Genauigkeit numerischer Lösungen gewöhnlicher DGLn

Lernziele

Kenntnisse

- grundlegende Fehlerphänomene numerischer Lösungsverfahren von DGLn
- elementare Fehlerschätzungen
- elementare Einschätzung der Stabilität

Fertigkeiten

- ditto anwenden können: Praktikum!

Ausgangspunkt:
Anfangswertproblem
für explizite gewöhnliche DGL 1. Ordnung

$d\mathbf{y}/dt = f(t, \mathbf{y})$

$\mathbf{y}(0) = \mathbf{y}_0$

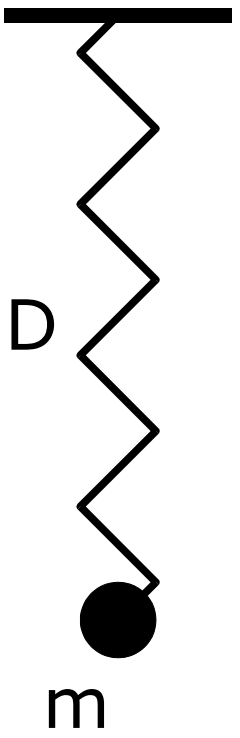
Numerische Behandlung:

- endliche Schrittweite für dt
- endliche Zahlenauflösung/Rechengenauigkeit

>> Codebeispiel 1

Softwaretechnik: Lösungsmethode abstrakt formuliert

Qualitativer Vergleich der klassischen numerischen Verfahren



Anwendungsbeispiel: harmonischer Oszillator

$dy/dt = p/m$

$dp/dt = -Dy$

$y(0) = y_0$

$p(0) = p_0$

(Welche Abhängigkeit y(t) ist zu erwarten?)

>> Codebeispiel 2

Euler, Mittelpunkt, Heun, Runge-Kutta (4. Ord.)
für harmonischen Oszillator: Fehler?

Beobachtung: bei kleinerer Schrittweite besser

Quantitativer Vergleich der klassischen numerischen Verfahren

Wie hängt die Genauigkeit von der Schrittweite ab?

