

5. Gewöhnliche Differentialgleichungen

Für viele technische Anwendungen unverzichtbar!

Einführung

Approximation von . . .

Konsistenz und . . .

Mehrschrittverfahren

Anwendungsbeispiele . . .

Page 1 of 26

5.1. Einführung

Differentialgleichungen

- Eines der wichtigsten Einsatzgebiete numerischer Verfahren sind **Differentialgleichungen**, also Gleichungen, die Funktionen sowie deren Ableitungen in Bezug zueinander setzen.
- Bei **gewöhnlichen Differentialgleichungen (ordinary differential equations, ODE)**, die wir im Folgenden diskutieren wollen, tritt nur eine Unabhängige auf (typischerweise die Zeit); einfache Anwendungsbeispiele sind etwa

- die Oszillation eines Pendels

$$\ddot{y}(t) = -y(t)$$

mit der Lösung

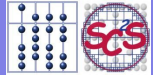
$$y(t) = c_1 \cdot \sin(t) + c_2 \cdot \cos(t);$$

- das exponentielle Wachstum

$$\dot{y}(t) = y(t)$$

mit der Lösung

$$y(t) = c \cdot e^t.$$



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 2 of 26

- Das Gegenteil von „gewöhnlich“ ist hier „partiell“: Bei **partiellen Differentialgleichungen (partial differential equations, PDE)** kommen mehrere Unabhängige vor (mehrere Raumkoordinaten oder Raum und Zeit); einfache Anwendungsbeispiele hierfür sind

- die **Poisson-Gleichung** in 2 D, die bspw. die Verformung einer am Rand eingespannten Membran unter einer äußeren Last f beschreibt:

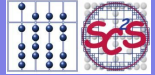
$$\Delta u(x, y) := u_{xx}(x, y) + u_{yy}(x, y) = f(x, y) \quad \text{auf } [0, 1]^2$$

(hier wird's mit der expliziten Angabe von Lösungen schon viel schwerer!);

- die **Wärmeleitungsgleichung** in 1 D, die bspw. die Temperaturverteilung in einem Metalldraht bei vorgegebener Temperatur an den Endpunkten beschreibt:

$$u_t(x, t) = u_{xx}(x, t) \quad \text{auf } [0, 1]^2$$

(hier liegt eine instationäre Gleichung vor (Zeitabhängigkeit!).



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

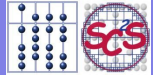
Page 3 of 26

Szenarien mit Differentialgleichungen

- Die Differentialgleichung allein bestimmt die Lösung i.A. noch nicht eindeutig (siehe die Konstanten in den Lösungen zu den gewöhnlichen Differentialgleichungen). Zusätzliche Bedingungen müssen angegeben werden (oben z.T. in Prosa umschrieben, aber noch nicht formelmäßig ausgedrückt):
 - *am Rand* eingespannte Membran,
 - vorgegebene Temperatur *an den Endpunkten* des Stabes,
 - *Startposition* eines Pendels,
 - Populationsstärke *am Anfang*.
- Solche Bedingungen treten in Erscheinung als **Anfangsbedingungen** (etwa die Populationsstärke zu Beginn der Zeitrechnung) oder als **Randbedingungen** (ein Space Shuttle soll schließlich an wohldefinierten Orten starten und landen).
- Gesucht ist dann jeweils die Funktion u , die die Differentialgleichung *und* diese Bedingungen erfüllt. Oft hat man es zudem nicht nur mit einer Differentialgleichung zu tun, sondern mit einem ganzen System von Differentialgleichungen.
- Dementsprechend spricht man von **Anfangswertproblemen (AWP)** oder **Randwertproblemen (RWP)**.
- Wir werden uns in diesem Kapitel nur mit AWP von ODE befassen, genauer mit

$$\dot{y}(t) = f(t, y(t)), \quad y(t_0) = y_0.$$

Hier benötigen wir *eine* Anfangsbedingung, da es sich um eine ODE *erster Ordnung* handelt (nur erste Ableitung).



