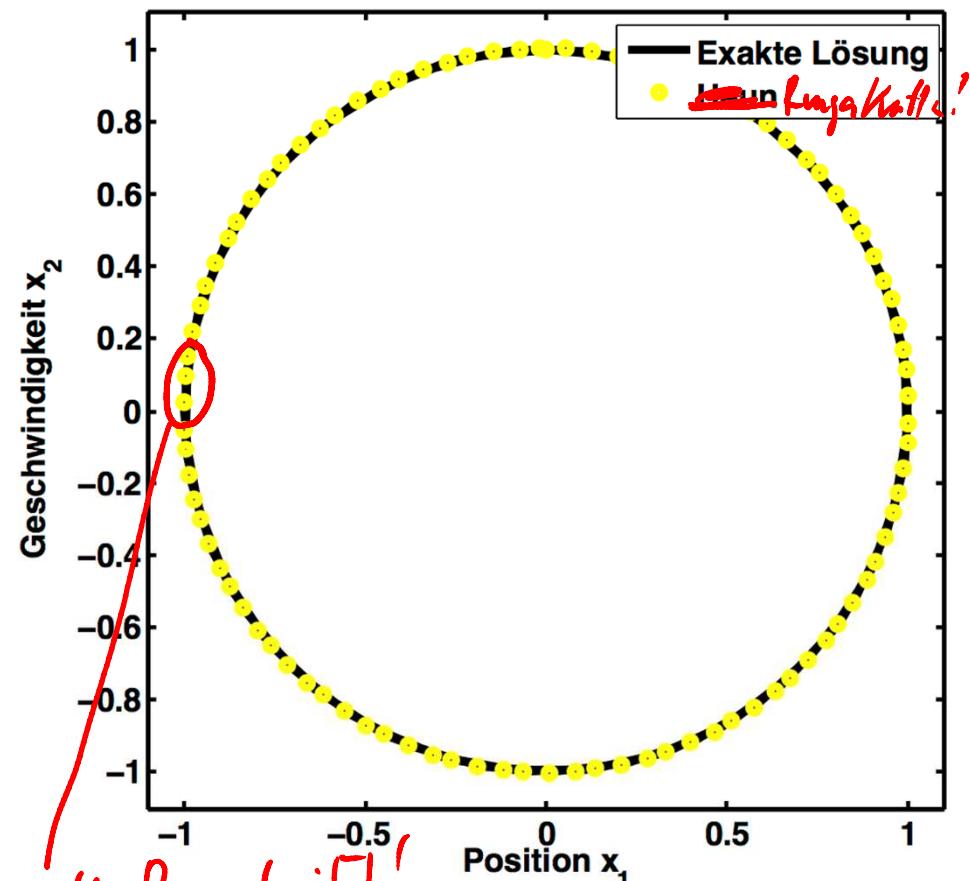
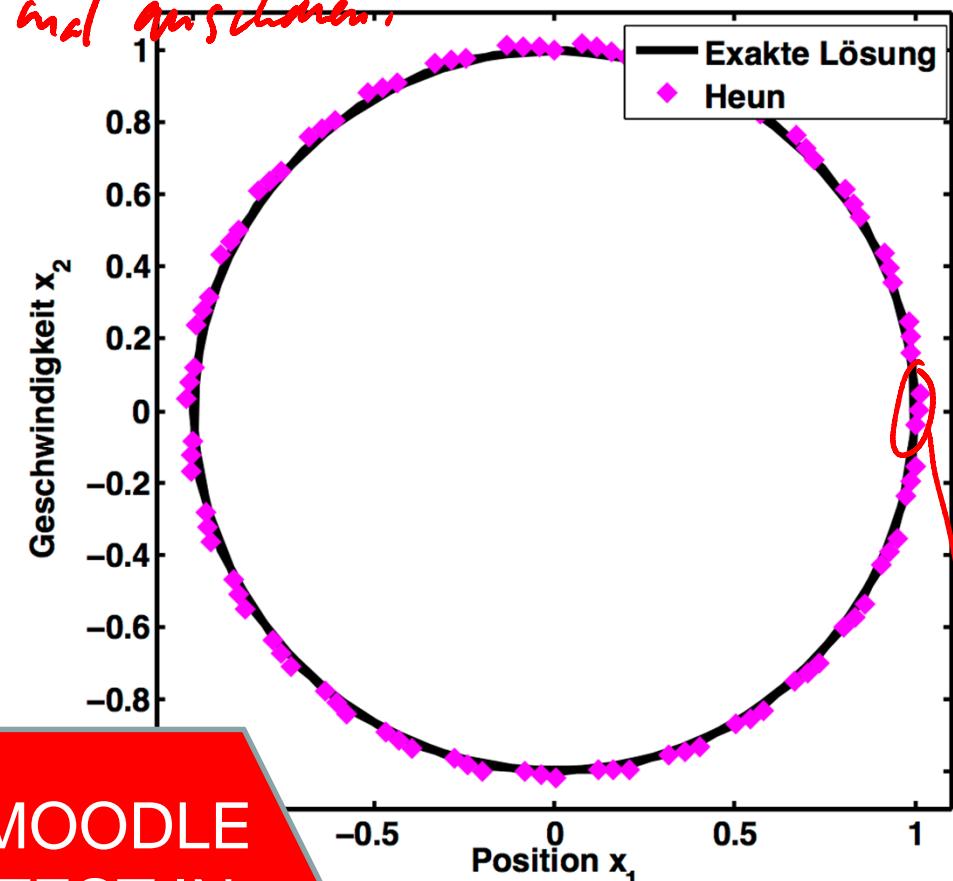
*Hier*

