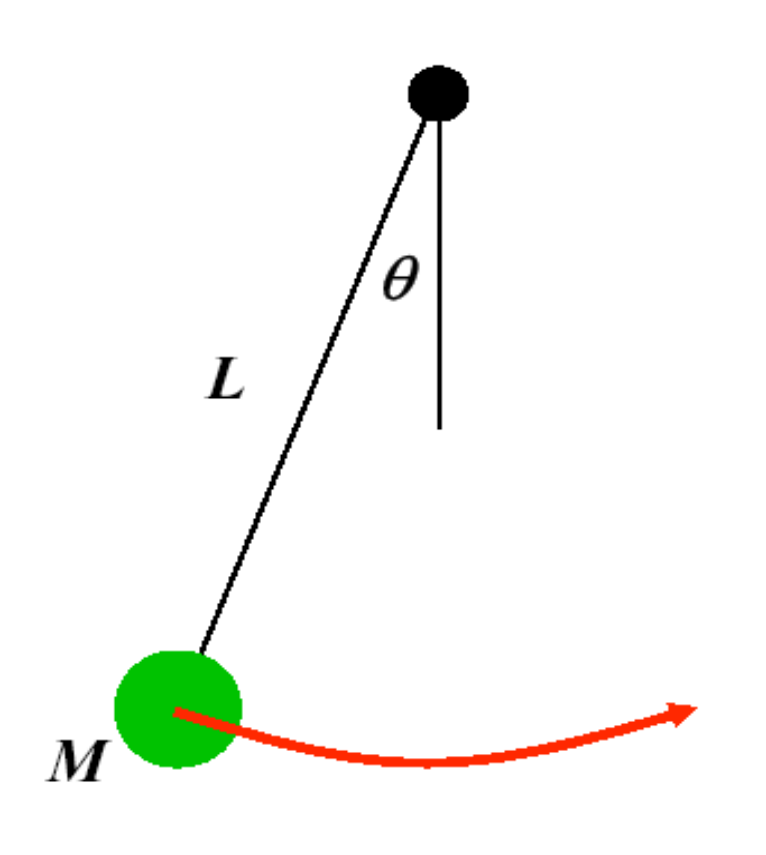
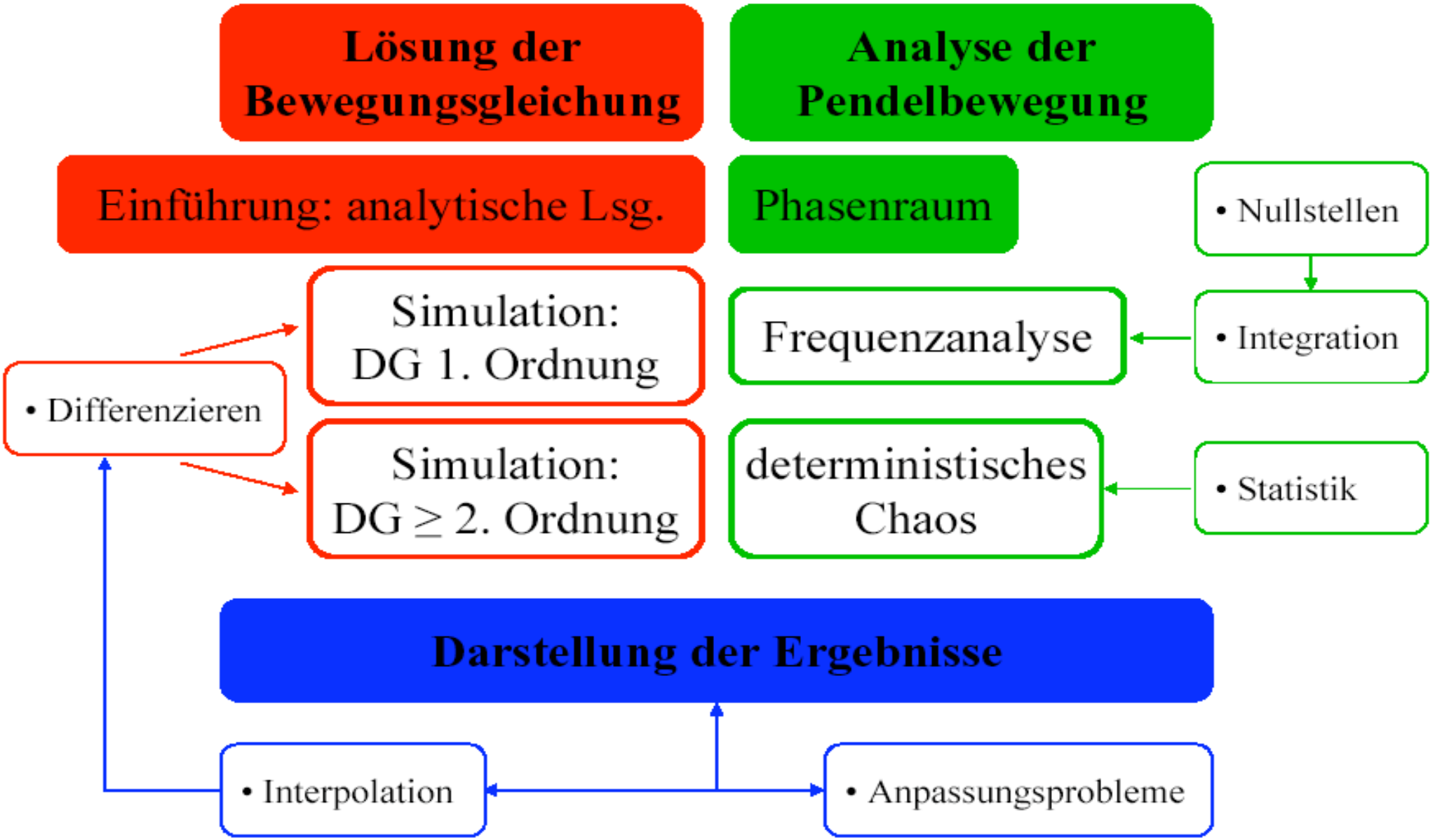
Das Pendel