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 4 of 26

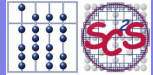
Analytische Lösbarkeit

- In einfachen Fällen sind gewöhnliche Differentialgleichungen analytisch lösbar:

- Beim obigen Beispiel $\dot{y}(t) = y(t)$ ist die Lösung offensichtlich.
- Manchmal helfen Techniken wie die **Separation der Variablen**:

$$\begin{aligned}\dot{y}(t) &= t \cdot y(t) \\ \frac{1}{y(t)} \cdot \frac{dy}{dt} &= t \\ \frac{1}{y} \cdot dy &= t \cdot dt \\ \int_{y_0}^y \frac{1}{\eta} \cdot d\eta &= \int_{t_0}^t \tau \cdot d\tau \\ \ln(y) - \ln(y_0) &= \frac{t^2}{2} - \frac{t_0^2}{2} \\ y(t) &= y_0 \cdot e^{t^2/2} \cdot e^{-t_0^2/2}.\end{aligned}$$

- Offensichtlich löst diese Funktion $y(t)$ die Differentialgleichung und erfüllt $y(t_0) = y_0$!
- Achtung! Unsere Vorgehensweise ist sehr hemdsärmelig: Darf ich durch $y(t)$ dividieren, ist der Logarithmus von $y(t)$ überhaupt definiert etc.?



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 5 of 26

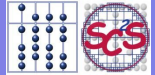
- In vielen Fällen kann man zumindest zur Lösbarkeit etwas sagen. So garantiert die **Lipschitz-Bedingung** (eine Eigenschaft von f und y)

$$\|f(t, y_1) - f(t, y_2)\| \leq L \cdot \|y_1 - y_2\|$$

für das Anfangswertproblem

$$\dot{y}(t) = f(t, y(t)), \quad y(a) = y_a, \quad t \in [a, b]$$

Existenz und Eindeutigkeit einer Lösung.



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 6 of 26

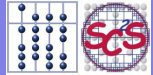
Kondition

- Doch nun zur Numerik von Anfangswertproblemen gewöhnlicher Differentialgleichungen. Besonderen Ärger machen natürlich – wie immer – schlecht konditionierte Probleme, von denen wir im Folgenden die Finger lassen wollen. Dennoch ein kleines Beispiel eines schlecht konditionierten AWP zur Abschreckung:

- Gleichung: $\ddot{y}(t) - N\dot{y}(t) - (N+1)y(t) = 0, \quad t \geq 0$
- Anfangsbedingungen (zwei wegen zweiter Ableitung):

$$y(0) = 1, \quad \dot{y}(0) = -1$$

- Lösung: $y(t) = e^{-t}$
- jetzt: gestörte Anfangsbedingung $y_\varepsilon(0) = 1 + \varepsilon$, sonst alles wie zuvor
- neue Lösung: $y_\varepsilon(t) = (1 + \frac{N+1}{N+2}\varepsilon)e^{-t} + \frac{\varepsilon}{N+2}e^{(N+1)t}$
- Man sieht: $y(t)$ und $y_\varepsilon(t)$ haben einen völlig verschiedenen Charakter; insb. geht $y(t)$ mit $t \rightarrow \infty$ gegen Null, wohingegen $y_\varepsilon(t)$ für $N+1 > 0$ unbeschränkt wächst, und zwar für beliebiges (d.h. insbesondere noch so kleines) $\varepsilon > 0$!
- Kleinste Trübungen in den Eingabedaten (hier eine der beiden Anfangsbedingungen) können sich somit desaströs auf die Lösung des AWP auswirken – ein klarer Fall von miserabler Kondition!



Einführung

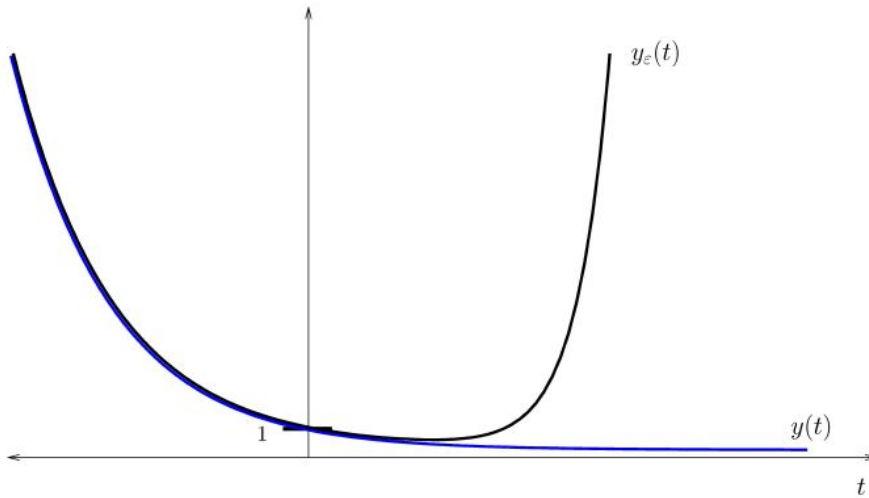
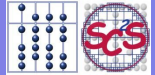
Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 7 of 26