Studienobjekt: exponentielles Wachstum

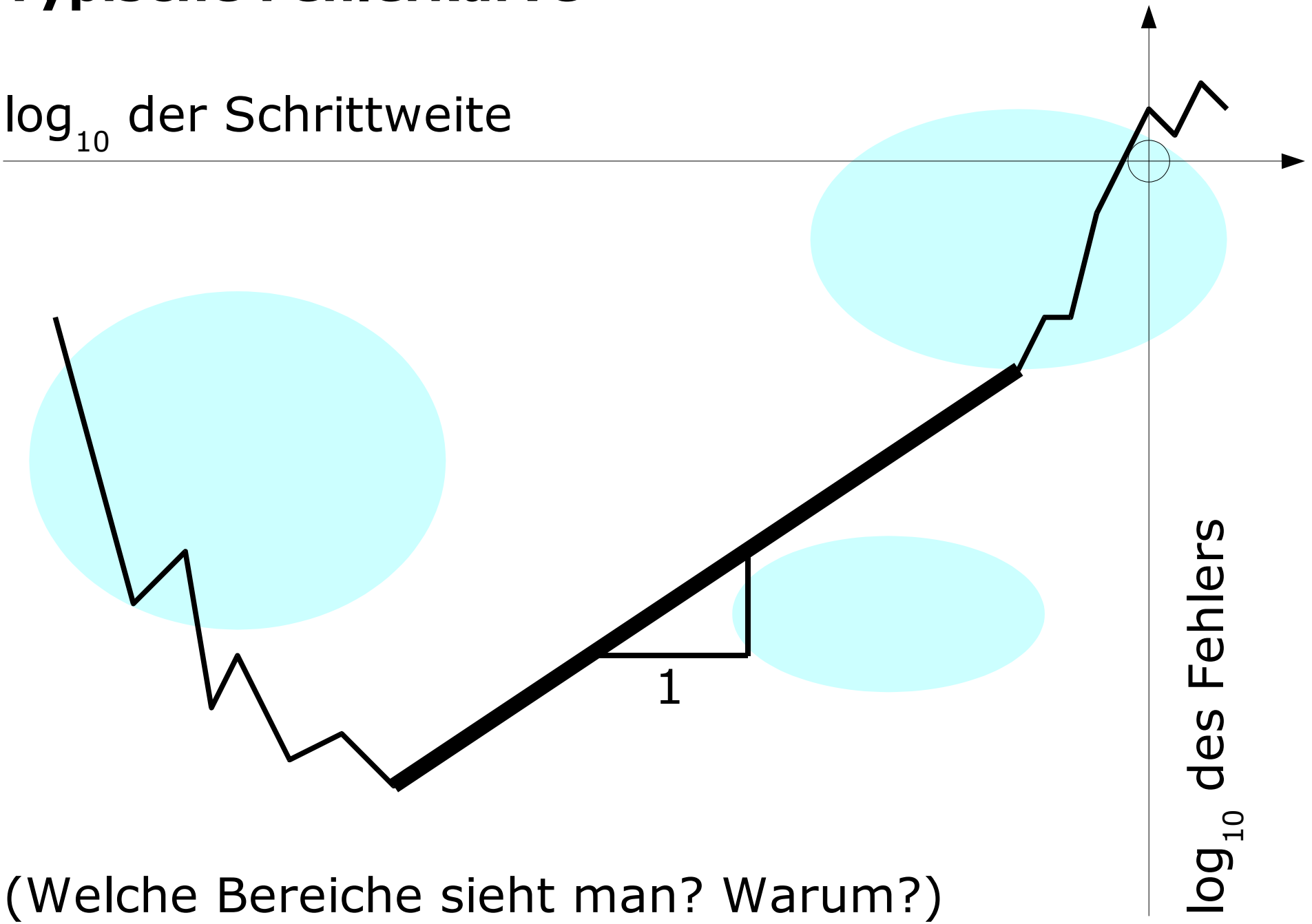
$dy/dt = y$
$y(0) = 1$

(Wie groß soll y(1) sein?)

>> Codebeispiel 3

Euler, Mittelpunkt, Heun, Runge-Kutta (4. Ord.);
doppelt-logarithmischer Plot:
Fehler von y(1) als Funktion der Schrittweite

Typische Fehlerkurve



(Welche Bereiche sieht man? Warum?)

Beobachtung im mittleren Bereich:
Steigung = Ordnung des Verfahrens

Rechnen mit einem Zehntel
der ursprünglichen Schrittweite
heißt im mittleren Bereich des Diagramms:

Rechenzeit: überall zehnfach

Genauigkeit:

- Euler:
- Mittelpunkt:
- Heun:
- Runge-Kutta (4. Ord.):

(Wie groß ist die relative Rechenzeit der Verfahren?)

(Und ist das von Bedeutung?)

Bestimmung der Ordnung eines Verfahrens

Umständliche Rechnung; hier nur Mittelpunkts-verfahren für „autonomes“ eindim. Problem:

$y'(t) = f(y(t))$

Exakte Lösung (Taylor!):

$$y(t+h) = y(t) + y'(t)\, h + \tfrac{1}{2}\, y''(t)\, h^2 + O(h^3)$$
$$= y(t) + f(y(t))\, h + \tfrac{1}{2}\, f'(y(t))\, f(y(t))\, h^2 + O(h^3)$$

Numerisch:

$$y_1 = y_0 + h\, f(y_0 + \tfrac{1}{2}\, h\, f(y_0))$$
$$= y_0 + h\, \{f(y_0) + f'(y_0)\, \tfrac{1}{2}\, h\, f(y_0) + O(h^2)\}$$

Folglich Fehler pro Schritt: $O(h^3)$,
naiv: Gesamtfehler $O(h^3) \cdot N = O(h^2)$.

Beobachtung im linken Bereich:
Schrittweite zu klein → nicht nur Zeitvergeudung, sondern sogar Genauigkeitsverlust

Naives Modell:
In jedem Schritt erhöht sich der Rundungsfehler um $\approx 10^{-15}$ (double-Zahlenformat, Zahlen ≈ 1).

Integration t = 0 bis 1, Schrittweite h, Schrittanzahl N, also Rundungs-Gesamtfehler naiv $\approx 10^{-15}$ N = 10^{-15}/h.

