

# Einführung in die Numerik

Peter Bastian

Universität Heidelberg  
Interdisziplinäres Zentrum für Wissenschaftliches Rechnen  
Im Neuenheimer Feld 368, D-69120 Heidelberg  
email: [peter.bastian@iwr.uni-heidelberg.de](mailto:peter.bastian@iwr.uni-heidelberg.de)

17. April 2013

## Vorlesung

- Dozent: Peter Bastian, INF 368, R. 420, Sprechstunde Do 13-14 ([peter.bastian@iwr.uni-heidelberg.de](mailto:peter.bastian@iwr.uni-heidelberg.de))
- Übungsleitung: Rebecca Neumann / Pavel Hron ([{rebecca.neumann,pavel.hron}@iwr.uni-heidelberg.de](mailto:{rebecca.neumann,pavel.hron}@iwr.uni-heidelberg.de))
- Vorlesungstermine: Mittwoch 11-13, Freitag 9-11, HS 1, INF 288.
- Webseite zur Vorlesung (Materialien, Übungsaufgaben, ...)

[http://conan.iwr.uni-heidelberg.de/teaching/numerik0\\_ss2013/](http://conan.iwr.uni-heidelberg.de/teaching/numerik0_ss2013/)

- Vorlesung vorwiegend als Tafelanschrieb mit praktischen Beispielen auf Folien.
- Inhalt der Folien ist als PDF auf Webseite erhältlich.
- Skript auf Basis des Mitschriebs von Stefan Breunig wird im Laufe des Semesters erstellt und stückweise freigegeben.

- Vorlesung und praktische Programmierübungen verwenden Programmiersprache C++.
- Zur Vorlesung werden C++ Klassen für Matrizen, Vektoren und Zeitmessung zur Verfügung gestellt.
- Wir empfehlen (und unterstützen) Programmieren in einer LINUX Umgebung.
- Mac user: Analog Linux
- Windows user (**kein support!**):
  - LiveCD/Stick (<http://www.ubuntu.com>)
  - Wubi (<http://wubi-installer.org>)
  - Virtuelle Maschine mit Linux (<http://www.virtualbox.org>)
  - Cygwin (<http://www.cygwin.com>), Editor: Notepad++ (<http://notepad-plus-plus.org>)

Einsteigerkurs für Programmieranfänger (in zwei Teilen):

- Dienstag, 23.04.2011, 14-16 Uhr, Otto-Meyerhof-Zentrum, INF 350, Untergeschoss, Poolräume

Alternativtermin: Mo, 22.04., 9-11

**Dieser Kurs ist verpflichtend für Hörer, die einen Programmierschein erwerben wollen und allen die nicht sehr gut C++ Programmieren können dringend empfohlen**

Begleitend zur Vorlesung finden Übungen statt:

- Pro Woche wird ein Übungsblatt mit theoretischen und praktischen Übungen (Programmierübungen) ausgegeben.
- Punkteverhältnis ca. 70% (theoretisch) zu 30% (praktisch).
- Eine erfolgreiche Teilnahme an den Übungen ist Voraussetzung für die Teilnahme an der Klausur.
- Einmal pro Woche findet **betreutes Programmieren** im CIP-Pool statt. Die Teilnehmer können hier Unterstützung bei Schwierigkeiten mit der Lösung der Programmieraufgaben erhalten.
- Die regelmäßige Teilnahme an den praktischen Übungen ist für den Erwerb eines "Programmierscheins" notwendig (siehe unten).

Eine Abgabe in Gruppen von zwei bis drei Teilnehmer ist möglich und ausdrücklich erwünscht.

- Übungsblätter immer Freitags (erstes Übungsblatt Freitag 19.04.2011).
- Bearbeitungszeit jeweils 1 Woche.
- Abgabe im Foyer INF 288 (weißer Schrank, links), Freitags vor der Vorlesung.
- Beginn der Übungsgruppen ab 29.04.2011.

## Termine der Übungsgruppen:

Ab Mi 17.4., ca. 16 Uhr werden zunächst 5 Übungsgruppen geöffnet.

## Übungstermine Praktische Übungen:

Di 14-16 Uhr INF 350 (OMZ), U011 + U012

*Alternativ:* Mo 9-11 Uhr INF 350 (OMZ), U011 + U012  
Fr 11-13 Uhr INF 350 (OMZ), U011 + U012

## Anmeldung über *Muesli*:

<https://www.mathi.uni-heidelberg.de/muesli/>

## Klausur:

Die Scheinvergabe und Benotung erfolgt aufgrund der Ergebnisse der verpflichtenden Klausur.

**Klausurtermin: Freitag, 26.7.2013, 9-11 Uhr**

## Zulassung zur Klausur:

- 50 % der Punkte aus allen Übungen
- Vorführung mindestens einer Lösung in der Übungsgruppe (einmal pro **Teilnehmer**, nicht einmal pro Abgabegruppe).

Lehramtsstudenten können in dieser Vorlesung auch einen speziellen Schein über den Besuch mathematischer Programmierübungen (“Programmierschein”) erwerben. Voraussetzung hierfür sind:

- Besuch des C++-Kurzurses
- die Vorführung wenigstens jeder zweiten Programmieraufgabe in der Poolstunde.
- Betreutes Programmieren wird angeboten, ist aber nicht verpflichtend.