Plot von $y(t)$ und $y_\varepsilon(t)$ für $N = 2$ und $\varepsilon = 0.01$

Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 8 of 26

5.2. Approximation von AWP durch Finite Differenzen

- Wir betrachten im Folgenden das soeben erwähnte allgemeine AWP erster Ordnung (nur erste Ableitung)

$$\dot{y}(t) = f(t, y(t)), \quad y(a) = y_a, \quad t \in [a, b]$$

und gehen dabei von dessen eindeutiger Lösbarkeit aus.

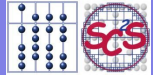
- Falls f nicht von seinem zweiten Argument y abhängt, ist dies ein einfaches Integrationsproblem!
- Startpunkt ist wie immer die *Diskretisierung*, d.h. hier: ersetze Ableitungen bzw. *Differentialquotienten* durch *Differenzenquotienten* bzw. **finite Differenzen**, bspw.

$$\frac{y(t + \delta t) - y(t)}{\delta t} \quad \text{oder} \quad \frac{y(t) - y(t - \delta t)}{\delta t} \quad \text{oder} \quad \frac{y(t + \delta t) - y(t - \delta t)}{2 \cdot \delta t}$$

für $\dot{y}(t)$ bzw., bei AWP zweiter Ordnung,

$$\frac{\frac{y(t + \delta t) - y(t)}{\delta t} - \frac{y(t) - y(t - \delta t)}{\delta t}}{\delta t} = \frac{y(t + \delta t) - 2 \cdot y(t) + y(t - \delta t)}{(\delta t)^2}$$

für $\ddot{y}(t)$ usw.



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 9 of 26

- Die erste der obigen Näherungen für $y(t)$ führt auf

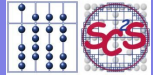
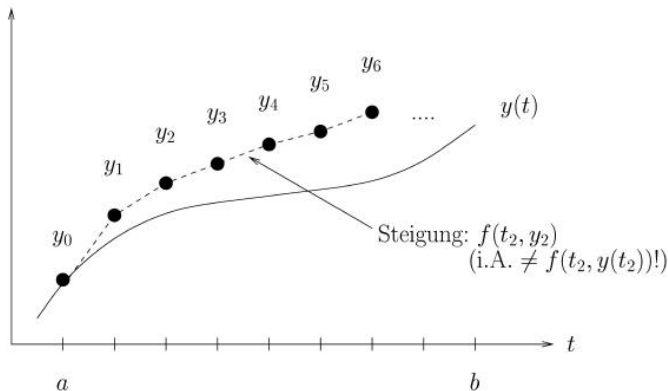
$$y(a + \delta t) \approx y(a) + \delta t \cdot f(t, y(a)), \quad \text{also}$$

$$y_{k+1} := y_k + \delta t \cdot f(t_k, y_k),$$

$$t_k = a + k\delta t, \quad k = 0, 1, \dots, N, \quad a + N \cdot \delta t = b$$

als einfachste Vorschrift zur Erzeugung diskreter Näherungen y_k für $y(t_k)$.

- Man nimmt also an der Stelle t_k den bereits berechneten Näherungswert y_k , ermittelt daraus mit Hilfe von f eine Näherung für die Steigung (Ableitung) von y und nutzt diese für eine Schätzung von y im nächsten Zeitpunkt t_{k+1} . Diese Methode wird **Euler-Verfahren** genannt.



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

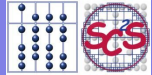
Page 10 of 26

Das Verfahren von Heun

- Neben dem soeben eingeführten Euler-Verfahren gibt es eine Reihe weiterer Verfahren für AWP von ODE, beispielsweise das Verfahren von **Heun**:

$$y_{k+1} := y_k + \frac{\delta t}{2} (f(t_k, y_k) + f(t_{k+1}, y_k + \delta t f(t_k, y_k))) .$$