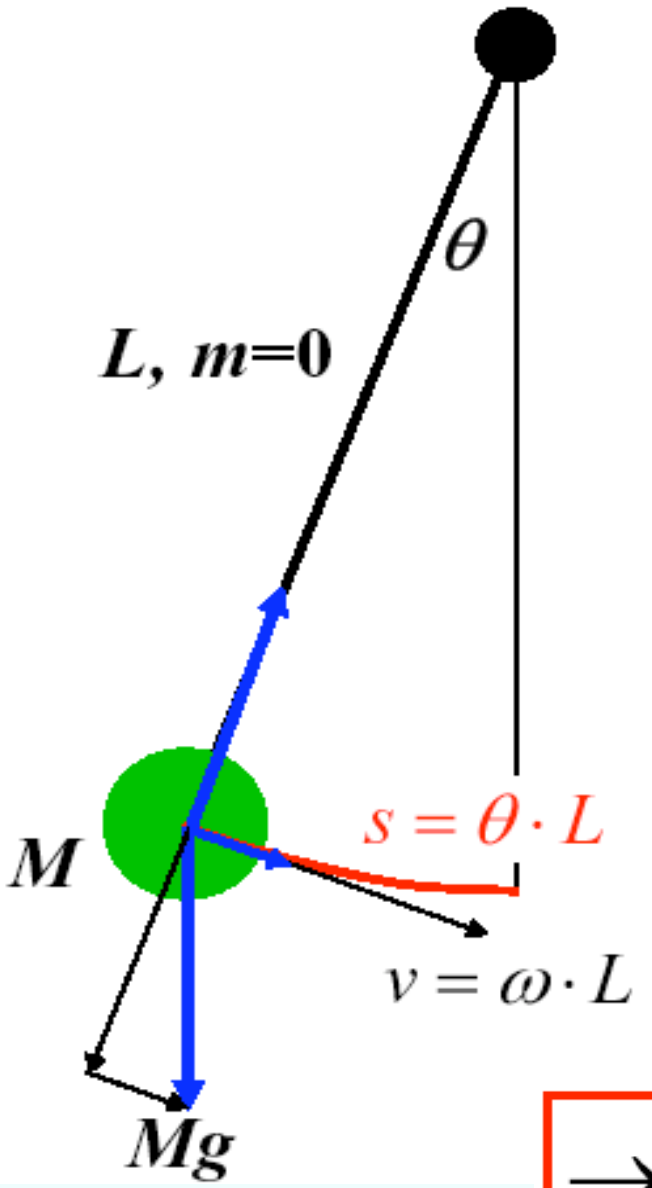
Simulation und analyse

- Das Physikalische Pendel
- Phasenraum
- Gedämpfte Schwingungen
- Frequenzanalyse

Das Pendel


Lösung der Bewegungsgleichung
Analyse der Pendelbewegung
Einführung: analytische Lsg.
Phasenraum
• Nullstellen
Simulation: DG 1. Ordnung
Frequenzanalyse
• Integration
• Differenzieren
Simulation: DG ≥ 2. Ordnung
deterministisches Chaos
• Statistik
Darstellung der Ergebnisse
• Interpolation
• Anpassungsprobleme

Das physikalische Pendel



Bewegungsgleichung

- masseloser Stab, Länge L
- punktförmige Masse M
- reibungslose Aufhängung

- Längskomponente der Schwerkraft wird vom Stab kompensiert
- Querkomponente beschleunigt die Masse

Bogenlänge: $s = \theta \cdot L$

Geschwindigkeit: $v = \dot{\theta} \cdot L = \omega \cdot L$

Beschleunigung: $a = \ddot{\theta} \cdot L = \dot{\omega} \cdot L$

$$\rightarrow \quad F = Ma = ML\ddot{\theta} = -Mg\sin(\theta)$$

Das physikalische Pendel

Lösung der linearisierten Bewegungsgleichung
(gültig für kleine Auslenkungen $\theta << 1$)

Für größere Auslenkungen: nichtlineare DG=>
deterministisches Chaos

$$ML\ddot{\theta} = -Mg\sin(\theta)$$

$$\rightarrow \ddot{\theta} = -\frac{g}{L}\sin(\theta) = -\frac{g}{L}\left(\theta - \theta^3/3! + \theta^5/5! + \ldots\right)$$

$$\rightarrow \ddot{\theta} \approx -\frac{g}{L}\cdot\theta = -\omega_0^2\cdot\theta \quad \text{mit} \quad \omega_0 = \sqrt{g/L}$$

Lösung:

$$\theta = \theta_0 \cdot \sin\left(\omega_0 t + \varphi\right)$$

Randbedingungen!

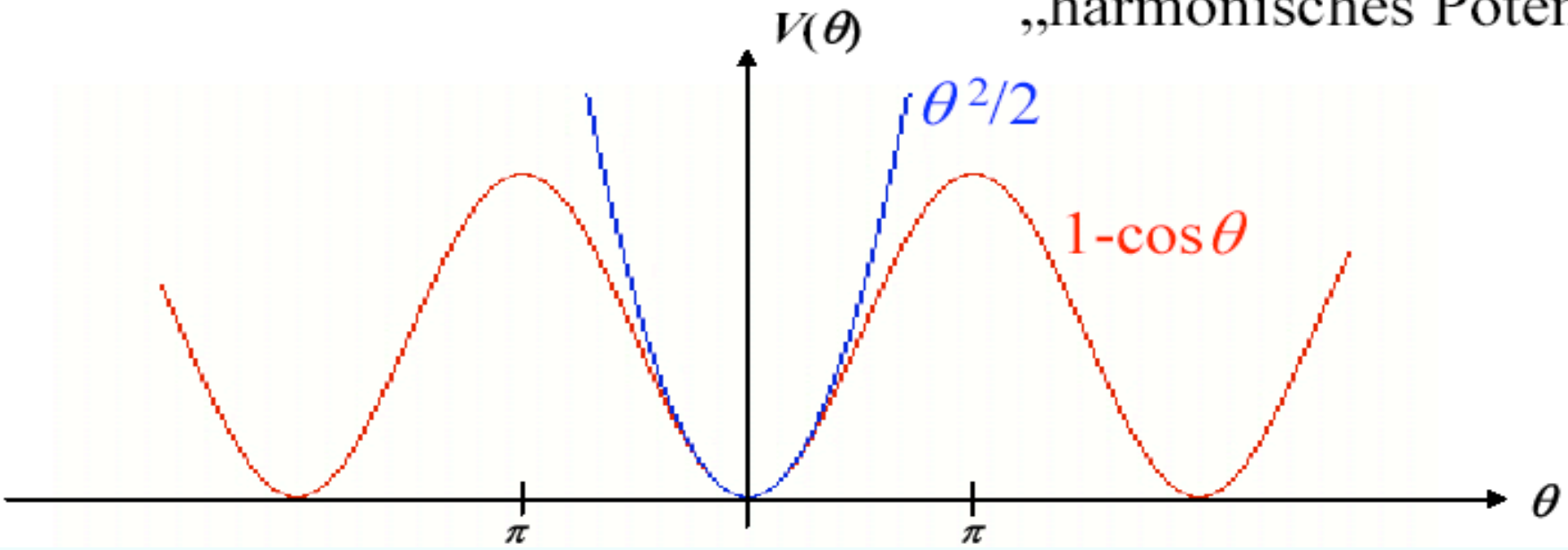
Energieerhaltung

$$H = E_{kin} + E_{pot} = T + V = E = \text{const.}$$ ⟵ „konservatives System“

$$= \frac{Mv^2}{2} + Mgh = \frac{1}{2} ML^2\omega^2 + MgL\left(1-\cos\theta\right)$$

$$= \frac{p_\theta^2}{2ML^2} + MgL\left(1-\cos\theta\right) \approx \frac{p_\theta^2}{2ML^2} + \frac{1}{2} M\omega_0^2 L^2\theta^2$$

„harmonisches Potential“



Der Phasenraum

Darstellung der Energie als Funktion der verallgemeinerten Orts- (q_i) und Impuls-(p_i) koordinaten

$$H(q_i, p_i) = T + V \quad \rightarrow \quad E$$

i :Freiheitsgrad

$H(q_i, p_i)$:Hamiltonfunktion

Dynamik des Systems beschrieben durch Hamilton-gleichung:

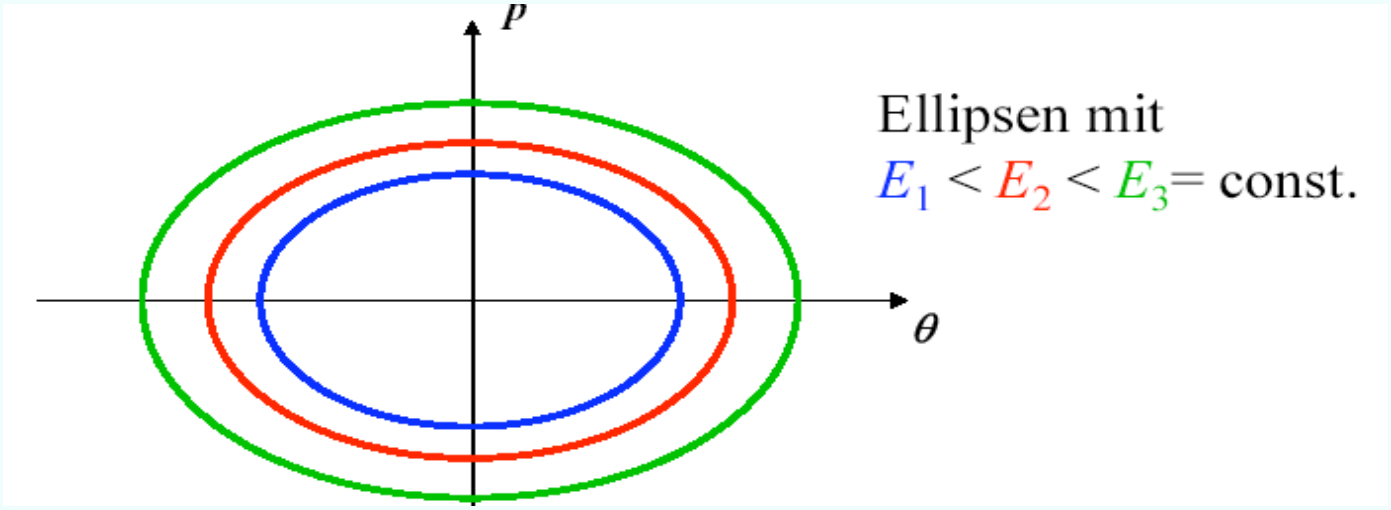
$$\dot{q}_i = \frac{\partial H}{\partial p_i}, \quad \dot{p}_i = -\frac{\partial H}{\partial q_i}; \qquad i = 1, \ldots, M$$

Der Phasenraum

Hamiltonfunktion der linearisierten Pendelbewegung:

$$H(p_\theta, \theta) = \frac{p_\theta^2}{2ML^2} + \frac{1}{2} M \omega_0^2 L^2 \theta^2 = E$$

Ellipse mit $a = p_\theta^{\max} = \sqrt{2ML^2 E}, \quad b = \theta^{\max} = \sqrt{\frac{2E}{M\omega_0^2 L^2}}$



Das gedämpfte Pendel

Linearisierte Bewegungsgleichung:

$$ML\ddot{\theta} + \gamma L\dot{\theta} + Mg\theta = 0 \quad \rightarrow \quad \ddot{\theta} + \frac{1}{\tau}\dot{\theta} + \omega_0^2\theta = 0; \quad \tau = \frac{M}{\gamma}$$

Ansatz:

$$\theta = \theta_0 \cdot e^{-\beta t} \cdot \sin(\omega t + \varphi)$$

$$\beta = \frac{1}{2\tau}, \quad \omega = \sqrt{\omega_0^2 - \beta^2} = \omega_0\sqrt{1 - \left(\frac{1}{2\tau\omega_0}\right)^2}$$

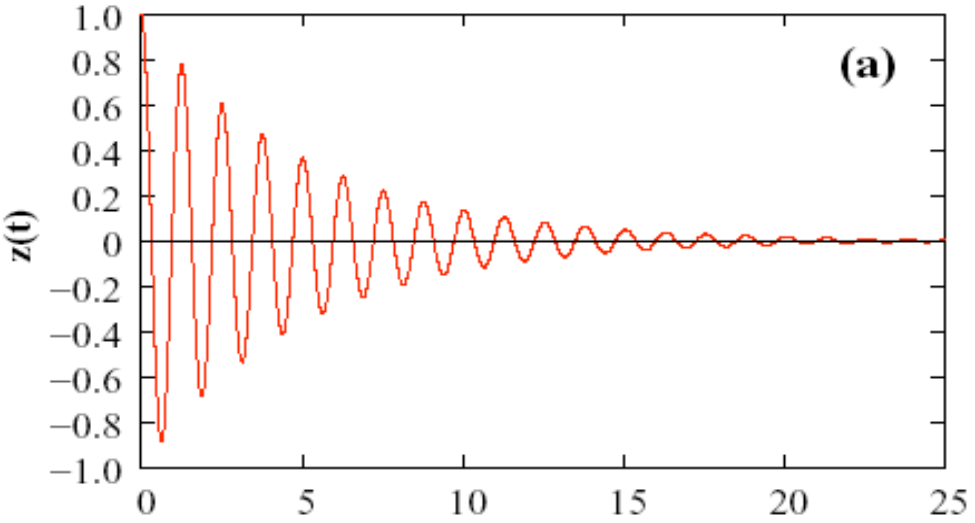
Schwache Dämpfung $(\omega_0\tau \gg 1)$

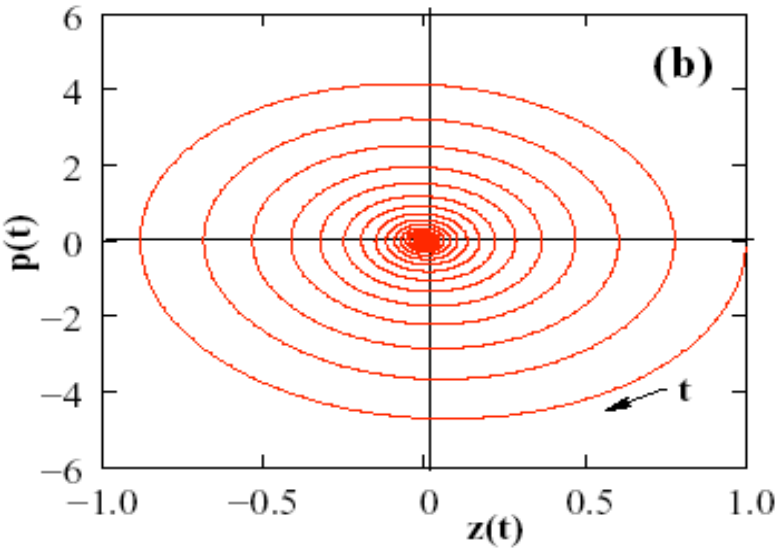
$$\theta \approx \theta_0 \cdot e^{-t/2\tau} \cdot \sin(\omega_0 t + \varphi)$$

Das gedämpfte Pendel

Zeitliche Entwicklung

Phasenraum





$$\theta \approx \theta_0 \cdot e^{-t/2\tau} \cdot \sin\left(\omega_0 t + \varphi\right)$$

Frequenzanalyse

Fouriertransformation: $F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{i\omega t} \cdot dt$

