

Computational Physics

Florian Bruckner

Christian Doppler Laboratory of Advanced Magnetic Sensing and Materials, Faculty of Physics, University of Vienna, Austria

2015-02-25

1 Gewöhnliche Differentialgleichungen (ODEs)

- Wichtigsten Einsatzgebiete numerischer Verfahren sind **Differentialgleichungen**, also Gleichungen, die Funktionen sowie deren Ableitungen in Bezug zueinander setzen.
- Bei **Gewöhnlichen** Differentialgleichungen (Ordinary Differential Equations, ODE), tritt nur eine Unabhängige auf (typischerweise die Zeit).

Beispiel: Oszillation eines Pendels $\ddot{y} = -y(t)$

Analytische Lösung: $y(t) = c_1 \sin(t) + c_2 \cos(t)$ gegeben.

Konstanten c_1 und c_2 werden durch Anfangsortes $y(t_0) = y_0$ und der Anfangsgeschwindigkeit $\dot{y}(t_0) = v_0$ bestimmt (Anfangswertproblem).

- Bei partiellen Differentialgleichungen (partial differential equations, PDE) kommen mehrere Unabhängige vor (mehrere Raumkoordinaten oder Raum und Zeit)

Beispiel: Poisson Gleichung $\Delta u(x, y) = f(x, y)$

- Verformung $u(x, y)$ einer am Rand eingespannten Membran
- Last $f(x, y)$
- Angabe analytischer Lösungen nur für Spezialfälle möglich
- Numerische Methoden erforderlich

- Die Differentialgleichung allein bestimmt die Lösung i.A. noch nicht eindeutig
- Zusätzliche Bedingungen müssen angegeben werden
 - Anfangswertproblem
 - Randwertaufgaben
 - Anfangsrandwertproblem

Beispiele für Randwerte

- Verformung am Rand eingespannte Membran
- Start- und Ziel-Position eines Space-Shuttles
- vorgegebene Temperatur an den Endpunkten des Stabes
- Startposition und Geschwindigkeit eines Pendels
- Populationsstärke zu Beginn der Zeitrechnung
- Wärmeleitung

■ Allgemeinerer Fall von gewöhnlichen Differentialgleichungen

System gewöhnliche Differentialgleichung erster Ordnung

Ein System gewöhnlicher Differentialgleichungen erster Ordnung ist allgemein gegeben durch

$$\dot{\mathbf{Y}} = \mathbf{F}(t, \mathbf{Y})$$

$$\text{wobei } \mathbf{Y}(t) = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} \text{ und } \mathbf{F}(t) = \begin{bmatrix} F_1(t, Y_1, Y_2, \dots, Y_n) \\ \vdots \\ F_n(t, Y_1, Y_2, \dots, Y_n) \end{bmatrix}.$$

- Jede gewöhnliche Differentialgleichung zweiter Ordnung kann in ein System von Differentialgleichungen erster Ordnung umgewandelt werden

Transformation von Differentialgleichung zweiter Ordnung in erste Ordnung

Sei eine Differentialgleichung zweiter Ordnung gegeben durch $\ddot{y} = g(t, y, \dot{y})$. Dann folgt durch die Transformation $\dot{y}(t) = y_2$ und $\dot{y}_2 = g(t, y, \dot{y})$ das System erster Ordnung

$$\dot{\mathbf{Y}} = \mathbf{F}(t, \mathbf{Y}) \text{ mit } \mathbf{Y}(t) = \begin{bmatrix} y \\ y_2 \end{bmatrix} \text{ und } \mathbf{F}(t) = \begin{bmatrix} y_2 \\ g(t, y, y_2) \end{bmatrix}.$$

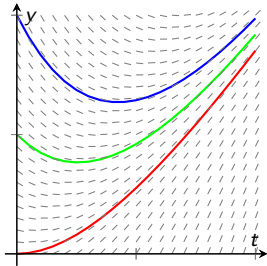
- Beispiel

Für die Eindeutigkeit von gewöhnlichen Differentialgleichungen gilt folgender Satz:

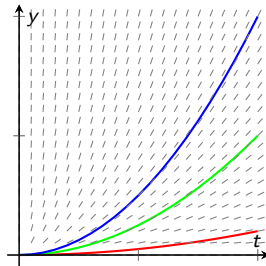
Eindeutigkeitsbedingung

Die Differentialgleichung $\dot{y} = f(t, y)$ besitzt eine eindeutige Lösung für $a \leq t \leq b$, falls f und die erste partielle Ableitung f_y im besagten Intervall stetig sind.

$$y' = t - y + 1$$



$$y' = 2y/t \text{ (Beispiel)}$$



Die Kondition eines Systems bestimmt wie sehr sich Änderungen der Anfangsdaten in der Lösung auswirken.

Gut konditionierte Probleme

- Kleine Änderungen in den Eingabedaten führen nur zu kleinen Änderungen in der Lösung.
- Störungen in der Eingabe sind also relativ unkritisch.
- Es lohnt sich, in einen guten Algorithmus zu investieren.

Schlecht konditionierte Probleme

- Kleinste Änderungen in den Eingabedaten führen zu völlig verschiedenen Lösungen.
- Hier tun sich im Allgemeinen auch exzellente Algorithmen schwer.
- Schlecht konditionierte Probleme sind numerisch nur sehr schwer (im Extremfall auch gar nicht) zu behandeln.
- Ungenauigkeit beim Rechnen kann das berechnete Resultat völlig verfälschen.

Als Beispiel eines schlecht konditionierten Problems betrachten wir das Anfangswertproblem:

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

Fall 1: $y(0) = 0$

- Es ergibt sich die Lösung

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

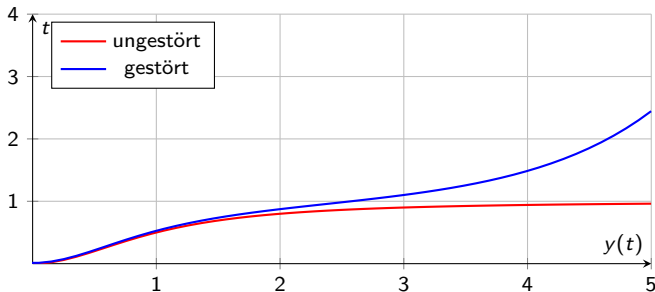
Fall 2: $y(0) = \epsilon$

- Anfangsbedingung nur leicht gestört
- Es ergibt sich eine völlig andere Lösung

$$y_\epsilon(t) = \epsilon e^t + \frac{t^2}{1 + t^2}$$

Lösung des Beispiels

- Völlig unterschiedliche Lösungen für Fall 1 und Fall 2.
- Fall 1: $y(t) \rightarrow 1$ für $t \rightarrow \infty$
- Fall 2: $y_\epsilon(t) \rightarrow \infty$ für beliebig kleines $\epsilon > 0$.
- Kleinste Trübungen in den Eingabedaten wirken sich desaströs auf die Lösung des AWP auswirken - ein klarer Fall von schlechter Kondition!



- Beispiel: Kondition eines linearen Gleichungssystems
- Beispiel: Kondition von $\ddot{y} - y = 0$, $y(0) = 1$, $\dot{y}(0) = -1$

- Einfachstes Verfahren: Vorwärts Euler Verfahren
- Kaum verwendet aber verdeutlicht und erklärt die Methode und die Begriffe
- Beim Euler Verfahren wird der Differentialquotienten durch Differenzenquotienten bzw. finite Differenzen ersetzt

$$\dot{y}(t) \approx \frac{y_{i+1} - y_i}{\Delta t} = f(t, y)$$

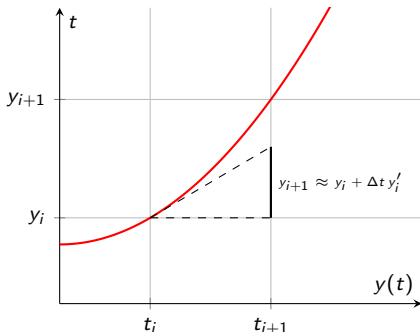
Konvention

y_{i+1}	Lösung zur Zeit t_{i+1}
y_i	Lösung zur Zeit t_i
Δt	Zeitschritt

Explizites Euler Verfahren (1)