- Das Grundmuster ist unverändert: Man nehme die bereits berechnete Näherung y_k in t_k , bestimme eine Näherung für die Steigung \dot{y} und ermittle daraus mittels Multiplizieren mit der Schrittweite δt eine Näherung für den Wert der Lösung y im nächsten Zeitpunkt t_{k+1} .
- Neu ist, wie man die Steigung schätzt. Beim Euler-Verfahren nimmt man einfach $f(t_k, y_k)$ her. Das Heun-Verfahren versucht, die Steigung im gesamten Intervall $[t_k, t_{k+1}]$ besser zu approximieren, indem der Mittelwert aus zwei Schätzern für \dot{y} in t_k und in t_{k+1} herangezogen wird. Das Problem, dass man y_{k+1} ja erst bestimmen will, umgeht man durch Verwendung der Euler-Schätzung als zweites Argument von f !
- Wie man leicht sieht, ist der einzelne Zeitschritt aufwändiger geworden (zwei Funktionsauswertungen von f , mehr elementare Rechenoperationen als beim einfachen Euler-Verfahren). Wir hoffen natürlich, dass wir dafür auch etwas „Besseres“ bekommen haben – doch dazu mehr im nächsten Abschnitt.



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 11 of 26

Das Verfahren von Runge und Kutta

- Das Verfahren nach **Runge** und **Kutta** geht noch einen Schritt weiter in diese Richtung:

$$y_{k+1} := y_k + \frac{\delta t}{6} (T_1 + 2T_2 + 2T_3 + T_4)$$

mit

$$T_1 := f(t_k, y_k),$$

$$T_2 := f\left(t_k + \frac{\delta t}{2}, y_k + \frac{\delta t}{2}T_1\right),$$

$$T_3 := f\left(t_k + \frac{\delta t}{2}, y_k + \frac{\delta t}{2}T_2\right),$$

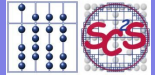
$$T_4 := f(t_{k+1}, y_k + \delta t T_3).$$

- Auch diese Vorschrift folgt unserem Grundprinzip

$$y_{k+1} := y_k + \delta t \cdot \text{Naherung_der_Steigung},$$

allerdings ist die Berechnung des Naherungswerts fur y jetzt noch einen Kick komplizierter.

- Ausgehend von der einfachen Euler-Naherung $f(t_k, y_k)$, werden durch kunstvolles Verschachteln vier geeignete Naherungswerte ermittelt, die dann – passend gewichtet – zur Approximation herangezogen werden.
- Worin liegt der Reiz dieser offenkundig noch komplizierteren Regel? Naturlich in der hoheren Genauigkeit der von ihr produzierten diskreten Naherungen fur $y(t)$, wie wir gleich sehen werden!



Einfuhrung

Approximation von ...

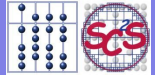
Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 12 of 26

- Man beachte die Analogien zwischen dem Euler-, Heun- und Runge-Kutta-Verfahren einerseits und ihren Pendants aus der numerischen Quadratur (Rechtecksregel, Trapezregel und Fassregel) andererseits.



Einführung

Approximation von . . .

Konsistenz und . . .

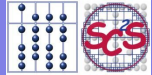
Mehrschrittverfahren

Anwendungsbeispiele . . .

Page 13 of 26

5.3. Konsistenz und Konvergenz

- Schon mehrfach ist angeklungen, dass kompliziertere Diskretisierungsregeln zu genaueren Approximationen führen. Um dies zu quantifizieren, müssen wir den Begriff der *Genauigkeit* eines Verfahrens zur Diskretisierung gewöhnlicher Differentialgleichungen griffiger machen. Zwei Dinge sind sorgsam zu trennen:
 - erstens der Fehler, der auch ohne jede Verwendung von Näherungslösungen *lokal* an jedem Punkt t_k einfach dadurch entsteht, dass man statt der Ableitungen $\dot{y}(t)$ der exakten Lösung $y(t)$ die vom Algorithmus verwendeten Differenzenquotienten einsetzt, und zwar mit dem exakten $y(t)$;
 - zweitens der Fehler, der sich insgesamt *global* im Laufe der Berechnung von a nach b , also über das gesamte betrachtete Zeitintervall, ansammelt.
- Dementsprechend unterscheiden wir zwei Arten von Diskretisierungsfehlern,
 - den **lokalen Diskretisierungsfehler** (d.h. das, was in jedem Zeitschritt neu an Fehler entsteht, auch wenn man den Differenzenquotienten mit dem exakten $y(t)$ bilden würde) sowie
 - den **globalen Diskretisierungsfehler** (d.h. das, um was man am Ende über das gesamte Zeitintervall mit seinen Berechnungen maximal daneben liegt).