$f(t)$ meistens reellwertig: dann gilt $F(-\omega) = F^*(\omega)$

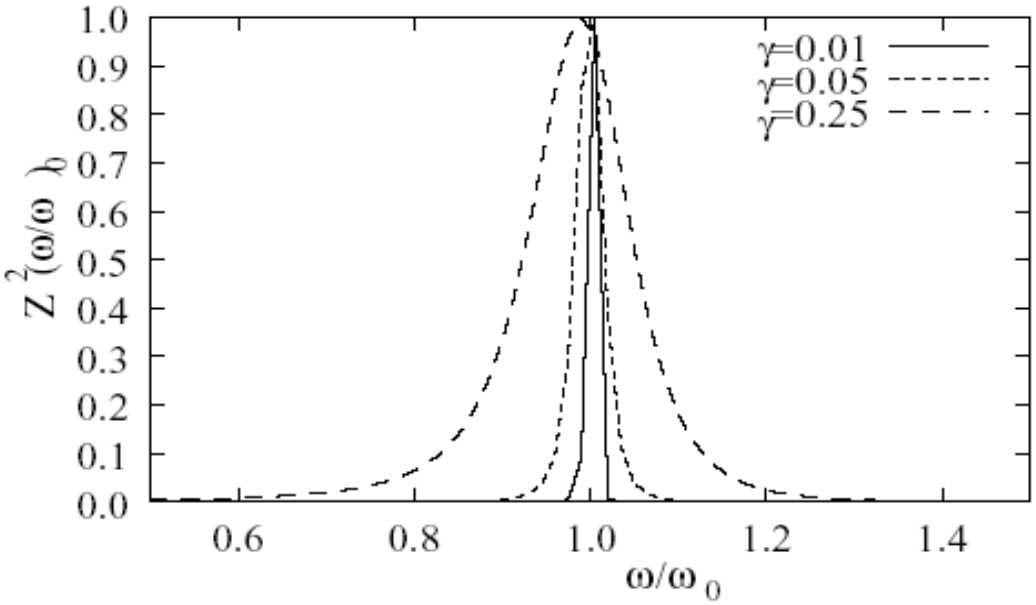
linearisiertes Pendel: $F(\omega) = i\pi\theta_0 \left[e^{-i\alpha}\delta(\omega - \omega_0) - e^{i\alpha}\delta(\omega + \omega_0) \right]$

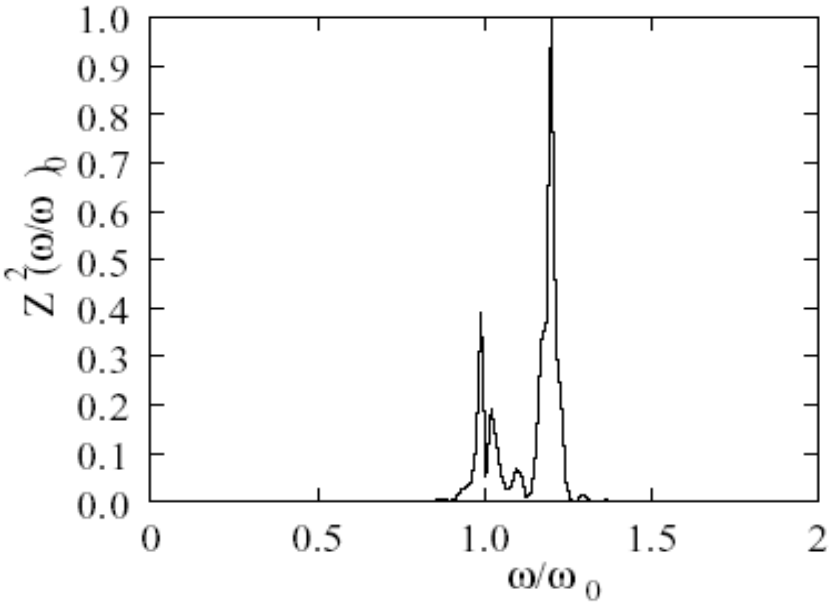
gedämpftes Pendel: Verbreiterung, Verschiebung

$$F(\omega) = \frac{1}{2}\theta_0 \left[\frac{e^{i\alpha}}{(\omega + \omega_0) + i\gamma} - \frac{e^{-i\alpha}}{(\omega - \omega_0) + i\gamma} \right]$$

Anm: “ $\alpha = \phi$ “

Frequenzspektrum

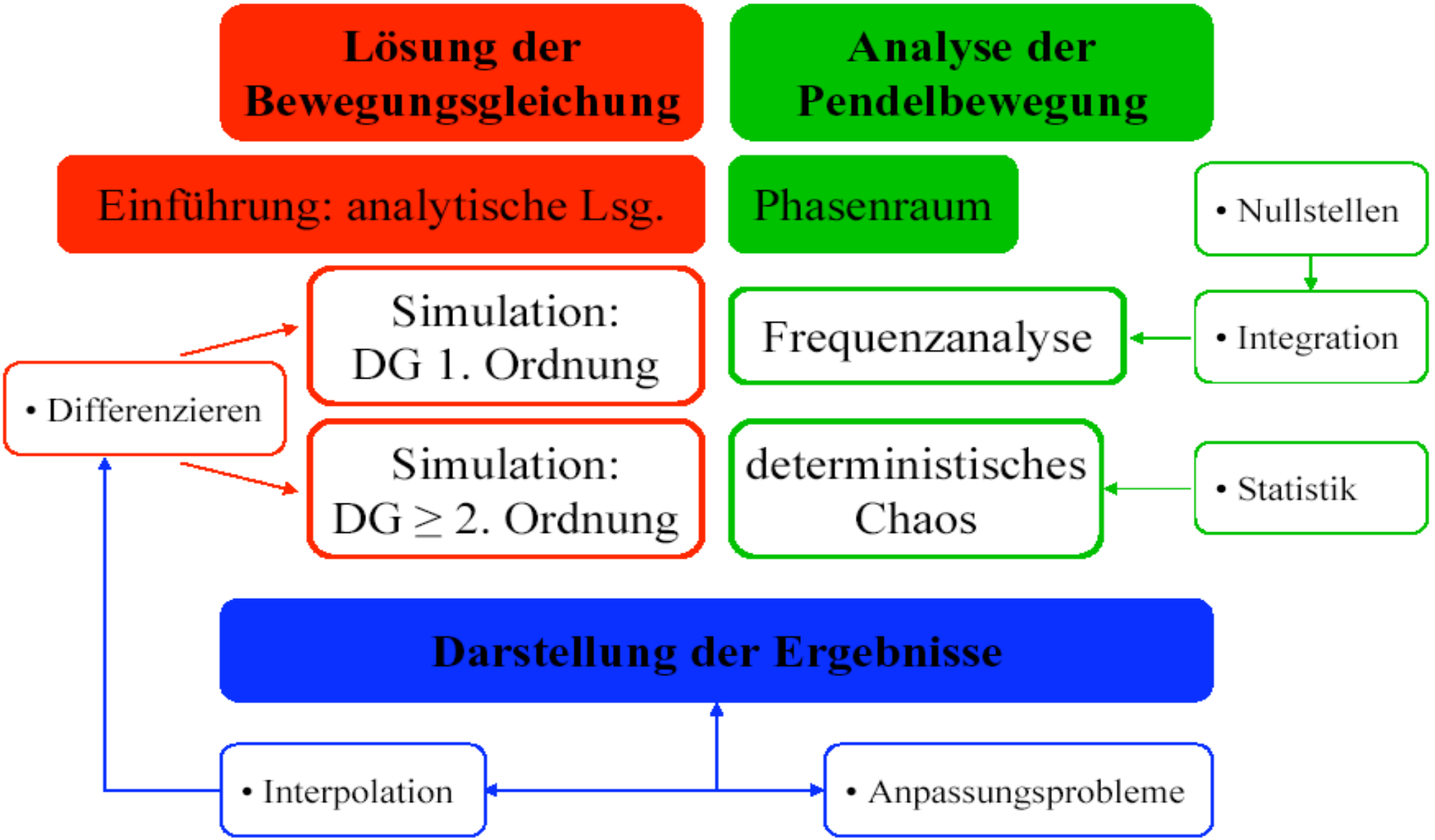




Frei, gedämpft

getrieben

Das Pendel


Lösung der Bewegungsgleichung
Analyse der Pendelbewegung
Einführung: analytische Lsg.
Phasenraum
• Nullstellen
Simulation: DG 1. Ordnung
Frequenzanalyse
• Integration
• Differenzieren
Simulation: DG ≥ 2. Ordnung
deterministisches Chaos
• Statistik
Darstellung der Ergebnisse
• Interpolation
• Anpassungsprobleme

Interpolation & Differentiation

Interpolation

- Wenig Messpunkte
- Übergang auf anderes Gitter von Datenpunkten
- Interpolationsfunktionen können einfach differenziert und integriert werden.