- Differentialquotienten durch Differenzenquotienten ersetzen
- Für die rechte Seite der gewöhnlichen Differentialgleichung wird die Lösung aus dem letzten Zeitschritt eingesetzt

$$\frac{y_{i+1} - y_i}{\Delta t} = f(t_i, y_i)$$
$$y_{i+1} = y_i + \Delta t f(t_i, y_i)$$



Explizites Euler Verfahren (2)

- Exakte Form: $y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} f(t, y) dt = y_i + \Delta t \cdot \text{mittlere Steigung}$
- Integral wird approximiert: Beim expliziten Euler Verfahren wird es durch eine einfache Rechteckregel approximiert

$$\int_{t_i}^{t_{i+1}} f(t, y) dt = \Delta t f(t_i, y_i) \Rightarrow y_{i+1} = y_i + \Delta t f(t_i, y_i)$$

- Vergleich mit Taylor-Reihe zeigt dass Euler Verfahren Terme der Ordnung Δt^2 vernachlässigt

$$y_{i+1} = y_i + \Delta t \dot{y}_i + \frac{\Delta t^2}{2} \ddot{y}(\zeta) \quad \text{für } \zeta \in [t_i, t_{i+1}]$$

- Der Term $\ddot{y}(\zeta)$ kann verwendet werden um den lokalen Fehler der Methode abzuschätzen

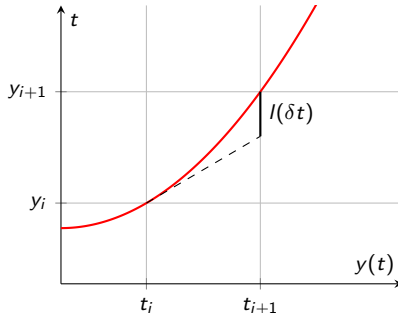
- Verschiedene Verfahren sollen Quantifiziert werden
- Unterscheidung lokaler und globaler Fehler
- Sehr wichtig ist auch der Begriff der Stabilität

Lokaler Diskretisierungsfehler - Konsistenz

- Unter dem lokalen Diskretisierungsfehler versteht man die Abweichung der Fortschrittsrichtung von der exakten Richtung

$$l(\Delta t) = y_{\text{exact}}(t_i + \Delta t) - y_{\text{num}}(t_i + \Delta t)$$

- Falls $l(\delta t) \rightarrow 0$ für $\delta t \rightarrow 0$ so wird das Diskretisierungsschema **konsistent** genannt.
- Konsistenz ist eine Mindestanforderung für jedes Diskretisierungsschema



Beispiel Euler Verfahren

$$\begin{aligned}l(\Delta t) &= y_{\text{exact}}(t_i + \Delta t) - y_{\text{num}}(t_i + \Delta t) \\y_{\text{exact}}(t_i + h) &= y(t_i) + \Delta t \dot{y}(t_i) + O(\Delta t^2) = y(t_i) + \Delta t f(t_i) + O(\Delta t^2) \\y_{\text{num}}(t_i + h) &= y(t_i) + \Delta t f(t_i) \\ \Rightarrow l(\Delta t) &= O(\Delta t^2)\end{aligned}$$

Der lokale Fehler beim Euler Verfahren ist somit von der Ordnung $O(\Delta t^2)$. Es handelt sich um ein Verfahren 1. Ordnung.