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

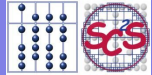
Page 14 of 26

Der lokale Diskretisierungsfehler

- Unter dem **lokalen Diskretisierungsfehler** versteht man den maximalen Fehler, der allein durch den lokalen Übergang von Differential- zu Differenzenquotienten entsteht – auch, wenn man dabei stets die exakte Lösung $y(t)$ heranzieht.
- Für das Euler-Verfahren ergibt sich der lokale Diskretisierungsfehler also zu

$$l(\delta t) := \max_{a \leq t \leq b - \delta t} \left\{ \left| \frac{y(t + \delta t) - y(t)}{\delta t} - f(t, y(t)) \right| \right\} .$$

- Bei der Maximumsbildung gehen wir in jedem Punkt von der Verfügbarkeit der lokalen Lösung aus und betrachten, was durch „Differenzen statt Ableitungen“ lokal an Fehler entsteht.
- Falls $l(\delta t) \rightarrow 0$ für $\delta t \rightarrow 0$, so wird das Diskretisierungsschema **konsistent** genannt.
- Konsistenz ist offensichtlich das Minimum, was zu fordern ist. Eine nicht konsistente Diskretisierung taugt überhaupt nichts: Wenn nicht einmal lokal in jedem Zeitschritt vernünftig approximiert wird (wenn z.B. im Nenner statt δt vielmehr $(\delta t)^2$ oder $2\delta t$ steht) und somit immer mehr Rechenaufwand eben nicht zu immer besseren Ergebnissen führt, kann man auch nicht erwarten, dass unser gegebenes AWP sinnvoll gelöst wird.



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

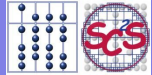
Page 15 of 26

Der globale Diskretisierungsfehler

- Der **globale Diskretisierungsfehler** bezeichnet den maximalen Fehler zwischen den berechneten Näherungen y_k und den entsprechenden Werten $y(t_k)$ der exakten Lösung $y(t)$ an den diskreten Zeitpunkten t_k :

$$e(\delta t) := \max_{k=0, \dots, N} \{ | y_k - y(t_k) | \} .$$

- Das ist natürlich die eigentlich spannende Größe – der globale Diskretisierungsfehler gibt an, wie gut die von unserem Verfahren gelieferte Approximation schlussendlich ist!
- Falls $e(\delta t) \rightarrow 0$ für $\delta t \rightarrow 0$, so wird das Diskretisierungsschema **konvergent** genannt. Das Investieren von immer mehr Rechenaufwand (immer kleinere Zeitschritte δt) führt dann auch zu immer besseren Approximationen an die exakte Lösung (verschwindender Fehler).
- Konsistenz ist der schwächere der beiden Begriffe, eher technischer Natur und oft relativ einfach zu beweisen. Konvergenz dagegen ist der stärkere Begriff (Konvergenz impliziert Konsistenz, umgekehrt nicht!), von fundamentaler praktischer Bedeutung und oft nicht ganz trivial zu zeigen.



Einführung

Approximation von ...

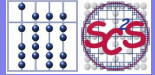
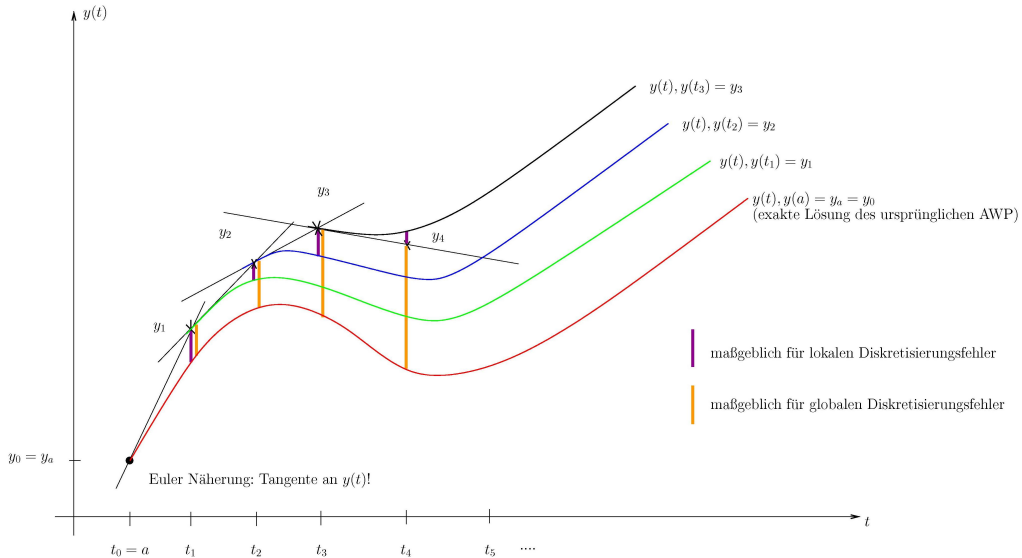
Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 16 of 26

Lokaler und globaler Diskretisierungsfehler im Vergleich



Einführung

Approximation von ...