$$x(t_{i+1}) = \underbrace{x(t_i) + hAx(t_i)}_{\text{Expliziter Euler}} + \underbrace{\frac{h^2}{2}A^2x(t_i)}_{\text{Term höherer Ordnung... (aber nicht Taylor!)}} + \underbrace{\frac{h^3}{8}A^3x(t_i)}_{\text{Term höherer Ordnung... (aber nicht Taylor!)}} + \underbrace{\frac{h^4}{48}A^4x(t_i)}_{\text{Term höherer Ordnung... (aber nicht Taylor!)}} \text{ folgt.}$$



$$x(t) = A \exp(i\omega t + \varphi) \Rightarrow x(t) = A \sin(\omega t)$$

Bitte  
GLS hoch  
und ausschalten!



Höhere zeitl. Regularität!

MOODLE  
TEST IN  
ZWEI  
FOLIEN

### 3.6.4 Bemerkungen zu Verfahren höherer Ordnung

- Bei gegebener Schrittweite  $h_k$ 
  - Approximationsfehler für Methode höherer Ordnung ist kleiner
  - Berechnungsaufwand ist höher
- Die Auswahl des Integrationsverfahrens ist abhängig von den Genauigkeitsanforderungen
  - Je höhere Genauigkeit der Lösung benötigt wird, desto vorteilhafter sind Verfahren höherer Ordnung
  - Rechte Seite von  $f$  und Lösung  $x$  müssen ausreichend oft differenzierbar sein

### 3.6.4 Bemerkungen zu Verfahren höherer Ordnung

- Beispiele zu Genauigkeitsanforderungen
  - Bei (relativer) Genauigkeitsanforderung von  $10^{-4}$  eher Heun-Verfahren verwenden (2.Ordnung)
  - Bei  $10^{-8}$  besser Verfahren höherer Ordnung verwenden also z.B. RK-Verfahren 7. oder 8. Ordnung
  - Verfahren 1.Ordnung (Euler-Verfahren) oft nicht genau genug und können schnell von der Lösung divergieren (siehe Feder-Dämpfer Beispiel)
  - Ausnahme: symplektischer Euler bei mechanischen Systemen.
- Bei sehr großen und sehr kleinen Schrittweiten gilt vorangehende Aussage nicht uneingeschränkt
  - Bei zu kleinen Schrittweiten dominieren Rundungsfehler

Bitte jetzt auf Moodle eine Frage  
beantworten!

### 3.6.5 Einschrittverfahren: Butcher Tableau

- **Exkurs:** Das Butcher Tableau ist ein simpler Mechanismus, bei welchem sich die Gewichte der Zwischenschritte  $k_i$  der numerischen Integrationsverfahren feststellen lassen
  - Ist für explizite ESV eine strikte untere Dreiecksmatrix von der Form:

$c$	$A$	$b^T$
	$c_1 \mid a_{11}$ $c_2 \mid a_{12} \quad a_{22}$ $b_1 \quad b_2$	

$$x(t_{i+1}) = x(t_i) + \sum_{j=1}^n b_j k_j$$

$$k_j = f\left(t_i + c_j h, x(t_i) + \sum_{s=1}^{j-1} a_{sj} k_s\right)$$

$$\sum a_{ij} c_j = b_j$$

- Beispiel: Verfahren von Heun

0	0
1	1 0
	1/2 1/2

$$x(t_{i+1}) = x(t_i) + \frac{1}{2} h k_1 + \frac{1}{2} h k_2,$$

$$k_1 = f(t_i, x(t_i))$$

$$k_2 = f(t_i + 1 \cdot h, x(t_i) + 1 k_1)$$

### 3.7.1 Konvergenzordnung

- Die Konvergenzordnung  $p \in \mathbb{N}$  ergibt sich mit:

$$g(t, h) = x(t_i) - x_*(t_i) = O(h^p)$$

für die größte natürliche Zahl  $p$ .

$$g(t_i) = \sum_{k=-\infty}^{i-1} g(t_k)$$

*Aufschlüsselung*

- Hier ist  $g(t, h)$  der globale Diskretisierungsfehler,  $x(t_i)$  die Näherungslösung und  $x_*(t_i)$  die exakte Lösung zum Zeitpunkt  $t_i$
- Gibt an wie gut die exakte Lösung approximiert wird

*"Alle Fehler der Vergangenheit werden durch Integration gesammelt"*



### 3.7.2 Konsistenzordnung

- ESV ist von der Form  $x(t_{i+1}) = x(t_i) + h \cdot \Phi(t_i, x_i, x_{i+1}, h; f)$

- Wie gut approximiert die Zuwachsfunktion  $\Phi$  die Ableitung

$$\dot{x}(t_i) = f(t_i, x(t_i))?$$

- Betrachtung mithilfe des lokalen Diskretisierungsfehlers

$$l(t, h) = \frac{x(t_i + h_i) - x(t_i)}{h} - \Phi(t_i, x_i, x_{i+1}, h; f) = O(h^p)$$