- Numerik (gewöhnlicher Differentialgleichungen)
- Numerik partieller Differentialgleichungen
- Parallele Löser für große Gleichungssysteme
- Kontinuumsmechanik
- Strömungsmechanik
- Algorithmische Optimierung I/II
- Objektorientiertes Programmieren im Wissenschaftlichen Rechnen

Es gibt noch freie Plätze im Softwarepraktikum Wissenschaftliches Rechnen (Anfänger und Fortgeschrittene)

## 1 Motivation

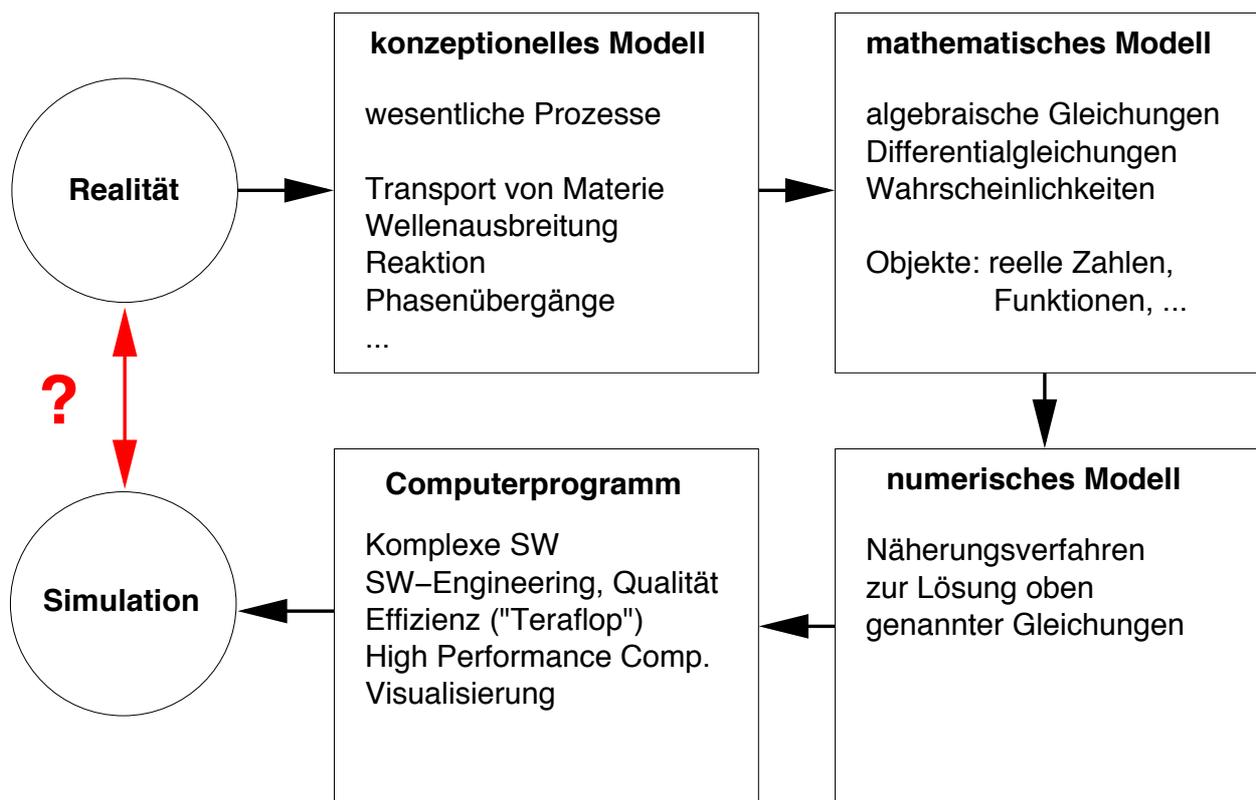
- Modellbildung und Simulation
- Ein einfaches Beispiel: Das Fadenpendel
- Inhaltsübersicht der Vorlesung

# Die Wissenschaftliche Methode

- Experiment: Beobachte (und messe) was passiert.
- Theorie: Versuche die Beobachtung mit Hilfe von Modellen zu erklären.
- Theorie und Experiment werden sukzessive verfeinert und verglichen, bis eine akzeptable Übereinstimmung vorliegt.
- In Naturwissenschaft und Technik liegen Modelle oft in Form mathematischer Gleichungen vor. Z. B. gewöhnliche oder partielle Differentialgleichungen.
- Oft können die Modellgleichungen nicht geschlossen (mit Papier und Bleistift oder Mathematica . . . ) gelöst werden.

⇒ **Numerische Simulation und Optimierung**

- Simulation: Gleichungen numerisch lösen.
  - Undurchführbare Experimente werden möglich (z. B. Galaxienkollisionen).
  - Teure Experimente werden eingespart (z. B. Modelle im Windkanal).
  - Parameterstudien schneller durchführbar.
  - (Automatische) Optimierung von Prozessen.
  - Identifikation von Modellparametern aus Messungen.
- Vielfältiger Einsatz in Naturwissenschaft, Technik und Industrie: Strömungsberechnung (Wetter, Klima), Festigkeit von Bauwerken ...
- Grundlage für alle diese Anwendungen sind numerische Algorithmen!
- Auch Informatiker sind beteiligt: SW-Engineering, High-Performance (Parallel) Computing, ...