Interpolationsproblem:

Stützstellen: $x_i,\ f_i = f(x_i) \quad i = 0, 1, \ldots, N$

Funktionsklasse: $\phi(x; a_0, a_1, \ldots, a_N)$

Wähle die Paramete a_i so,
dass die Funktion durch alle Stützstellen geht :

$$\phi(x_i; a_0, a_1, \ldots, a_N) = f_i \quad \forall\ i = 0, 1, \ldots, N.$$

EDV2: Polynomialinterpolation (linear)

Polynominterpolation

- $(N+1)$ beliebig vorgegebene Stützpunkte

$(x_i, f_i), \quad i = 0, 1, 2, \ldots, N$

•Suche ein Polynom $p(x_i) = f_i$ für $i = 0, 1, 2, \ldots, N$

$$\phi(x; a_0, a_1, \ldots, a_N) = p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_N x^N \, .$$

welches die Bedingung

$$\phi(x_i; a_0, a_1, \ldots, a_N) = f_i \quad \forall \; i = 0, 1, \ldots, N.$$

erfüllt.

•Problem ist eindeutig

•z.B. Lagrange Interpolationsformel

$$p(x) = \sum_{i=0}^{N} f_i L_i(x)$$

$$L_i(x) = \prod_{\substack{k=0 \\ k \neq i}}^{N} \frac{x - x_k}{x_i - x_k} \quad \text{für} \quad i = 0, 1, \ldots, N \, .$$

Polynominterpolation

Beispiel Lagrange Interpolationsformel , N=3:

$\{x_i, f(x_i)\},\ i = 0, \ldots 3$

$$p_3(x) = \frac{(x-x_1)\ (x-x_2)\ (x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} f(x_0) + \frac{(x-x_0)\ (x-x_2)\ (x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} f(x_1)$$

$$+ \frac{(x-x_0)\ (x-x_1)\ (x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} f(x_2) + \frac{(x-x_0)\ (x-x_1)\ (x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} f(x_3).$$

- Interpolationsfehler wenn die zu interpolierende Funktion selbst kein Polynom ist.
- Effizientere Algorithmen existieren

Algorithmus von Neville
(Interpolation an *einem* zusätzlichen Punkt)

ausgehend von Polynomen 0. Grades

$$P_i(x) = f_i \quad \forall\, x$$

Definition einer Rekursion

$$P_{i_0 i_1 \ldots i_k}(x) = \frac{(x - x_{i_0}) P_{i_1 i_2 \ldots i_k}(x) - (x - x_{i_k}) P_{i_0 i_1 \ldots i_{k-1}}(x)}{x_{i_k} - x_{i_0}}$$

$\rightarrow$ bei N Datenpunkten

$$p_N(x) = P_{i_0 i_1 \ldots i_N}(x)$$

Polynominterpolation: Neville Algorithmus

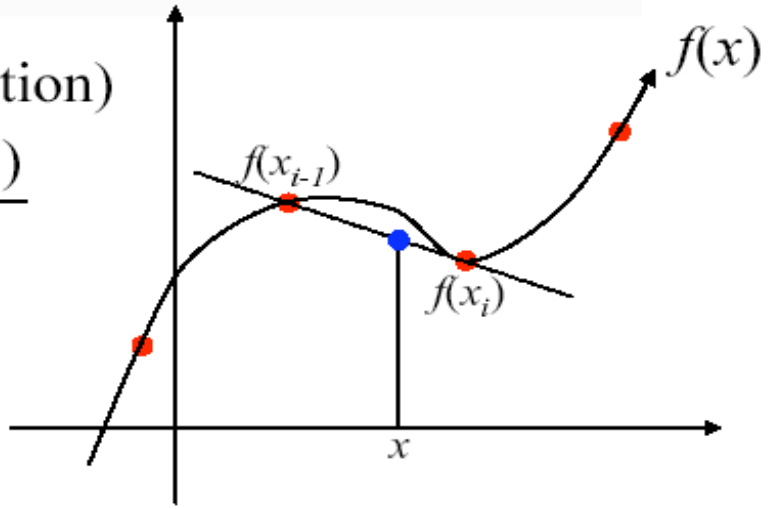
schematisch:

	$k=0$	$k=1$	$k=2$	$k=3$
x_0	$f_0 = P_0(x)$			
		$P_{01}(x)$		
x_1	$f_1 = P_1(x)$		$P_{012}(x)$	
		$P_{12}(x)$		$P_{0123}(x)$
x_2	$f_2 = P_2(x)$		$P_{123}(x)$	
		$P_{23}(x)$		
x_3	$f_3 = P_3(x)$			

Beispiel: zwei Punkte (lineare Interpolation)

$$f(x) = f(x_{i-1}) + (x - x_{i-1})\frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

$$= \frac{(x - x_{i-1})f(x_i) - (x - x_i)f(x_{i-1})}{x_i - x_{i-1}}$$



Newtonsche Interpolationsformel

Interpolationspolynom $p_N(x)$ durch N+1 Stützstellen $(x_i, f(x_i))$:

„dividierte Differenz“:

$$f[x_i, x_j] = \frac{f_i - f_j}{x_i - x_j} = f[x_j, x_i]$$

Definition einer Rekursion

$$f[x_0, x_1, \ldots, x_i] = \frac{f[x_1, \ldots, x_i] - f[x_0, x_1, \ldots, x_{i-1}]}{x_i - x_0}$$

gesuchtes Polynom

$$p_N(x) = a_0 + \sum_{k=1}^{N} a_k \cdot \left(\prod_{i=1}^{k} (x - x_{i-1}) \right); \quad a_k = f[x_0, x_1, \ldots, x_k]$$

Newtonsche Interpolationsformel

i	x_i	f_i	$f[x_i, x_{i+1}]$	$f[x_i, \ldots, x_{i+2}]$	$f[x_i, \ldots, x_{i+3}]$	$f[x_i, \ldots, x_{i+4}]$
0	3.2	22.0				
			8.400			
1	2.7	17.8		2.856		
			2.118		-0.5280	
2	1.0	14.2		2.012		0.256
			6.342		0.0865	
3	4.8	38.3		2.263		
			16.750			
4	5.6	51.7				

$$p_N(x) = a_0 + (x - x_0)a_1 + (x - x_0)(x - x_1)a_2 + \cdots$$
$$\cdots + (x - x_0)(x - x_1)\ldots(x - x_{N-1})a_N$$