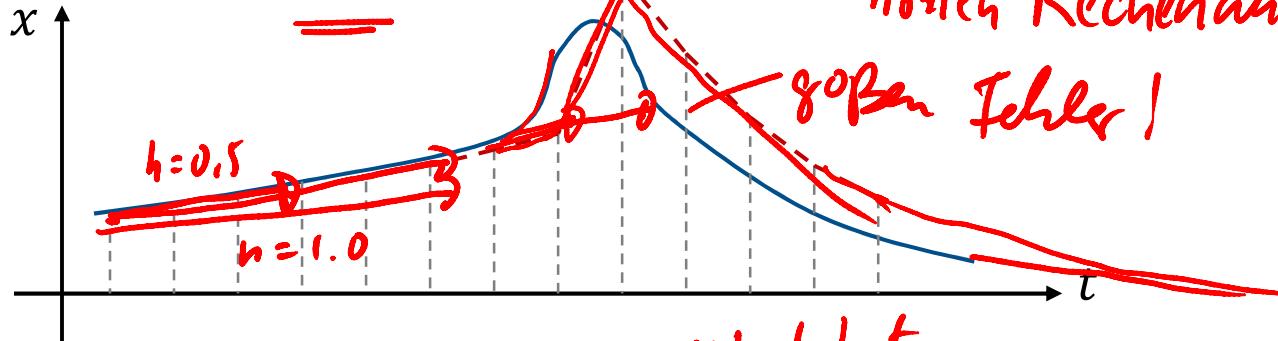
mit Konsistenzordnung  $p \in \mathbb{N}$

- Ein Verfahren ist konsistent mit Ordnung  $p$  (siehe auch Konsistenzbedingung)

## 3.8 Schrittweitensteuerung



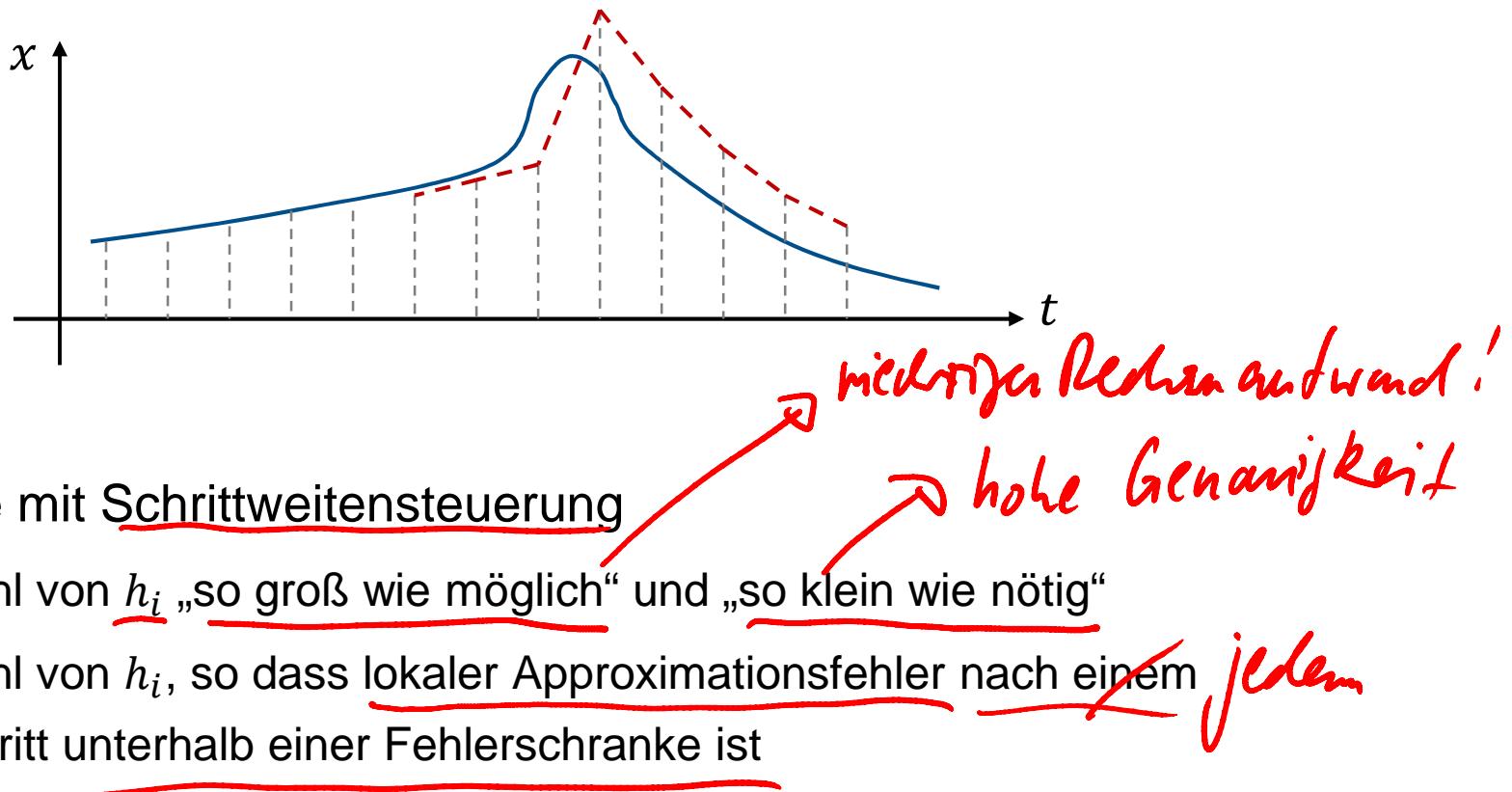
$$x(t_{i+1}) = x(t_i) + h_i g$$



- Konstante Schrittweite  $h_i$ 
  - Ungenau, wenn gesuchte Lösung sich lokal stark ändert
  - Ineffizient, wenn gesuchte Lösung sich lokal sehr wenig ändert