Globaler Diskretisierungsfehler - Konvergenz

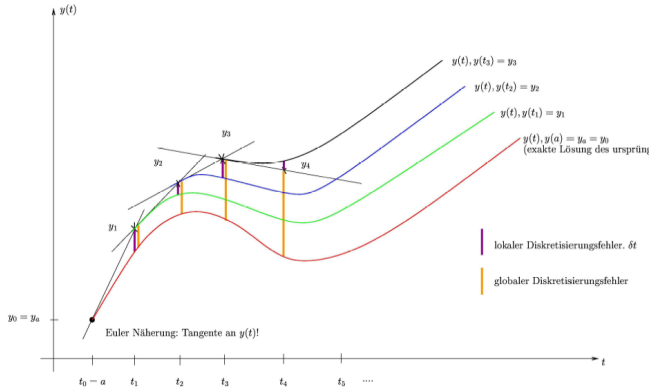
- Globale Diskretisierungsfehler ist maximale Fehler zwischen den berechneten Näherungen y_k und den entsprechenden Werten $y(t_k)$ der exakten Lösung $y(t)$ an den diskreten Zeitpunkten t_k

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

- Falls $e(\Delta t) \rightarrow 0$ für $\delta t \rightarrow 0$ so wird das Diskretisierungsschema **konvergent** genannt.
- Für konvergente Methoden führt ein kleiner Zeitschritt auf bessere Lösungen
- Konsistenz + Stabilität \Rightarrow Konvergenz (Dahlquist Äquivalenz Theorem, Lax-Richtmyer Theorem)
- **Problem:** Konvergenz oft schwer zu zeigen
- **Beispiel:** Abschätzung globaler Fehler

Konsistenz - Konvergenz

- Konsistenz ist der schwächere Begriff, (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 trivial zu zeigen.



Explizite Zeitintegrationsverfahren (1)

- Einfachstes explizites Zeitintegrationsverfahren ist Eulerverfahren
- Rechte Seite nur von y_i und t_i abhängig
- Taylor-Terme höherer Ordnung können durch mehrfache Auswertungen der Systemfunktion approximiert werden
- Oft auch Runge-Kutta Verfahren genannt

Beispiel Heun Verfahren

$$y_{i+1} = y_i + \frac{\Delta t}{2} \left[f(t_i, y_i) + f(t_{i+1}, \underbrace{y_i + \Delta t f(t_i, y_i)}_{\text{Euler-Step}}) \right]$$

Der lokale Fehler ist $l(\Delta t) = O(\Delta t^3)$.

- Wie man leicht sieht, ist der einzelne Zeitschritt aufwändiger geworden (zwei Funktionsauswertungen von f , mehr elementare Rechenoperationen als beim einfachen Euler-Verfahren).
- Aber die Ordnung des Verfahrens hat zugenommen.
- In Bezug auf den lokalen Fehler ist eine Erhöhung der Ordnung einer Schrittweiten-Reduktion vorzuziehen (Allerdings führt die höhere Ordnung meist zu schlechterer Stabilität!).

Beispiel Runge-Kutta Verfahren 4. Ordnung

$$y_{i+1} = y_i + \frac{\Delta t}{6}(T_1 + 2T_2 + 2T_3 + T_4)$$

$$T_1 = f(t_i, y_i)$$

$$T_2 = f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{\Delta t}{2} T_1\right)$$

$$T_3 = f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{\Delta t}{2} T_2\right)$$

$$T_4 = f(t_i + \Delta t, y_i + \Delta t T_3)$$

Der lokale Fehler ist $l(\Delta t) = O(\Delta t^5)$.