Differentiation

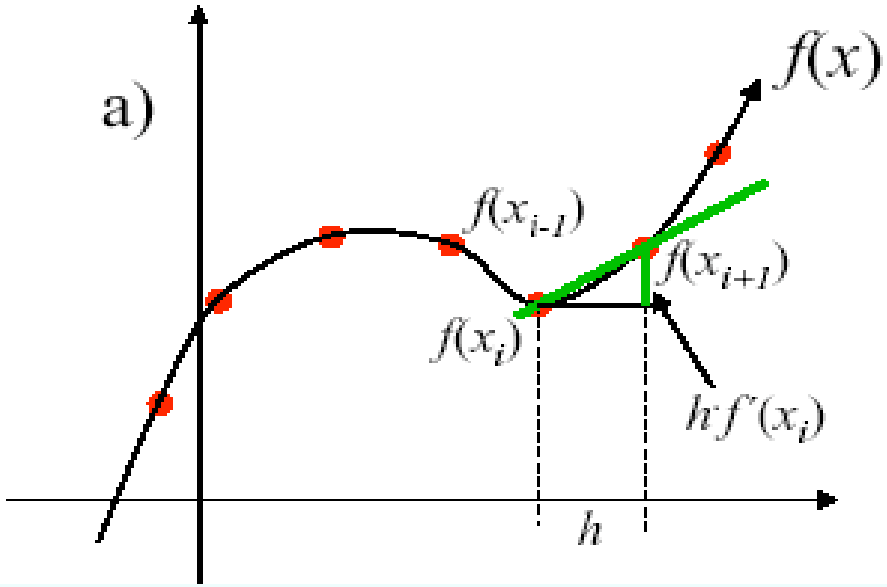
- Ableitung von Interpolationspolynom
- gleichmäßig verteilte Datenpunkte
- symmetrisch, asymmetrisch
- höhere Ordnung genauer, aber Randpunkte problematisch

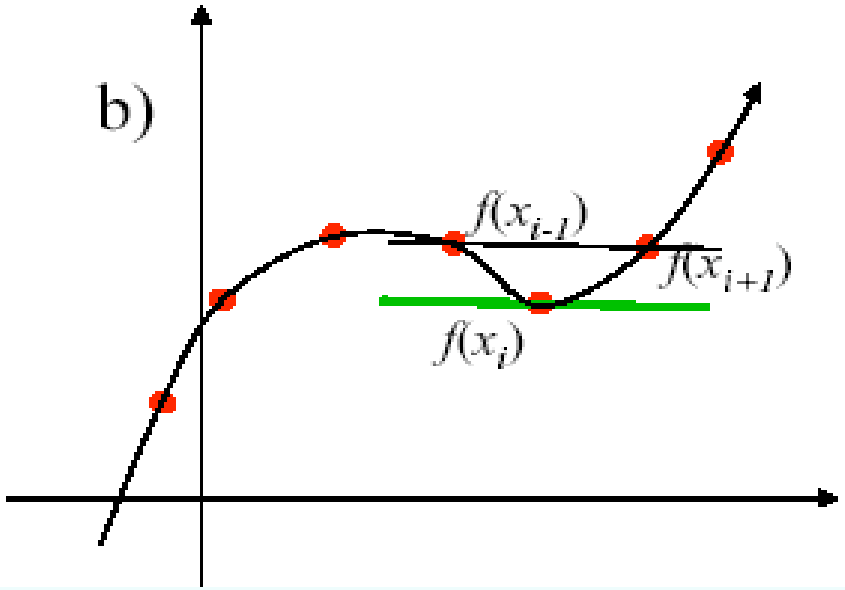
Differentiation

a) $f'(x_0) = \frac{f_1 - f_0}{h} + O(h) \quad \rightarrow \quad f''(x_0) = \frac{f_2 - 2f_1 + f_0}{h^2} + O(h)$

b) $f'(x_0) = \frac{f_1 - f_{-1}}{2h} + O(h^2) \quad \rightarrow \quad f''(x_0) = \frac{f_2 - 2f_0 + f_{-1}}{h^2} + O(h^2)$

c) $f'(x_0) = \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h} + O(h^4) \quad \rightarrow \quad \ldots$





Aufgaben

Schreiben Sie ein Programm, das die erste und zweite Ableitung einer beliebigen Funktion berechnet. Testen Sie Ihr Programm fuer f(x)=ln(x) und vergleichen Sie das Ergebnis mit der analytischen Lösung (graphische Darstellung mit gnuplot im Bereich]0,3]). Untersuchen Sie die Ergebnisse, wenn Sie die Schrittweite h variieren.

Der Name der source-code Datei soll "diff" lauten (diff.f oder diff.c).
Verwenden Sie beim Kompilieren den switch "-o filename" und benennen Sie auf diese Weise die executable-datei ebenfalls "diff":

```
f77 diff.f -o diff  -lm
```

Der name der output Dateien soll "diff.h1" und "diff.h2" sein.

Aufgaben

Diese Dateien sollen durch Leerzeichen getrennte Spalten enthalten. (Die Zahlen in eckigen Klammern verweisen auf die Formelnummern im Skriptum)

- Spalte 1: x
- Spalte 2: f(x)
- Spalte 3: f'(x) [analytisch]
- Spalte 4: f'(x) [5.45]
- Spalte 5: f'(x) [5.46]
- Spalte 6: f''(x) [analytisch]
- Spalte 7: f''(x) [5.49]
- Spalte 8: f''(x) [5.50]

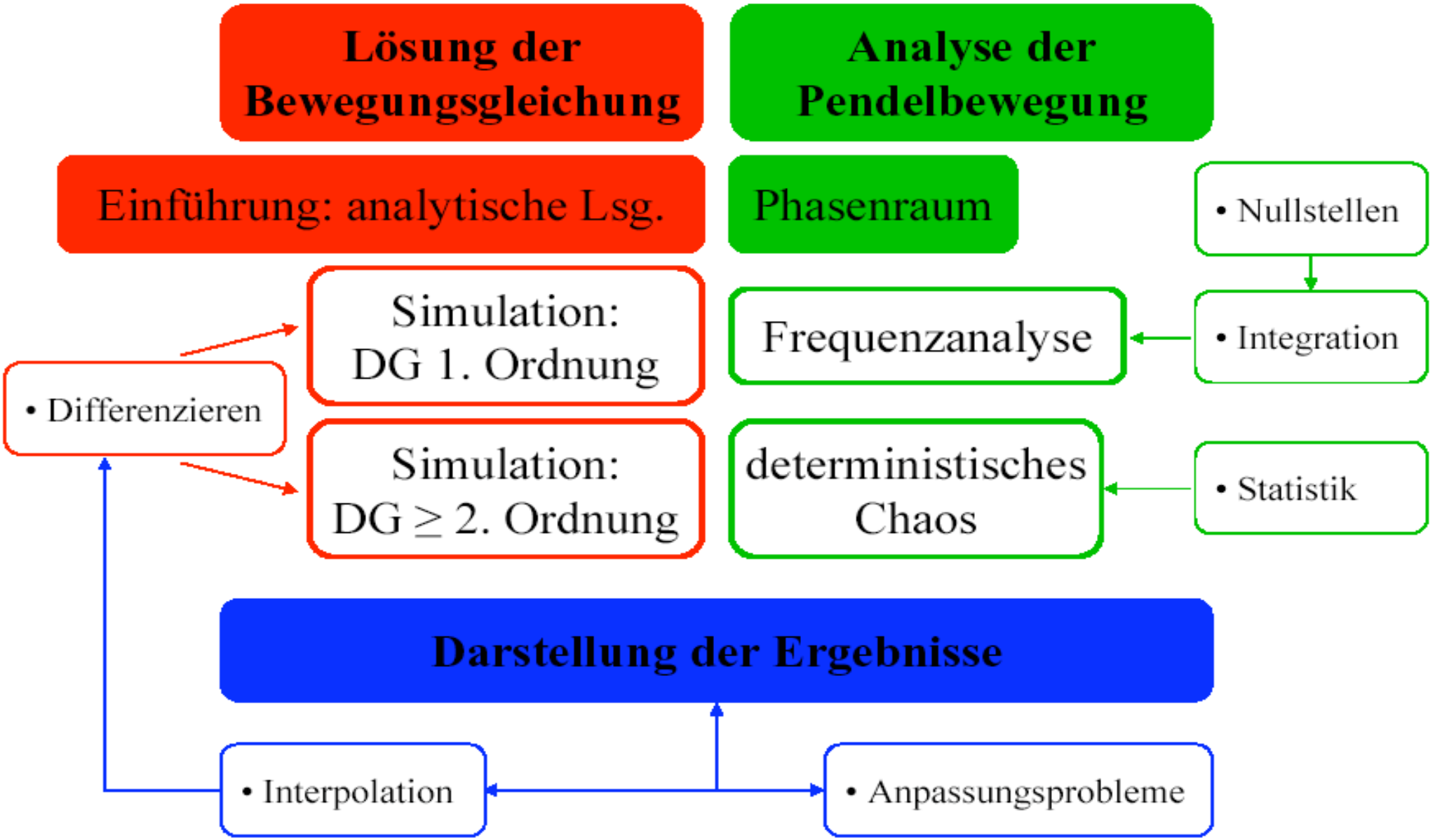
diff.h1 enthält die Ergebnisse für eine Schrittgrösse h=0.1 (im Intervall]0,3]) diff.h2 enthält die Ergebnisse für eine Schrittgrösse h=0.001 (im Intervall]0,3])