*Lineares oder Konstantes  $f \Rightarrow$  großes  $h$  lohnt sich!*

## 3.8 Schrittweitensteuerung



- Abhilfe mit Schrittweitensteuerung
  - Wahl von  $h_i$  „so groß wie möglich“ und „so klein wie nötig“
  - Wahl von  $h_i$ , so dass lokaler Approximationsfehler nach einem Schritt unterhalb einer Fehlerschranke ist ~~jedem~~

## 3.8 Schrittweitensteuerung: übliche Varianten (1)



- Man betrachte ein ESV der Ordnung  $p$  mit  $k$  Stufen
- Man berechne zwei Näherungen für  $x_{i+1}$ 
  - Einmal mit Schrittweite  $h_i$  und
  - Einmal mit zwei Schritten mit Schrittweite  $h_i/2$
- Daraus Schätzung des lokalen Fehlers
- Anpassung von  $h_i$  so dass lokale Fehlereinschätzung unter einer vorgegebenen Schranke liegt

} Sensitivität der Integrazon  
zu Veränderungen in  
der Abtastrate  $h$

## 3.8 Schrittweitensteuerung: übliche Varianten (1)



- Ist der lokale Fehler unterhalb der vorgegebenen Schranke
  - Akzeptiere den Schritt und nutze die mit  $h_i/2$  berechnete Lösung
  - Mache einen Schrittweitenvorschlag für den nächsten Schritt
- Ist der lokale Fehler oberhalb der vorgegebenen Schranke
  - Wiederhole den Schritt mit neuem Schrittweitenvorschlag  $h_i$
- Aufwand: Die rechte Seite von  $f$  wird  $s + 2s$  oft ausgewertet
  - 
  -

## 3.8 Schrittweitensteuerung: übliche Varianten (2)



- Man betrachte zwei Verfahren mit unterschiedlicher Ordnung
  - Z.B. mit der Ordnung  $p$  und  $p + 1$   
 $7, \quad 8. - RK$
- Berechnung von zwei Näherungen für  $x_{i+1}$  für dieselbe Schrittweite  $h_i$ 
  - Daraus schätzt man den lokalen Fehler
- Bei geeigneter „Einbettung“ der beiden Verfahren, unterscheiden sich die beiden Verfahren nur in den Koeffizienten und NICHT IHRE LÖSUNGEN!
  - z.B. als RK-Verfahren 4. und 5. Ordnung wodurch nur  $s + 1$  Funktionsauswertungen notwendig sind

## 3.8 Schrittweitensteuerung: übliche Varianten



- Ansatz zwei ist effizienter als der erste Ansatz
- Bei geringer Ordnung ist Ansatz eins aber robuster

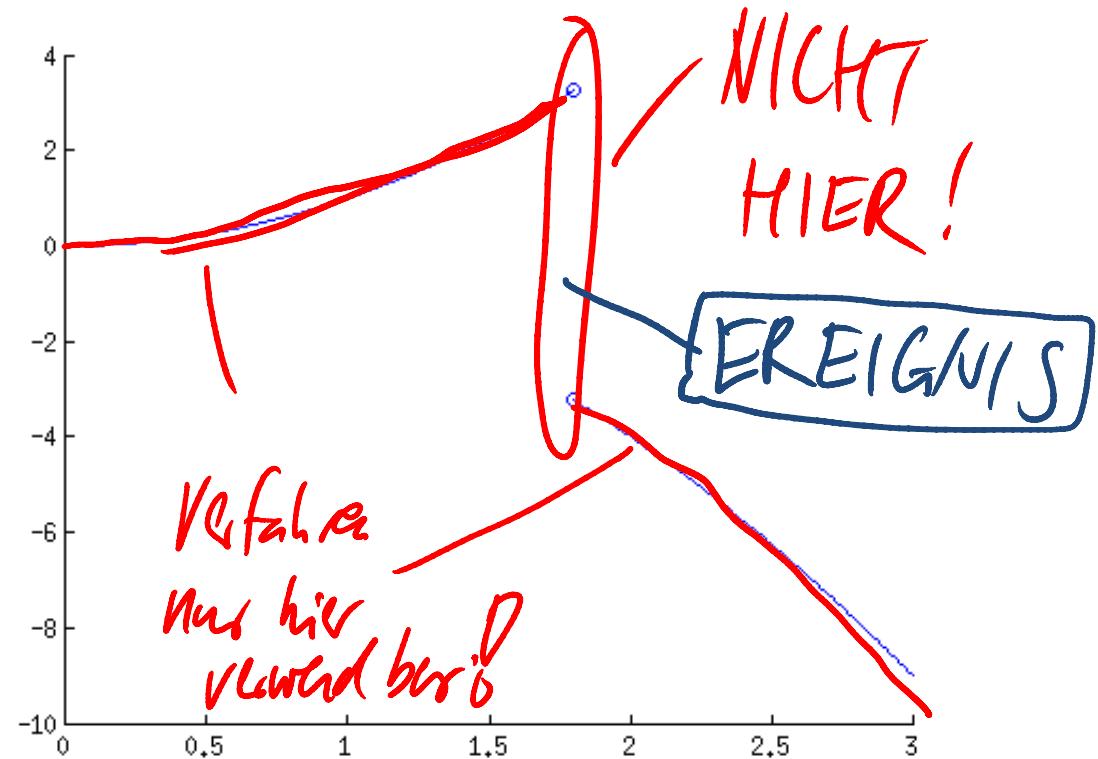
des Integrationsverfahrens

## 3.9 Integration von Unstetigkeiten



- Motivation: Integration von Funktionen mit Unstetigkeiten

- Was ist hier zu tun, wenn man so etwas mit numerischer Integration lösen möchte?