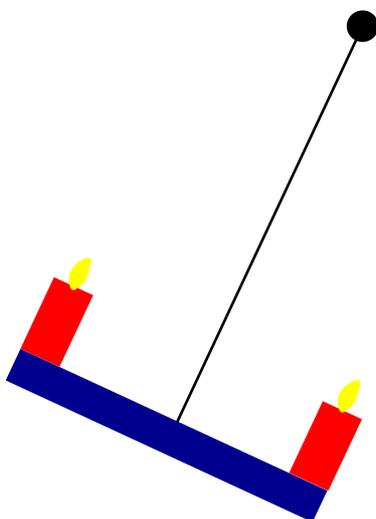
Unterschiede zwischen Experiment und Simulation haben verschiedene Gründe:

- *Modellfehler*: Ein relevanter Prozess wurde nicht oder ungenau modelliert (Temp. konstant, Luftwiderstand vernachlässigt, ...)
- *Datenfehler*: Messungen von Anfangsbedingungen, Randbedingungen, Werte für Parameter sind fehlerbehaftet.
- *Abschneidefehler*: Abbruch von Reihen oder Iterationsverfahren, Approximation von Funktionen (z.B. stückweise Polynome).
- *Rundungsfehler*: Reelle Zahlen werden im Rechner genähert dargestellt.

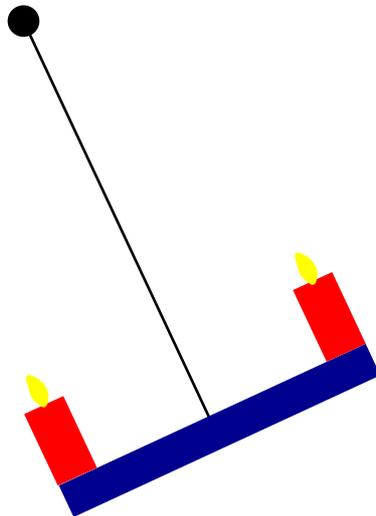
Untersuchung von Rundungsfehlern und Abschneidefehlern ist ein zentraler Aspekt der Vorlesung!

## Pisa, 1582

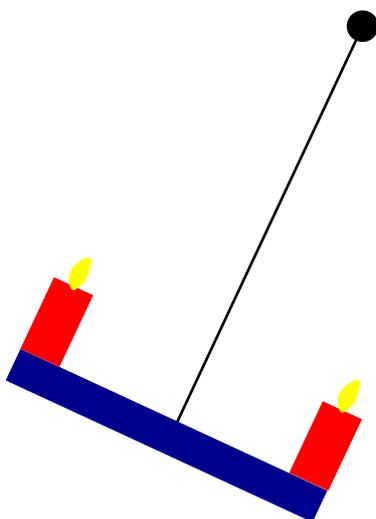
Der Student Galileo Galilei sitzt in der Kirche und ihm ist langweilig. Er beobachtet den langsam über ihm pendelnden Kerzenleuchter und denkt: „Wie kann ich nur die Bewegung dieses Leuchters beschreiben?“.



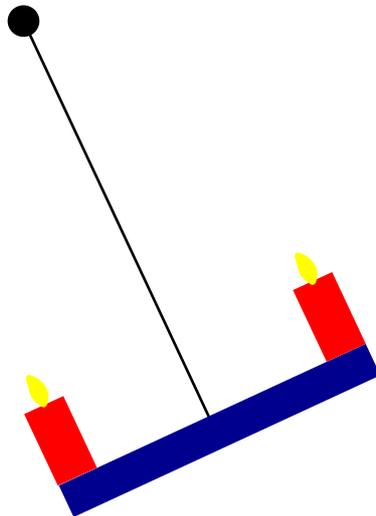
Der Student Galileo Galilei sitzt in der Kirche und ihm ist langweilig. Er beobachtet den langsam über ihm pendelnden Kerzenleuchter und denkt: „Wie kann ich nur die Bewegung dieses Leuchters beschreiben?“.



Der Student Galileo Galilei sitzt in der Kirche und ihm ist langweilig. Er beobachtet den langsam über ihm pendelnden Kerzenleuchter und denkt: „Wie kann ich nur die Bewegung dieses Leuchters beschreiben?“.



Der Student Galileo Galilei sitzt in der Kirche und ihm ist langweilig. Er beobachtet den langsam über ihm pendelnden Kerzenleuchter und denkt: „Wie kann ich nur die Bewegung dieses Leuchters beschreiben?“.

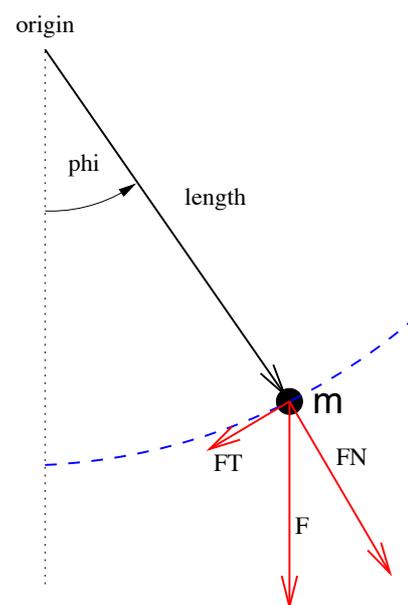


## Konzeptionelles Modell

Welche Eigenschaften (physikalischen Prozesse) sind für die gestellte Frage relevant?