Hinweis: Unter dem Betriebssystem UNIX können Sie diese Dateien über sogenannte "pipes" erzeugen. Schreiben Sie ausserdem ein gnuplot load file "diff.g" welches f''(x) analytisch und nach 5.49 und 5.50 für h1=0.1 und h2=0.001 mit einander vergleicht. Hinweis: wenn Sie diese Kurven in gnuplot dargestellt haben, können Sie die gnuplot-load-datei einfach mit dem Befehl "save 'diff.g' " erzeugen. Probieren Sie es aus.

Aufgaben

Implementieren Sie das Newton'sche Interpolationsschema [5.11] in einem Program namens "newton".
Das executable sollte auch "newton" heissen (siehe Verwendung des "-o" switch oben). Die Dateien "newton.i1" und "newton.i2" enthalten die Datenpunkte durch die das Polynom gelegt werden soll. Lesen Sie diese ein und erzeugen Sie daraus die Dateien mit den im Intervall [-5,5] interpolierten Werten (mit einer Schrittgrösse von 0.1) die Sie "newton.o1" und "newton.o2" nennen. Erzeugen Sie 2 gnuplot load files "newton.g1" und "newton.g2" welche die Ergebnisse mit den Daten an den Stützstellen vergleichen.

Num. Lösung von Diff .Gleichungen



Num. Lösung von Diff .Gleichungen

- Beispiel: Pendel
- Transformation auf DG erster Ordnung
- Energieerhaltung:

$$H = E = \frac{1}{2} M L^2 \dot{\theta}^2 + M \omega_0^2 L^2 \left(1 - \cos\theta\right)$$

- Taylorreihenentwicklung:

$$\theta\left(t + \Delta t\right) = \theta(t) + \dot{\theta} \cdot \Delta t + \frac{\Delta t^2}{2} \ddot{\theta}(t) + \ldots$$

- Erste Ordnung:

$$\dot{\theta} = \pm \sqrt{\frac{2E}{ML^2} - 2\omega_0^2 \left(1 - \cos\theta\right)}$$

$$\rightarrow \quad \theta(t + \Delta t) \approx \theta(t) \pm \Delta t \cdot \omega_0 \sqrt{\frac{2E}{MgL} - 2\left(1 - \cos\theta\right)}$$

- + Randbedingung

NB: Diese Formel soll in Aufgabe 1 Verwendet werden!!!!!!

Num. Lösung von Diff .Gleichungen

- Allgeine Form für ein System von n D.G. erster Ordnung:

$$\begin{aligned} y_1' &= f_1(x, y_1, \ldots, y_n) \\ y_2' &= f_2(x, y_1, \ldots, y_n) \\ &\vdots \\ y_n' &= f_n(x, y_1, \ldots, y_n) \end{aligned}$$

für gegebene Funktionen $f_i, i = 1, \ldots, n.$

Num. Lösung von Diff .Gleichungen

•Transformation einer DG *n*-ter Ordnung (Randwertproblem)

$$y^{(n)} = f(x, y, y', y'', \ldots, y^{(n-1)})$$

in *n* DG-en erster Ordnung. Über die Definition:

$$z_1 = y\,, \quad z_2 = y'\,, \quad \ldots\,, \quad z_n = y^{(n-1)}$$

Erhält man:

$$\begin{aligned} z_1' &= z_2 \\ z_2' &= z_3 \\ &\vdots \quad\quad \vdots \\ z_n' &= f_n(x, z_1, \ldots, z_n) \end{aligned}$$

Num. Lösung von Diff .Gleichungen

- Autonomes System, d.h. die unabhängige Veränderliche kommt auf der rechten Seite des DG-Systems nicht mehr explizit vor:

Definiere: $y_0 = x$

Damit wird das System von DG-en:

$$\begin{aligned} y_0' &= 1 \\ y_1' &= f_1(y_0, y_1, \ldots, y_n) \\ y_2' &= f_2(y_0, y_1, \ldots, y_n) \\ &\vdots \quad\quad \vdots \\ y_n' &= f_n(y_0, y_1, \ldots, y_n) \end{aligned}$$

Welches man als autonomes System bezeichnet.

- Vorteile für numerische Implementierung

Num. Lösung von Diff .Gleichungen

- Beispiele

- Freies Pendel:

$\theta'' = -c\ \theta$

$z_0 = t$
$z_1 = \theta$
$z_2 = \theta'$

Autonomes System:

$z_2' = \theta'' = -c\, z_1$
$z_1' = \theta' = z_2$
$z_0' = 1$

- Bewegungsgl. 3-Dimensional:

$F = ma$

$z_0 = t$
$z_1 = x$
$z_2 = y$
$z_3 = z$
$z_4 = v_x$
$z_5 = v_y$
$z_6 = v_z$

- Autonomes System:

$z_0' = 1$
$z_1' = z_4$
$z_2' = z_5$
$z_3' = z_6$
$z_4' = F_x(z_0, z_1, z_1' \ldots)/m$
$z_5' = F_y(z_0, z_1, z_1' \ldots)/m$
$z_6' = F_z(z_0, z_1, z_1' \ldots)/m$

Num. Lösung von Diff .Gleichungen

- **Vektornotation:**

$$\begin{aligned} y_1' &= f_1(x, y_1, \ldots, y_n) \\ y_2' &= f_2(x, y_1, \ldots, y_n) \\ &\vdots \\ y_n' &= f_n(x, y_1, \ldots, y_n) \end{aligned}$$

->

$$\underline{y'} = \underline{f}(x, \underline{y})$$

$$\underline{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad \text{und} \quad \underline{f}(x, \underline{y}) = \begin{pmatrix} f_1(x, y_1, \ldots, y_n) \\ \vdots \\ f_n(x, y_1, \ldots, y_n) \end{pmatrix}$$

Einschrittverfahren

- Stützstellen: $x_i, i = 0, \ldots, N$ im Intervall $[x_a, x_e]$
- Schrittweite zwischen *i*-ter und *i+1*-ter Iteration: $h_{i+1} = x_{i+1} - x_i$
- Rekursionsvorschrift:

$$\underline{y}(x_{i+1}) = \underline{y}(x_i) + h_{i+1}\underline{\Phi}(x_i, \underline{y}(x_i); h_{i+1})$$

- Randbedingung: $\underline{y}(x_0) = \underline{y}(x_a)$
- Lösung an *i*-ter Stützstelle: $\underline{y}(x_i)$
- Die charakteristische Funktion

$$\Phi(x, \underline{y}(x), h)$$

muß entsprechend gewählt werden und bestimmt das numerische Verhalten des Verfahrens.