## 3.9 Integration von Unstetigkeiten



- Gegeben ist  $\dot{x}(t) = f(x, u, t)$  mit  $x(0) = x_0$   
 $f$
- Numerische Integrationsverfahren fordern, dass  $f$  genau  $p$  mal stetig differenzierbar ist
  - $p$  beschreibt hier die Konsistenzordnung
- Falls nicht gegeben, dann kann es zu massiven Genaugkeitsverlust in der errechneten Lösung kommen!

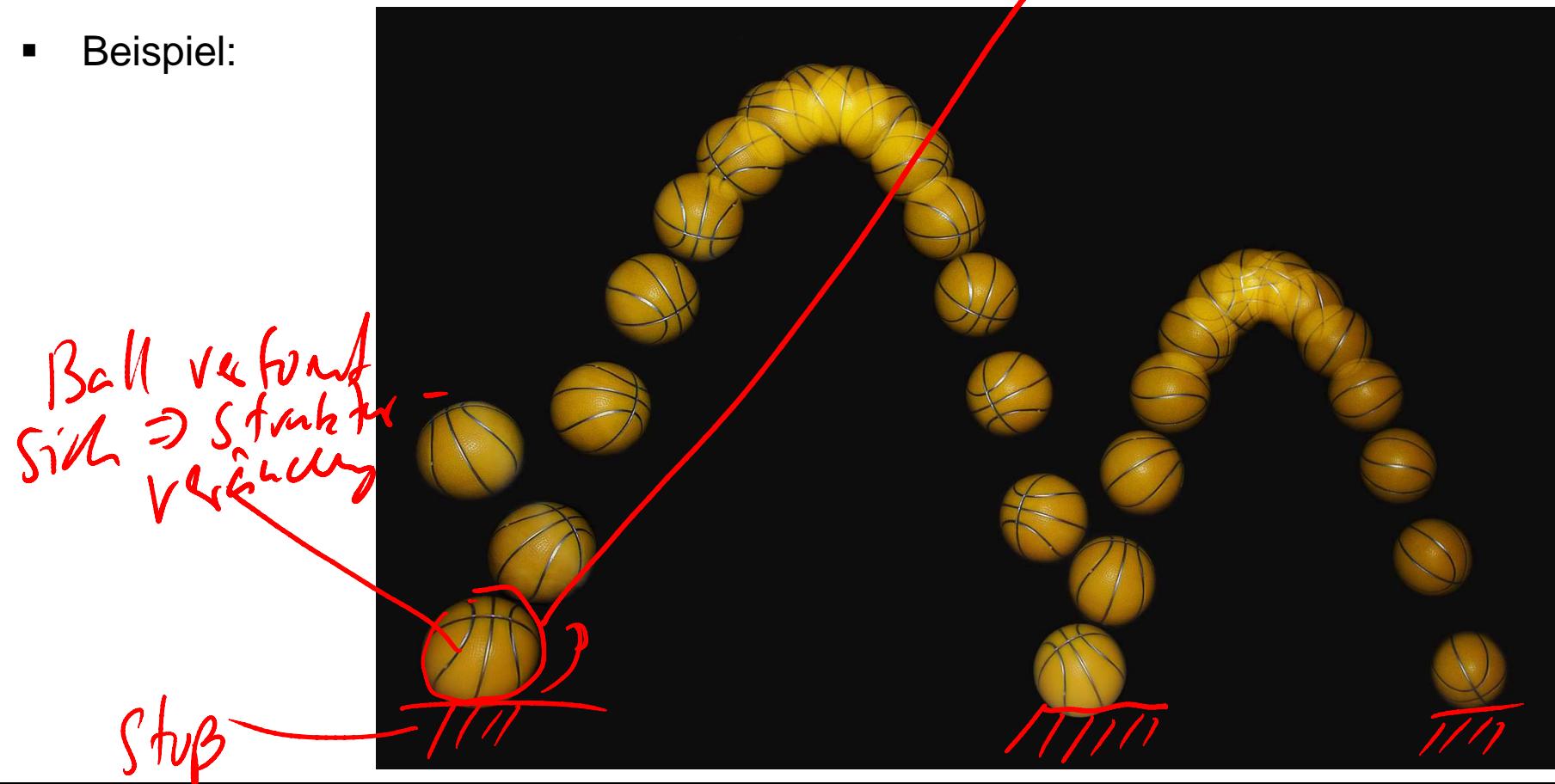
## 3.9 Integration von Unstetigkeiten



- Was für Lösungsmöglichkeiten gibt es?
- Man treffe folgende Annahmen:
  - Die Funktion ist abschnittsweise  $p$ -mal stetig differenzierbar
  - Die Unstetigkeitsstellen werden durch formulierbare Ereignisse (events) ausgelöst, die zustandsabhängig oder zeitgesteuert sein können

## 3.9 Integration von Unstetigkeiten

- Was sind Ursachen für solche Unstetigkeiten?
  - Beispiel:



## 3.9 Integration von Unstetigkeiten



- Was sind Ursachen für solche Unstetigkeiten?
- Stoßvorgänge
  - Stöße führen zu sprunghafter Änderung in der Geschwindigkeit
- Reibung in mechanischen Systemen "Stiction"
  - Übergänge zwischen Gleit- und Haftriebung sind meist unstetig
- Strukturvariable Systeme
  - Anzahl der Dimensionen des Zustands  $x$  ändert sich

Position, Orientierung + Geschwindigkeit  
+ Verformung

### 3.9.1 Hysterese: Beispiel



- Definition: Ein Teil von  $f$  hängt von der aktuellen Vorgeschichte von  $x$  ab

$$\dot{x}(t) = A(x(t), \underline{x(-\infty : t)})$$

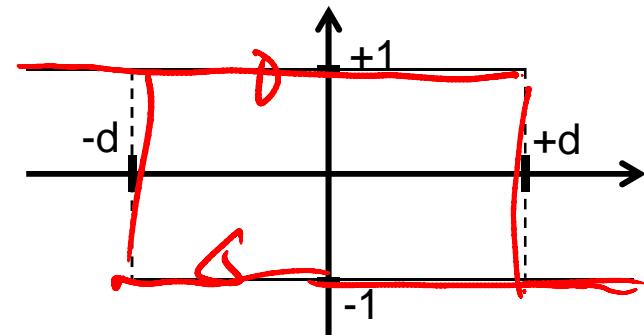
Vorgeschichte!

- Verwendung von zeitabhängigen Eingangsfunktionen / Stellgrößen  $u$ 
  - Zeitabhängige Eingangsfunktion mit Unstetigkeiten zu bekannten Zeitpunkten (Zeitereignisse)
  - Oder digitale Regelung: während eines „Abtastschritts“ ändert sich konstante Stellgröße sprunghaft an den Grenzen der Abtastintervalle

### 3.9.1 Hysterese: Beispiel

- Gegeben sei die skalare Zustands-DGL:  $\dot{x} = u + H(x)$  mit

$$H(x) = \begin{cases} +1, & x \leq +d \\ -1, & x \geq -d \end{cases}$$

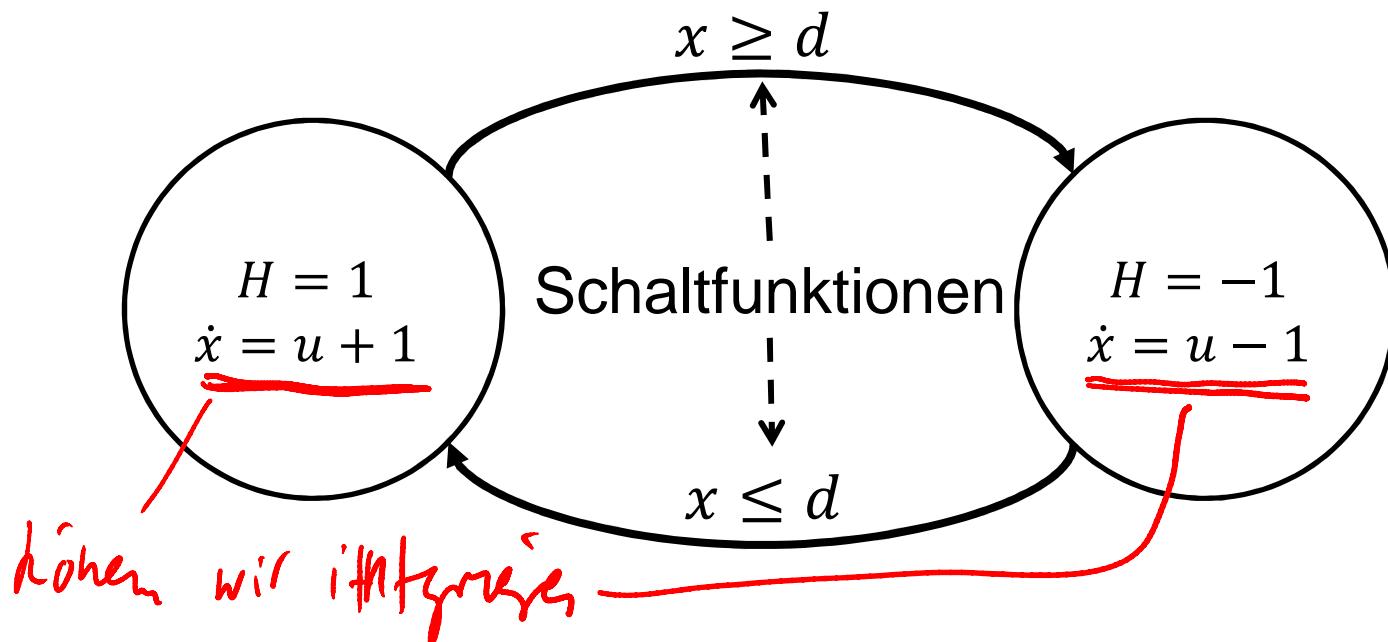


- Hysterese-Element mit „Gedächtnis“ der Vorgeschichte von  $x$
- Was passiert beim Erreichen der Abschnittsgrenze  $d$ ? Wie beschreibt man dieses Verhalten?
- Zusätzliche Schaltbedingung für Schaltfunktion  $u$  zur Systembeschreibung notwendig!