- Leuchter ist ein Massenpunkt mit der Masse  $m$ .
- Der Faden der Länge  $\ell$  wird als rigide und masselos angenommen.
- Der Luftwiderstand wird vernachlässigt.

Nun entwickle mathematisches Modell.



- *Annahme*: Pendel läuft auf Kreisbahn: Nur *Tangentialkraft* ist relevant.
- Tangentialkraft bei Auslenkung  $\phi$ :

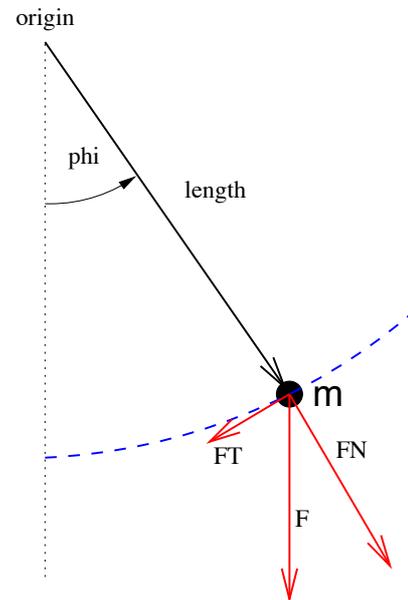
$$\vec{F}_T(\phi) = -mg \sin(\phi) \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \end{pmatrix}.$$

- Also etwa:

$$\vec{F}_T(0) = -mg \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

$$\vec{F}_T(\pi/2) = -mg \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

- Vorzeichen kodiert Richtung.



## Weg, Geschwindigkeit, Beschleunigung

- Weg  $s(t)$ , Geschwindigkeit  $v(t)$ , Beschleunigung  $a(t)$  erfüllen:

$$v(t) = \frac{ds(t)}{dt}, \quad a(t) = \frac{dv(t)}{dt}.$$

- Für den zurückgelegten Weg (mit Vorzeichen!) gilt  $s(t) = l\phi(t)$ .
- Also für die Geschwindigkeit

$$v(t) = \frac{ds(\phi(t))}{dt} = \frac{dl\phi(t)}{dt} = l \frac{d\phi(t)}{dt}$$

- und die Beschleunigung

$$a(t) = \frac{dv(\phi(t))}{dt} = l \frac{d^2\phi(t)}{dt^2}.$$

- Einsetzen in das 2. Newton'sche Gesetz  $m a(t) = F(t)$  liefert nun:

$$m\ell \frac{d^2\phi(t)}{dt^2} = -mg \sin(\phi(t)) \quad \forall t > t_0.$$

- Die Kraft ist hier skalar (vorzeichenbehafteter Betrag der Tangentialkraft), da wir nur den zurückgelegten Weg betrachten.
- Ergibt *gewöhnliche Differentialgleichung 2. Ordnung* für die Auslenkung  $\phi(t)$ :

$$\frac{d^2\phi(t)}{dt^2} = -\frac{g}{\ell} \sin(\phi(t)) \quad \forall t > t_0. \quad (1)$$

- Eindeutige Lösung erfordert *zwei* Anfangsbedingungen ( $t_0 = 0$ ):

$$\phi(0) = \phi_0, \quad \frac{d\phi}{dt}(0) = u_0. \quad (2)$$

## Lösung bei kleiner Auslenkung

- Allgemeine Gleichung für das Pendel ist schwer „analytisch“ zu lösen.
- Für *kleine* Winkel  $\phi$  gilt

$$\sin(\phi) \approx \phi,$$

z.B.  $\sin(0,1) = 0,099833417$ .

- Diese *Näherung* reduziert die Gleichung auf

$$\frac{d^2\phi(t)}{dt^2} = -\frac{g}{\ell} \phi(t).$$

- Ansatz  $\phi(t) = A \cos(\omega t)$  liefert mit  $\phi(0) = \phi_0$ ,  $\frac{d\phi}{dt}(0) = 0$  dann die aus der Schule bekannte Formel

$$\phi(t) = \phi_0 \cos\left(\sqrt{\frac{g}{\ell}} t\right) \quad (3)$$

- Löse das volle Modell mit zwei numerischen Verfahren.
- Ersetze Gleichung zweiter Ordnung durch zwei Gleichungen erster Ordnung:

$$\frac{d\phi(t)}{dt} = u(t), \quad \frac{d^2\phi(t)}{dt^2} = \frac{du(t)}{dt} = -\frac{g}{\ell} \sin(\phi(t)).$$

- Ersetze Ableitungen durch Differenzenquotienten:

$$\frac{\phi(t + \Delta t) - \phi(t)}{\Delta t} \approx \frac{d\phi(t)}{dt} = u(t),$$

$$\frac{u(t + \Delta t) - u(t)}{\Delta t} \approx \frac{du(t)}{dt} = -\frac{g}{\ell} \sin(\phi(t)).$$