- Euler: Rundungsfehler und Diskretisierungsfehler gleich groß für h $\approx 10^{-7}$
- Runge-Kutta (4. Ord.): Gleichstand bereits für h $\approx 10^{-3}$ und auf viel genauerem Niveau:
  Rundungsfehler $\approx 10^{-15}/10^{-3}$
  Diskretisierungsfehler $\approx (10^{-3})^4$

Beobachtung im rechten Bereich:
Schrittweite nicht klein genug → Instabilität

>> Codebeispiel 4

$y'(t) = t - y(t)^2$ mit $y(0) = 0$.

Lösung $y(t)$ ist $\approx \sqrt{t}$.

Alle Standardverfahren werden
hier nach einiger Zeit instabil;
mit kleinerer Schrittweite
dauert das nur länger.

>> Codebeispiel 5

„Implizite" Verfahren wie $y_{k+1} = y_k + h\, f(y_{k+1})$
verhalten sich meist stabiler.

Stabilitätsgebiete

Typischer Stabilitätstest für DGL-Löser:
$y'(t) = \lambda\, y(t)$ mit $y(0) = 1$
und fester komplexer Zahl λ, $\mathrm{Re}(\lambda) < 0$ (Abfall!).

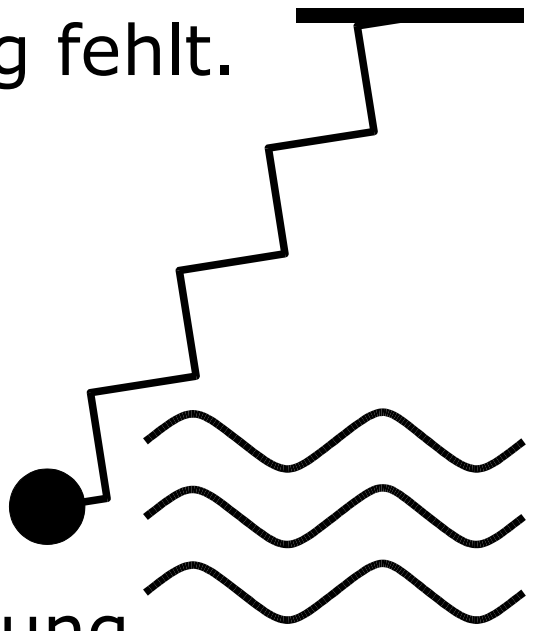
Euler:
$y_{k+1} = y_k + h\,\lambda\, y_k = (1 + h\,\lambda) y_k$
$y_N = (1 + h\,\lambda)^N\, y_0$
Also nur dann Abfall, wenn $|1 + h\,\lambda| < 1$.
„Stabilitätsgebiet": Menge solcher $h\,\lambda$

>> Codebeispiel 6
Stabilitätsgebiete für klassische Verfahren; so Stabilität auch für andere Aufgaben schätzbar (Praktikum?)

„Steife“ DGL-Systeme

… sind solche mit mehren stark verschiedenen Abklingkonstanten λ.

Typisch: „Schlimmste“ Konstante λ
beschränkt die Schrittweite, auch wenn
deren Komponente in der exakten Lösung fehlt.



>> Codebeispiel 7

frei drehbares, sehr steifes Federpendel
in Dämpfung

- schnelle Oszillation in Federrichtung
- langsames Zurückschwenken bei Drehung

Auch beim Start (fast) ohne Ausdehnung
gewinnt die schnelle Komponente.

Zusammenfassung

Verlauf des Fehlers
in Abhängigkeit von der Schrittweite h:

- h zu klein: Rundungsfehler nimmt überhand
- mittlerer Bereich: Fehler ~ $h^{Ordnung}$
- h zu groß: Instabilität

„h zu groß“ ist relativ:

- Stabilitätsbereich beachten
- Vorsicht mit steifen Systemen
- ggf. implizite Verfahren benutzen

Weiterführende Literatur

Roos/Schwetlink: Numerische Mathematik
ISBN 3-519-00221-3

Acheson: Vom Calculus zum Chaos
ISBN 3-486-24833-2

Press u.a.: Numerical Recipes
www.nr.com

Mein Beispielprogramm zum Download

www.*l*7h.cn/current/DGLn