### 3.9.1 Hysterese: Beispiel

Hybrides System = Discrete  
Event System + DGLs

- Beschreibung der Schaltfunktion  $u$  durch diskreten Zustandsautomaten der von den kontinuierlichen Zustandsvariablen abhängt (häufig aber nicht immer möglich)



### 3.9.2 Schaltfunktionen

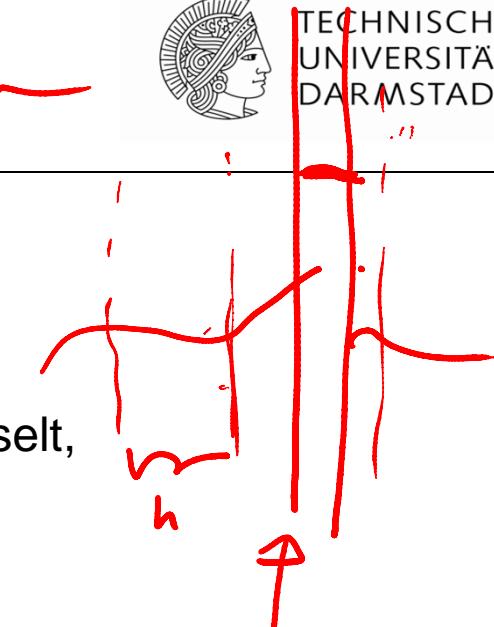


- Schaltzeitpunkte (events)  $t_{s,i}$  mit  $i = 1, \dots, n_s$  werden im allgemeinen als (einfache) Nullstellen  $n_q$  reell wertiger Schaltfunktionen modelliert
  - Schaltfunktion gegeben durch  $q_l(x(t_{s,i}), t_{s,i})$  mit  $l \in \{1, \dots, n_q\}$   
(z.B.  $q_1 = d - x(t)$ ,  $q_2 = d + x(t)$ )
- Betrachtung der Vorzeichen der Schaltfunktionen bei numerischer Integration

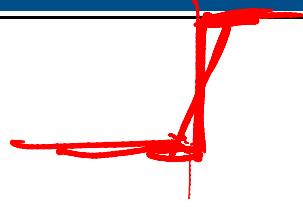
### 3.9.2 Schaltfunktionen



- Betrachtung der Vorzeichen der Schaltfunktionen bei numerischer Integration
  - Wenn zwischen  $x(t_{i+1})$  und  $x(t_i)$  sich das Vorzeichen wechselt, dann muss der Schaltzeitpunkt  $t_{s,i}$  ermittelt werden
  - Genauigkeit des ermittelten Schaltpunktes gegeben durch Wahl des Verfahrens und Anforderung des Nutzers
- Voraussetzung: Die Schrittweite  $h_i$  muss klein genug sein um den Vorzeichenwechsel zu detektieren. Dieser kann sonst unbemerkt bleiben! (z.B. bei doppelter Vorzeichenwechsel)



### 3.9.2 Schaltfunktionen



- Voraussetzung: Die Schrittweite  $h_i$  muss klein genug sein um den Vorzeichenwechsel zu detektieren. Dieser kann sonst unbemerkt bleiben! (z.B. bei doppelter Vorzeichenwechsel)
  
- Was passiert, wenn man einfach über Schaltpunkte drüber integriert?
  - Integrator mit konstanter Schrittweite merkt nichts, deshalb wird die Lösung nicht gut sein
  - Integrator mit Schrittweitensteuerung kann eine bessere Lösung erzielen (abhängig von Nutzereingaben)