- Mit  $\phi^n = \phi(n\Delta t)$ ,  $u^n = u(n\Delta t)$  erhält man Rekursion (*Euler*):

$$\phi^{n+1} = \phi^n + \Delta t u^n \qquad \phi^0 = \phi_0 \qquad (4)$$

$$u^{n+1} = u^n - \Delta t (g/\ell) \sin(\phi^n) \qquad u^0 = u_0 \qquad (5)$$

- Nutze Näherungsformel für die zweite Ableitung, sog. *Zentraler Differenzenquotient*:

$$\frac{\phi(t + \Delta t) - 2\phi(t) + \phi(t - \Delta t)}{\Delta t^2} \approx \frac{d^2\phi(t)}{dt^2} = -\frac{g}{\ell} \sin(\phi(t)).$$

- Auflösen nach  $\phi(t + \Delta t)$  ergibt Rekursionsformel ( $n \geq 2$ ):

$$\phi^{n+1} = 2\phi^n - \phi^{n-1} - \Delta t^2 (g/\ell) \sin(\phi^n) \qquad (6)$$

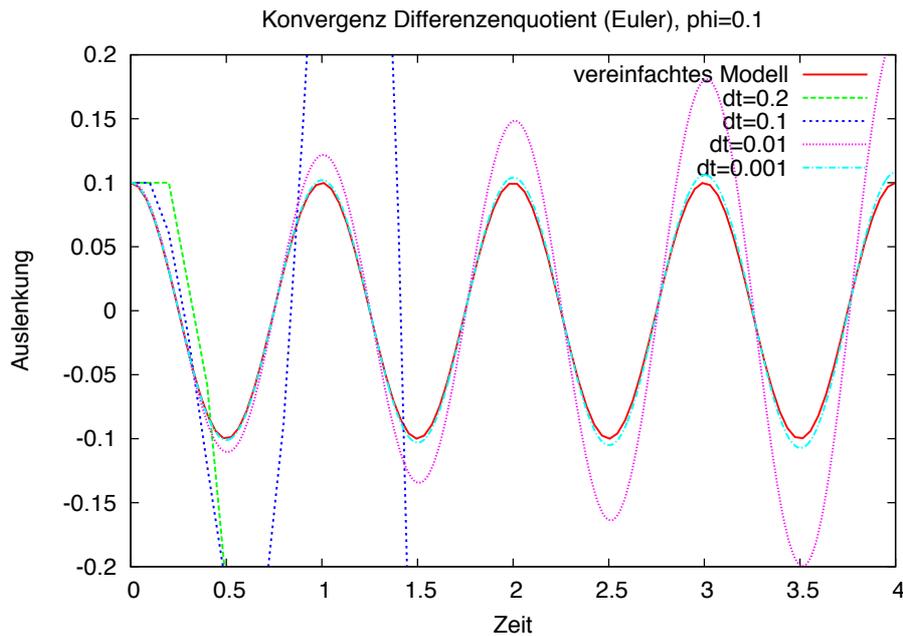
mit der Anfangsbedingung

$$\phi^0 = \phi_0, \quad \phi^1 = \phi_0 + \Delta t u_0. \qquad (7)$$

(Die zweite Bedingung kommt aus dem Eulerverfahren oben).

# Experiment 1: $\phi_0 = 0.1 \approx 5.7^\circ$

Differenzenquotient (Eulerverfahren)

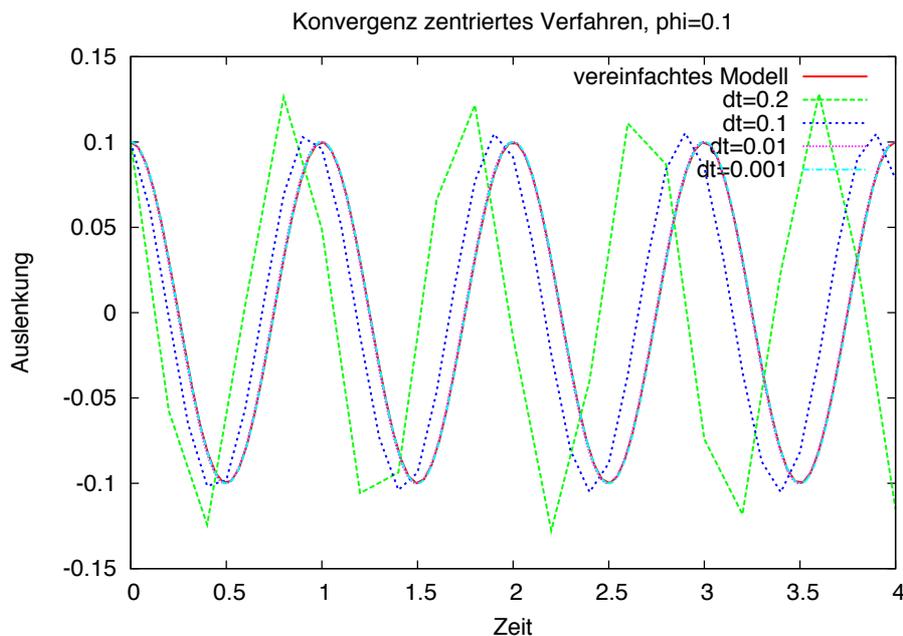


Für festen Zeitpunkt  $t$  und  $\Delta t \rightarrow 0$  konvergiert das Verfahren.

Für festes  $\Delta t$  und  $t \rightarrow \infty$  wächst der Fehler an.