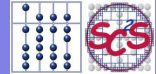
Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 17 of 26

Konsistenz und Konvergenz der Verfahren von Euler, Heun und Runge-Kutta



- Alle drei bisher vorgestellten Verfahren sind konsistent und konvergent:

- **Euler:** Verfahren *erster Ordnung*, d.h.

$$l(\delta t) = O(\delta t), \quad e(\delta t) = O(\delta t);$$

- **Heun:** Verfahren *zweiter Ordnung*, d.h.

$$l(\delta t) = O((\delta t)^2), \quad e(\delta t) = O((\delta t)^2);$$

- **Runge-Kutta:** Verfahren *vierter Ordnung*, d.h.

$$l(\delta t) = O((\delta t)^4), \quad e(\delta t) = O((\delta t)^4).$$

- Hier wird der Qualitätsunterschied deutlich: Je höher die Ordnung des Verfahrens, desto mehr bringt eine Aufwandssteigerung. Bei Halbierung der Schrittweite δt bspw. wird der Fehler bei Euler bzw. Heun bzw. Runge-Kutta asymptotisch um einen Faktor 2 bzw. 4 bzw. 16 reduziert.
- Die teureren Verfahren sind also (zumindest asymptotisch) die leistungsfähigeren.

Einführung

Approximation von ...

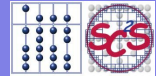
Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 18 of 26

- Natürlich sind wir noch nicht zufrieden. Die Zahl der *Auswertungen* der Funktion f für verschiedene Argumente hat stark zugenommen (vgl. die Runge-Kutta-Formeln: T_2 , T_3 und T_4 erfordern je eine zusätzliche Auswertung von f). In der numerischen Praxis ist f typischerweise sehr kompliziert (oft muss für eine einzige Auswertung von f eine weitere Differentialgleichung gelöst werden), so dass bereits eine Auswertung von f mit hohem Rechenaufwand verbunden ist.
- Im nächsten Abschnitt lernen wir deshalb, wie's auch anders geht.



Einführung

Approximation von . . .

Konsistenz und . . .

Mehrschrittverfahren

Anwendungsbeispiele . . .

Page 19 of 26

5.4. Mehrschrittverfahren

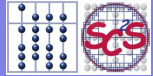
- Die bisherigen Verfahren sind allesamt so genannte **Einschrittverfahren**: Für die Berechnung von y_{k+1} werden keine weiter als t_k zurückliegenden Zeitpunkte herangezogen (sondern – wie gesagt – neue Auswertestellen).
- Anders bei den **Mehrschrittverfahren**: Hier werden keine zusätzlichen Auswertestellen von f produziert, sondern vielmehr werden ältere (und schon berechnete) Funktionswerte wiederverwertet, zum Beispiel in t_{k-1} beim **Adams-Bashforth-Verfahren zweiter Ordnung**:

$$y_{k+1} := y_k + \frac{\delta t}{2} (3f(t_k, y_k) - f(t_{k-1}, y_{k-1}))$$

(die Konsistenz zweiter Ordnung kann leicht gezeigt werden).

- Verfahren noch höherer Ordnung können analog konstruiert werden, indem man auf noch weiter zurückliegende Zeitpunkte t_{k-i} , $i = 1, 2, \dots$, zurückgreift.
 - Das Prinzip ist dabei ein guter alter Bekannter von der Quadratur: Ersetze f durch ein Polynom p von passendem Grad, das f in den betrachteten (t_i, y_i) interpoliert, und verwende dann dieses p gemäß
- $$y_{k+1} := y_k + \int_{t_k}^{t_{k+1}} \dot{y}(t) dt = y_k + \int_{t_k}^{t_{k+1}} f(t, y(t)) dt \approx y_k + \int_{t_k}^{t_{k+1}} p(t) dt,$$
- um y_{k+1} zu berechnen (das Polynom p ist leicht zu integrieren).
- Zu Beginn, d.h. solange es noch nicht genügend „alte“ Werte gibt, benutzt man in der Regel ein passendes Einschrittverfahren.