Einschrittverfahren 1. und 2. Ordnung

- Einzel DG: $y' = f(x, y)$
- Taylorreihe:

$$y(x+h) = \sum_{k=0}^{m} \frac{h^k}{k!}\, y^{(k)}(x) + \frac{h^{m+1}}{(m+1)!} y^{(m+1)}(x+\theta h)$$

- Ableitungen aus:

$$\begin{aligned} y' &= f(x,y)\,, \\ y'' &= \frac{d^2y}{dx^2} = \frac{df}{dx} + \frac{df}{dy}\frac{dy}{dx} = f_x + f_y f\,, \\ y^{(3)} &= f_{xx} + 2f f_{xy} + f_{yy} f^2 f_x f_y + f f_y^2\,, \\ \ldots &\quad \ldots \quad \ldots \end{aligned}$$

- 1. Ordnung: Euler- oder Polygonzugverfahren:

$$y(x+h) = y(x) + h f(x, y(x))$$

- Charakteristische Funktion: $\Phi(x, y; h) = f(x, y)$
- Das heisst: $y' \approx \dfrac{y(x+h) - y(x)}{h}$

Einschrittverfahren 2. Ordnung

- **Zur Erinnerung:**

$$y'' = \frac{d^2y}{dx^2} = \frac{df}{dx} + \frac{df}{dy}\frac{dy}{dx} = f_x + f_y f$$

- **=>Suche Charakteristische Funktion der Form:**

$$\Phi(x, y; h) = (a_1 + a_2) f(x, y) + a_2 h \left[p_1 f_x(x, y) + p_2 f_y(x, y) f(x, y)\right] + o(h^2)$$

- **Diese wird z.B. gegeben durch:**

$$\Phi(x, y; h) = a_1 f(x, y) + a_2 f(x + p_1 h, y + p_2 h f(x, y))$$

- **Verfahren von Heun:**

$$\begin{aligned} a_1 = a_2 &= \tfrac{1}{2}, \qquad p_1 = p_2 = 1 \\ \Phi(x, y; h) &= \frac{1}{2}\left[f(x, y) + f(x + h, y + h f(x, y))\right] \end{aligned}$$

- **Modifiziertes Euler Verfahren:**

$$\begin{aligned} a_1 = 0\ ; \qquad a_2 = 1\ ; \quad p_1 = p_2 &= \tfrac{1}{2} \\ \Phi(x, y; h) &= f(x + \tfrac{1}{2}h, y + \tfrac{1}{2}h f(x, y)) \end{aligned}$$

Das getriebene Pendel

- **Antriebskraft:** $F\sin(\Omega t+\beta)$

- **DG (2.Ordnung, nichtlinear):**

$$mL\frac{d^2\alpha}{dt^2}+\gamma L\frac{d\alpha}{dt}+mg\sin(\alpha)=F\sin(\Omega t+\beta)$$

- **reduzierte (dimensionslose) Veränderliche:** $\tau=t/t_0$ $t_0=\frac{1}{\omega_0}=\sqrt{\frac{L}{g}}$
- **=>**

$$\frac{d^2}{d\tau^2}+\Gamma\frac{d\alpha}{d\tau}+\sin(\alpha)=f\sin(\frac{\Omega}{\omega_0}\tau+\beta)$$

$$\Gamma=\frac{\gamma}{m}\sqrt{\frac{L}{g}} \qquad f=\frac{F}{mL}$$

Aufgaben

1. Schreiben Sie ein Programm zur numerischen Loesung der Schwingungsgleichung des freien Pendels. Verwenden Sie die in der Vorlesung angegebene Formel welche die zeitliche Auslenkung in Form einer D.G. erster Ordnung beschreibt. Die Quell- und exe-Datei sollen beide den Namen pendel1 haben. Vergleichen Sie die Ergebnisse fuer unterschiedliche Anfangsbedingungen mit der exakten Loesung der linearisierten DG und stellen Sie diesen Vergleich graphisch dar. Verwenden Sie zum Testen ein Pendel mit einer Laenge L=1 m und einer Masse M=1 kg. Erstellen Sie zwei Dateien, welche die Simulationsergebnisse und die exakte Lösung der linearisierten Gleichung fuer unterschiedliche Randbedingungen enthalten:
rb1.dat: die Pendelmasse wird aufeiner Hoehe von H=10 cm losgelassen
rb2.dat: die Pendelmasse wird auf einerHoehe von H=150 cm losgelassen
Die Dateien sollen wiefolgt formatiert sein:
1 Spalte: Zeit t (in sekunden)
2 Spalte: numerische Loesung fuer die Auslenkung
3 Spalte: Zeitableitung der numerischen Loesung
4 Spalte: exakte Loesung der linearisierten Gleichung.
5 Spalte: Zeitableitung der exakten Loesung derlinearisierten Gleichung.

Aufgaben

Das betrachtete Zeitintervall sollte einige Schwingungsperioden beinhalten. Ueberlegen Sie selbst einen sinnvollen Wert fuer das Zeitinterval Delta t. Erstellen Sie 2 gnuplot load-Dateien mit den Namen pendel_rb1.g und pendel_rb2.g welche Ihre Ergebnisse bei entsprechendem Aufruf darstellen. Stellen Sie die Trajektorien fuer die oa Anfangsbedingungen in einem Phasenraumdiagramm dar. Erstellen Sie 2 gnuplot load-Dateien mit den Namen pendel_phas1.g und pendel_phas2.g welche Ihre Ergebnisse bei entsprechendem Aufruf darstellen.

Aufgaben

2. Implementieren Sie das Euler-Verfahren zur Simulation des gedaempften Pendels in einem Programm namens euler. Verwenden Sie reduzierte Variablen und bringen Sie das Gleichungssystem in autonome Form. Rechnen Sie intern mit reduzierten Variablen welche bei der Ein- und Ausgabe entsprechend konvertiert werden. Vergleichen Sie das Ergebnis mit jenen in Angabe 2 wenn Sie für die reduzierte Daempfung den Wert 0 eingeben. Erstellen Sie die gnuplot load-Datei euler1.g welche die Darstellung dieses Vergleiches in gnuplot ermoeglicht. Fuehren Sie weiters die selben Simulationen wie in Angabe 1 beschrieben durch, fuer einen Wert der reduzierten Daempfungskonstanten Gamma=0.1. Das Namen der Dateien sollten nach dem selben Schema gewaehlt werden, wobei "pendel" durch "euler" ersetzt wird.