### 3.9.2 Schwinger mit Schaltstellen: Beispiel



- System beschrieben durch die Bewegungs-DGL

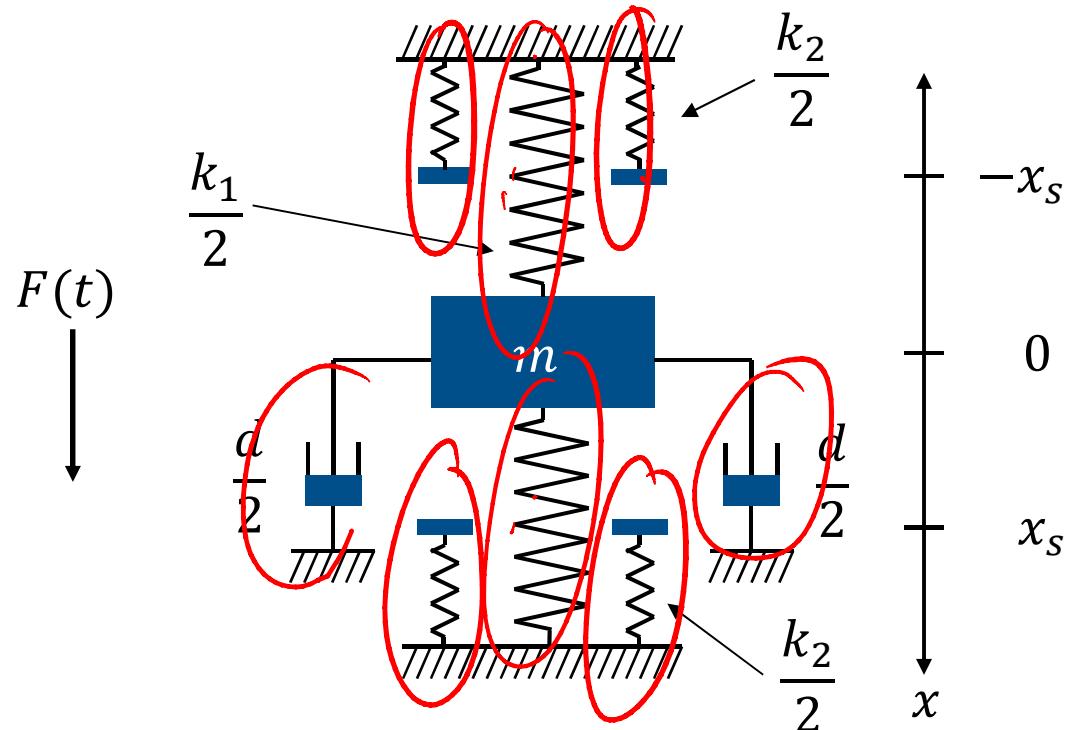
$$m\ddot{x} + d\dot{x} + h(x) = F(t)$$

mit anregender Kraft

$$F(t) = g \cdot \cos(\omega t)$$

und Auslenkungswinkel  $\omega$

(Pietruszka, 2005)



### 3.9.2 Schwinger mit Schaltstellen: Beispiel



- Die Federkraft ist abschnittsweise linear und gegeben mit

$$\underline{h(x)} = \begin{cases} k_1 x, & \text{für } -x_s \leq x \leq x_s \\ k_1 x + k_2 (x - x_s \cdot \underline{\operatorname{sgn}(x)}), & \text{für } x_s < |x| \end{cases}$$

- Modellierung der Schaltfunktion mit  $\underline{q = x - x_s \cdot \operatorname{sgn}(x)}$  oder

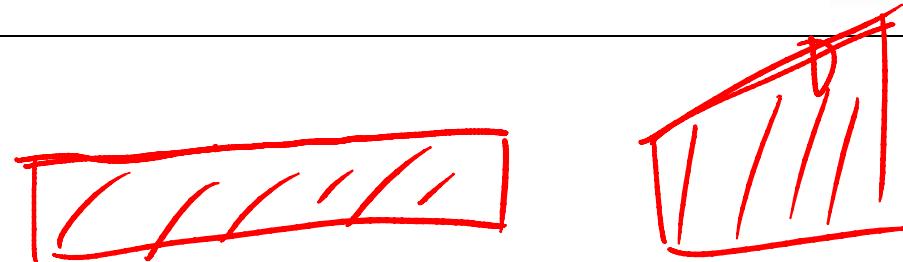
$$q = \underline{|x| - x_s}$$

Ungeeignet wegen Sprüngen um  $x \approx 0$

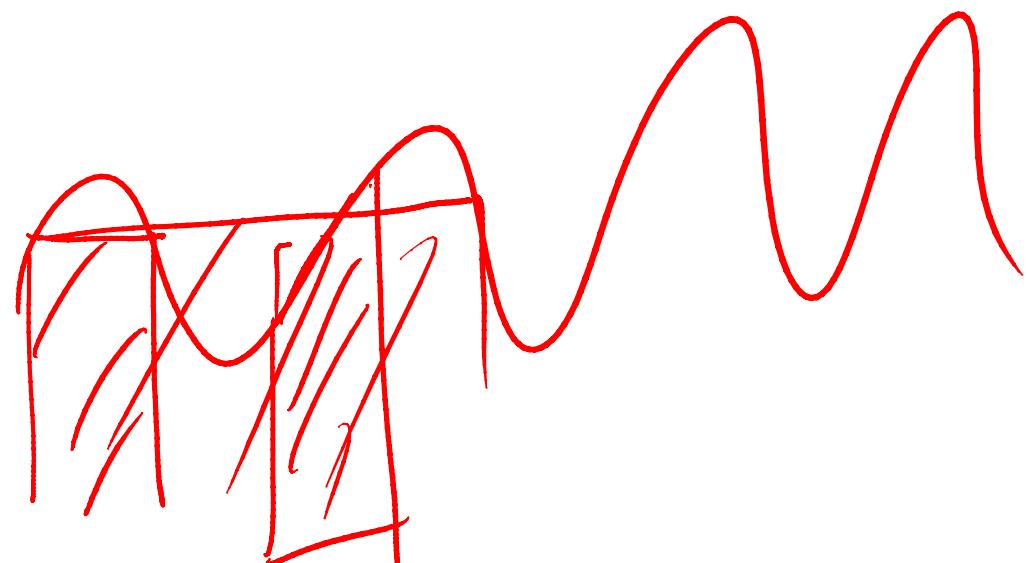
# MOODLE FRAGE



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



Bitte jetzt auf Moodle eine Frage  
beantworten!



# Heutige Lernziele: Kernfragen



- Was ist ein symplektischer Euler? Wie verhalten sich impliziter, expliziter und symplektischer Euler am periodischen System? ✓
- Was sind Prädikator-Korrektor Verfahren? ↗
- Was ist das Verfahren von Heun? ✓ Wie kann man es intuitiv interpretieren? ✓  
Warum ist es ein Prädikator-Korrektor Verfahren? ↗
- Was ist das Runge-Kutta Verfahren? Welche Ordnung hat es?
- Was sind Konvergenz- und Konsistenzordnung? ↗
- Wie funktioniert hier Schrittweitensteuerung? Welche Varianten gibt es? ↗
- Wie kann man mit Differentialgleichungen mit Unstetigkeiten umgehen?
- **Selbsttest:** Können Sie diese Fragen beantworten? Wenn nicht, schnell nochmal das Video anschauen!