# Experiment 2: $\phi_0 = 0.1 \approx 5.7^\circ$

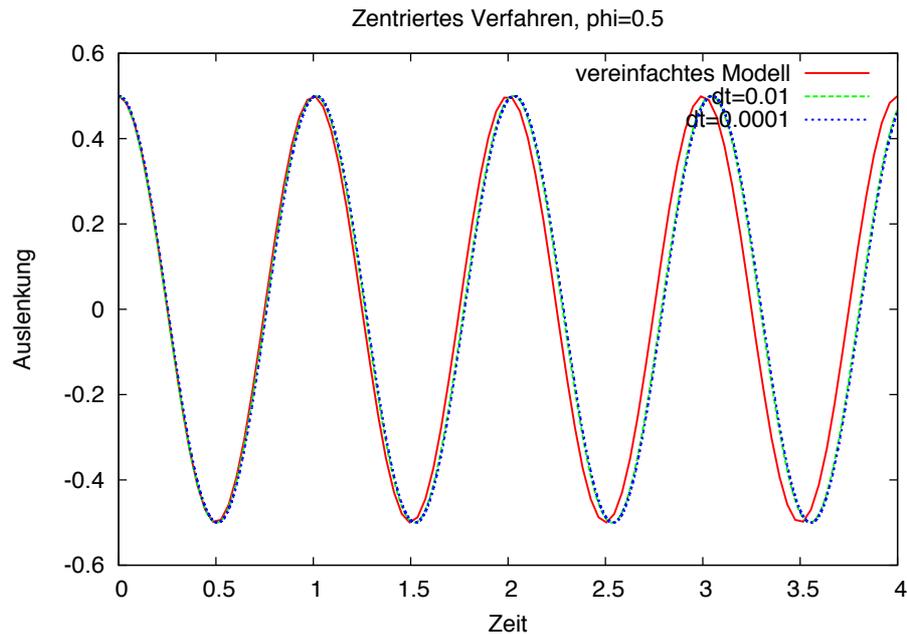
Zentrales Verfahren



Im Unterschied zum expliziten Euler scheint das Verfahren bei festem  $\Delta t$  und  $t \rightarrow \infty$  nicht unbeschränkt zu wachsen.

# Experiment 3: $\phi_0 = 0.5 \approx 28.6^\circ$

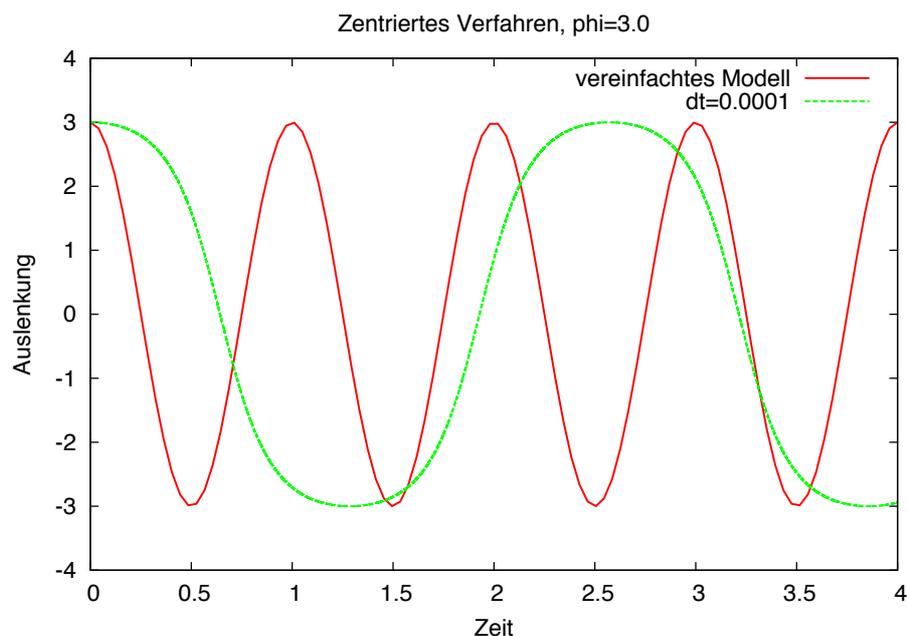
## Zentrales Verfahren



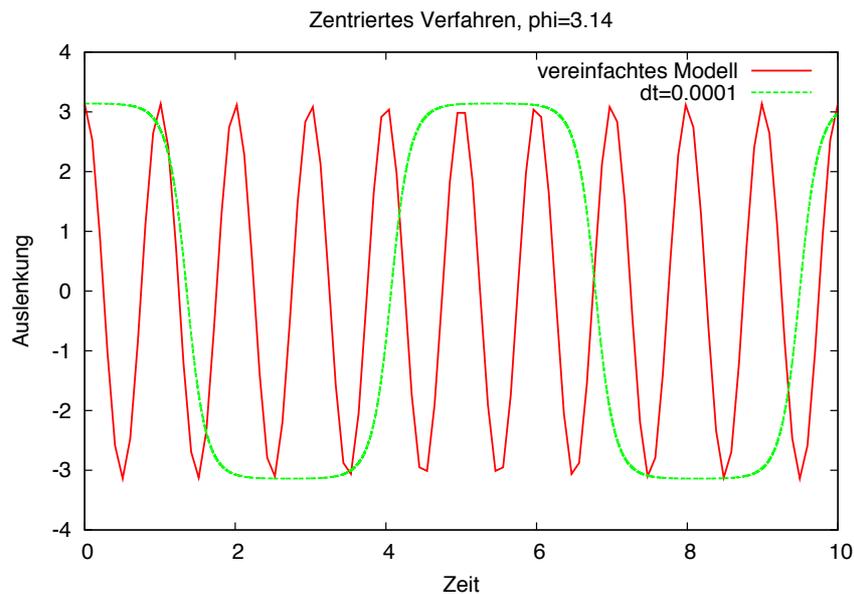
Selbst bei  $28.6^\circ$  ist die Übereinstimmung noch einigermaßen passabel.