- Die auf diesem Interpolationsprinzip beruhenden Mehrschrittverfahren heißen **Adams-Bashforth-Verfahren**.



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

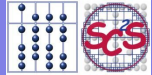
Anwendungsbeispiele ...

Page 20 of 26

Der Begriff der Stabilität

- Die Verfahren von Euler, Heun und Runge-Kutta sind konsistent und konvergent.
- Dies gilt auch für die soeben eingeführte Klasse der Mehrschrittverfahren vom Adams-Bashforth-Typ: auch sie sind sowohl konsistent als auch konvergent.
- Dennoch zeigt sich bei den Mehrschrittverfahren, dass Konsistenz und Konvergenz nicht immer zugleich gelten. Um konvergent zu sein, muss ein konsistentes Verfahren zusätzlich **stabil** sein (im Sinne unserer Definition eines numerisch stabilen Algorithmus aus Kapitel 1).
- Der Nachweis von Stabilität ist damit eminent wichtig. Eine genaue Diskussion würde an dieser Stelle zu weit führen, man sollte sich aber dennoch folgende Merkregel einprägen:

Konsistenz + Stabilität \Rightarrow Konvergenz .



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 21 of 26

- Um die Auswirkungen von Instabilität zu erleben, betrachten wir als Diskretisierungsvorschrift die **Mittelpunktsregel**

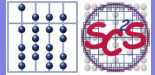
$$y_{k+1} := y_{k-1} + 2\delta t f_k.$$

- Die Mittelpunktsregel ist offensichtlich ein 2-Schritt-Verfahren.
- Ihre Konsistenz sieht man leicht daran, dass der Differenzenquotient

$$\frac{y_{k+1} - y_{k-1}}{2\delta t}$$

für $\delta t \rightarrow 0$ tatsächlich gegen die erste Ableitung von y in t_k und somit gegen f konvergiert.

- Die Konsistenzordnung beträgt 2 (leicht über Taylor-Entwicklung zu zeigen).



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 22 of 26

Ein Beispiel für Instabilität

- Nun wenden wir die Mittelpunktsregel auf folgendes AWP an:

$$\dot{y}(t) = -2y(t) + 1, \quad y(0) = 1, \quad t \geq 0,$$

mit der Lösung

$$y(t) = \frac{1}{2}(e^{-2t} + 1).$$

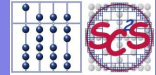
- Wir erhalten somit die Vorschrift

$$y_{k+1} := y_{k-1} + 2\delta t(-2y_k + 1) = y_{k-1} - 4\delta t y_k + 2\delta t, \quad y_0 = 1.$$

- Mit den exakten Werten $y(0)$ und $y(\delta t)$ als Startwerten liefert der Algorithmus folgende Resultate:

δt	y_9	y_{10}	y_{79}	y_{80}	y_{999}	y_{1000}
1.0	-4945.9	20953.9				
0.1	0.5820	0.5704	-1725.3	2105.7		
0.01	0.9176	0.9094	0.6030	0.6010	-154.6	158.7

- D.h.: Für jede (noch so kleine) Schrittweite δt oszilliert die Folge der berechneten y_k mit unbeschränkten Absolutwerten für $k \rightarrow \infty$, anstatt gegen die exakte Lösung $1/2$ zu konvergieren. Somit haben wir zwar Konsistenz, aber offenkundig *keine Konvergenz* – die Mittelpunktsregel ist kein stabiler Algorithmus!



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

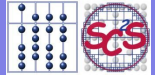
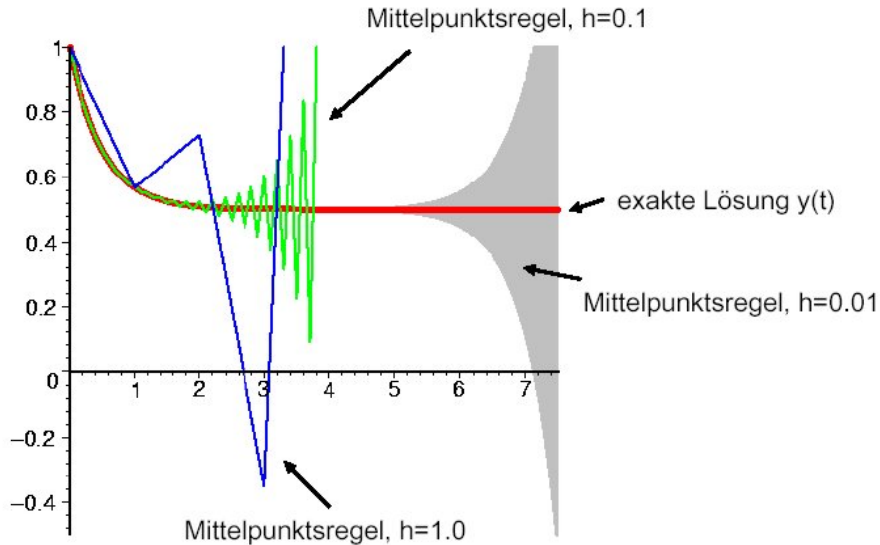
Anwendungsbeispiele ...