# Experiment 4: $\phi_0 = 3.0 \approx 171^\circ$

## Zentrales Verfahren

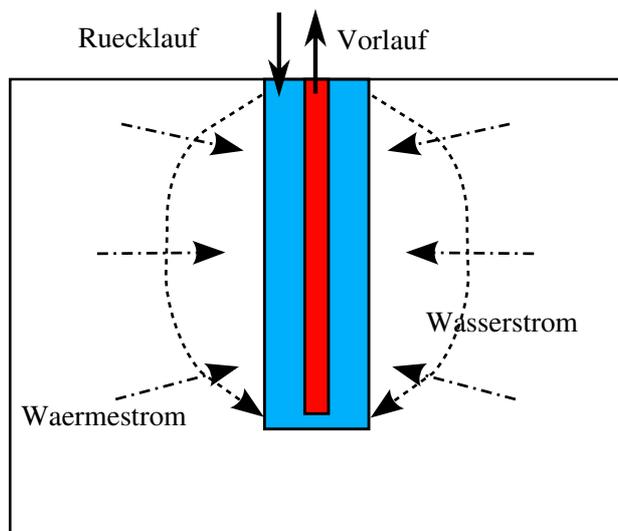


Für große Auslenkungen ist das vereinfachte Modell völlig unbrauchbar. Die Form der Schwingung ist kein Kosinus mehr.



Das Pendel wird nahe  $\pi$  immer langsamer. Das ist die Schiffschaukel, die fast auf dem Kopf steht. Wie würde denn die Kurve bei einer umlaufenden Schiffschaukel aussehen?

## Eine Geothermieanlage



- Grundwasserströmung gekoppelt mit Wärmetransport.
- Welche Leistung erzielt so eine Anlage?

- *Strömung des Wassers* in und um das Bohrloch

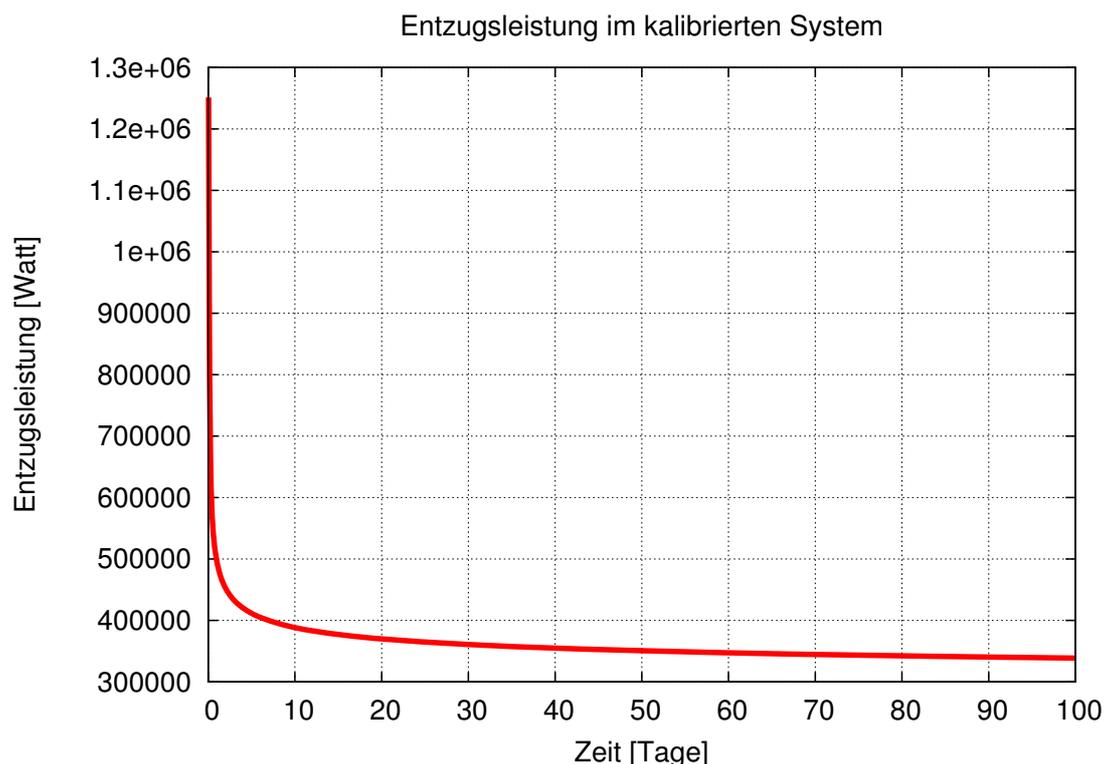
$$\nabla \cdot u = f,$$
$$u = -\frac{K}{\mu}(\nabla p - \rho_w G)$$

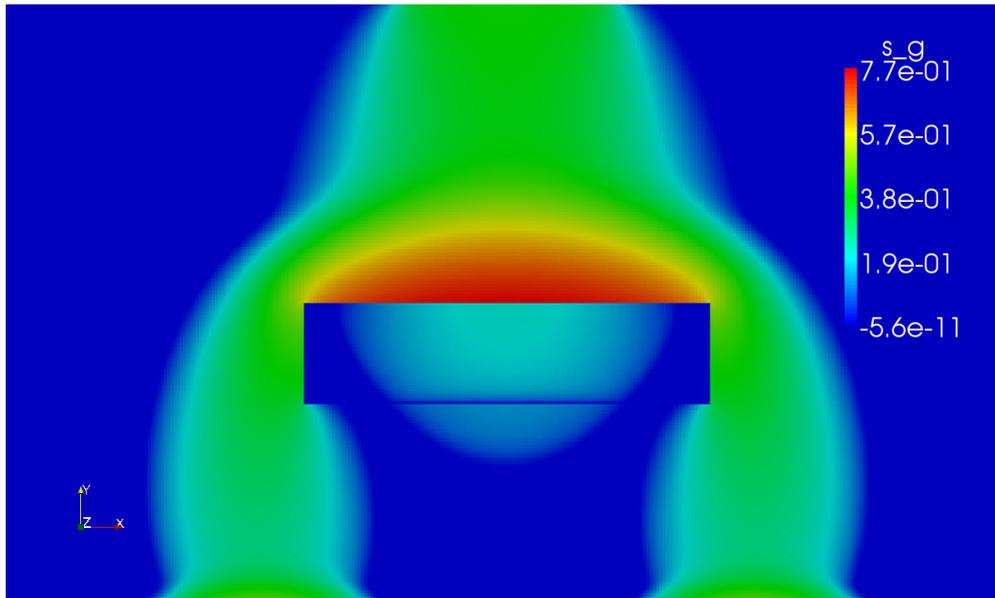
- Transport der Wärme durch *Konvektion* und *Wärmeleitung*

$$\frac{\partial(c_e \rho_e T)}{\partial t} + \nabla \cdot q + c_w \rho_w f T = g,$$
$$q = c_w \rho_w u T - \lambda \nabla T$$

in Abhängigkeit diverser *Parameter*: Bodendurchlässigkeit, Wärmekapazität, Dichte, Wärmeleitfähigkeit, Pumprate sowie Rand- und Anfangsbedingungen.

## Entzugsleistung





- Wo erreicht der Schadstoff welche Konzentrationen?
- Wie bekommt man den Schadstoff wieder weg?
- Wohin bewegt sich gelöster Schadstoff?

## Inhalt der Vorlesung

Wir werden in dieser Vorlesung die folgenden Themengebiete behandeln:

- Grundbegriffe, Gleitpunktzahlen, Gleitpunktarithmetik
- Direkte Methoden zur Lösung linearer Gleichungssysteme
- Interpolation und Approximation
- Numerische Integration
- Iterationsverfahren zur Lösung linearer Gleichungssysteme
- Iterationsverfahren zur Lösung nichtlinearer Gleichungssysteme
- Eigenwerte und Eigenvektoren

-  Deuffhard, P. und A. Hohmann: *Numerische Mathematik I, Eine algorithmisch orientierte Einführung*.  
de Gruyter, 2002.
-  Golub, G. H. und C. F. Van Loan: *Matrix Computations*.  
Johns Hopkins University Press, 2nd Auflage, 1989.
-  Quarteroni, A., R. Sacco und F. Saleri: *Numerical Mathematics*.  
Springer, 2nd Auflage, 2007.
-  Rannacher, R.: *Einführung in die Numerische Mathematik (Numerik 0)*.  
<http://numerik.iwr.uni-heidelberg.de/~lehre/notes>, 2006.

-  Stoer, J. und R. Bulirsch: *Numerische Mathematik II*.  
Springer, 5. Auflage, 2005.
-  Schwarz, H.-R. und N. Köckler: *Numerische Mathematik*.  
Teubner, 5. Auflage, 2005.
-  Stoer, J.: *Numerische Mathematik I*.  
Springer, 9. Auflage, 2005.
-  Schaback, R. und H. Wendland: *Numerische Mathematik*.  
Springer, 5th Auflage, 2005.

-  Eckel, B. und C. Allison: *Thinking in C++: Volume I: Introduction to Standard C++*.  
Prentice Hall Ptr, 2000.  
<http://mindview.net/Books/TICPP/ThinkingInCPP2e.html>.
-  Erlenkötter, H.: *C++ Objektorientiertes Programmieren von Anfang an*.  
Rowohlt Taschenbuchverlag GmbH, Reinbek bei Hamburg, 2000.
-  Stroustrup, B.: *Einführung in die Programmierung mit C++*.  
Pearson Studium, 2010.

-  Goldberg, D.: *What Every Computer Scientist Should Know About Floating Point Arithmetic*.  
Computing Surveys, 1991.  
<http://citeseer.ist.psu.edu/viewdoc/download;jsessionid=86013D0FEFFA6CD1A626176C5D4EF9E2?doi=10.1.1.102.244&rep=rep1&type=pdf>.
-  Knuth, D. E.: *The Art of Computer Programming, Band 2*.  
Addison-Wesley, 3. Auflage, 1998.