Page 23 of 26

Beispiel zu Stabilität und Mittelpunktsregel

- jetzt das Ganze im Schaubild:

AWP war $\dot{y}(t) = -2y(t) + 1$, $y(0) = 1$, **Lösung** $y(t) = \frac{e^{-2t+1}}{2}$



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 24 of 26

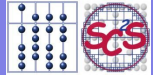
Was dem Mut zur Lücke zum Opfer fällt ...

- **Implizite Verfahren:** Für viele ODE führen **explizite** Verfahren, bei denen y_{k+1} unmittelbar berechnet werden kann, zu großen Problemen (Instabilitäten, kleine Zeitschritte erforderlich). Hier helfen implizite Verfahren wie das **implizite Euler-Verfahren:**

$$y_{k+1} = y_k + \delta t \cdot f(t_{k+1}, y_{k+1}).$$

Allerdings muss man sich hier für die konkrete Berechnung von y_{k+1} etwas einfallen lassen – es ist nur implizit gegeben.

- **Systeme von ODE:** Wir haben nur eine skalare Unbekannte $y(t)$ betrachtet, meistens treten diese aber in Rudeln als Vektoren auf.
- **ODE höherer Ordnung:** Wir haben nur ODE erster oder (in manchen Beispielen) zweiter Ordnung betrachtet. Es gibt aber durchaus auch ODE, in denen höhere Ableitungen auftreten.
- **Randwertprobleme:** Wir haben nur Anfangswertprobleme betrachtet, bei denen alle zusätzlichen Bedingungen (neben der eigentlichen ODE) als Anfangsbedingungen auftreten. In vielen Szenarien haben wir es aber mit Randbedingungen zu tun. Wenn bspw. die optimale Flugbahn eines Space Shuttle bestimmt werden soll, dann will die NASA naheliegenderweise Start- und Landepunkt vorgeben können!!
- ...



Einführung

Approximation von ...

Konsistenz und ...

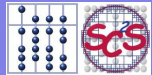
Mehrschrittverfahren

Anwendungsbeispiele ...

Page 25 of 26

5.5. Anwendungsbeispiele für ODE in der Informatik

- **Regelungstechnik:** Schon seit langem werden Computer zur Steuerung und Regelung von Anlagen und Prozessen eingesetzt (Prozessrechner). Die mathematische Beschreibung solcher Regelsysteme stützt sich oft auf ODE – dementsprechend müssen diese zur erfolgreichen Regelung gelöst werden.
- **Optimalsteuerung:** Die optimale Steuerung von Systemen ist eine Aufgabe, die z.B. in der Robotik essentiell ist: Roboter sollen sich zeit- oder energieoptimal bewegen, in einem Fußballspiel wie beim *RoboCup* vielleicht sogar als autonome Agenten interagieren. Die Modellierung solcher Optimalsteuerungsprobleme führt auf ODE.
- **Visualisierung:** Nach einer Strömungssimulation (Auto im Windkanal, Tornado-Vorhersage etc.) will man die berechneten Resultate (also konkret die Strömungsverhältnisse) visualisieren. Hierzu bringt man virtuelle Teilchen in das Strömungsfeld ein und stellt deren Bahnen dar, bspw. um einen Eindruck von den Verwirbelungen zu erhalten. Die Berechnung dieser Bahnen erfordert das Lösen von ODE.
- **Chip-Layout:** Die Beschreibung von Schaltkreisen stützt sich auf die Kirchhoffschen Gesetze, die – bei zeitlich veränderlichen Unbekannten – nicht mehr einfach algebraische Gleichungen sind, sondern ODE. Schaltkreissimulation ist somit ein wichtiges Anwendungsgebiet für ODE.



Einführung

Approximation von ...

Konsistenz und ...

Mehrschrittverfahren

Anwendungsbeispiele ...

Page 26 of 26