Explizite Zeitintegrationsverfahren (4)

- Allgemeine Darstellung mittels **Butcher-Tableaus** erleichtert Implementierung

Allgemeines Runge-Kutta Verfahren s-ter Stufe

$$y_{i+1} = y_i + \Delta t \sum_{i=1}^s b_i K_i$$

$$K_i = f \left(t_i + c_i \Delta t, y_i + \Delta t \sum_{j=1}^s a_{ij} K_j \right)$$

Butcher-Tableau:

c_1	a_{11}	\dots	a_{1s}
\vdots	\vdots		\vdots
c_s	a_{s1}	\dots	a_{ss}
<hr/>			
	b_1	\dots	b_s

Maximale Ordnung von Runge-Kutta Verfahren

Stufe s	1	2	3	4	5	6	7	8	9
Ordnung p	1	2	3	4	4	5	6	6	7

- Beispiel: Ableitung Heun-Ralston-Methode
- Beispiel: Butcher-Tableau für vorgestellte Runge-Kutta Verfahren

- Euler, Heun, Runge-Kutta konsistent
- Konvergenz nicht immer gegeben. Um konvergent zu sein muss ein konsistentes Verfahren zusätzlich stabil sein
- Stabilität: unter dem Einfluss von Rundungs- und Verfahrensfehlern für gut konditionierte Probleme akzeptable Resultate produziert.
- Ein stabiler Algorithmus kann durchaus große Fehler liefern - etwa, wenn das zu lösende Problem schlecht konditioniert ist.

Dahlquist Equivalenz Theorem (Lax-Richtmyer Theorem)

Konsistenz + Stabilität \Rightarrow Konvergenz

- Anwendung auf Standard-Problem $\dot{y} = \lambda y$
- Allgemeine Probleme können oft durch Linearisierung und Diagonalisierung auf das Standard-Problem zurückgeführt werden
- Einführung der Stability-Funktion $R(h\lambda)$: $y_{i+1} = R(h\lambda) y_i$
- Untersucht wird das Verhalten von y_i für $t \rightarrow \infty$
- Exakte Lösung: $y(t) = e^{\lambda t}$
- Gewünscht ist also ein Abklingen der Lösung falls $\operatorname{Re}(\lambda) < 0$

Beispiel - Explizites Euler Verfahren

- Standard-Problem $\dot{y} = \lambda y$

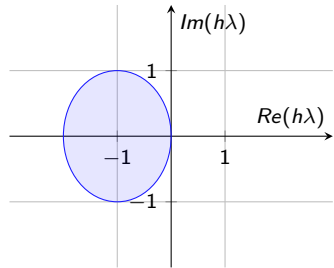
$$y_{i+1} = \underbrace{(1 + h\lambda)}_{R(h\lambda)} y_i$$

- Auflösen der Rekursion

$$y_i = R(h\lambda)^i y_0$$

- Stabilitätsbedingung

$$|R(h\lambda)| < 1$$



Beispiel - Explizites Euler Verfahren

- Standard-Problem $\dot{y} = \lambda y \quad y(0) = 1$
- Exakte Lösung

$$y_i = e^{\lambda t} = e^z$$

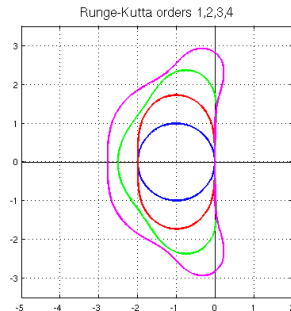
- Stability-Function für Methode p -ter Ordnung

$$y_1 = R(z) = \sum_{n=0}^p \frac{z^n}{n!} + O(z^{p+1})$$

$$R(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \dots$$

- Stabilitätsbedingung

$$|R(h\lambda)| < 1$$

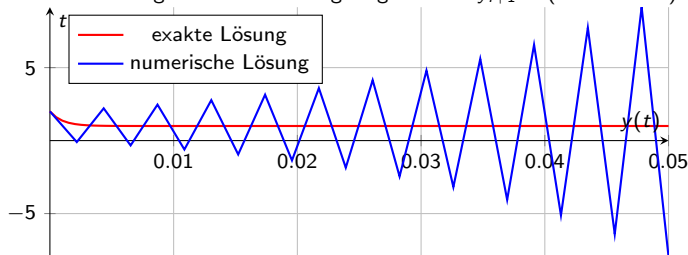


Steife Differentialgleichungen

- Betrachten wir das einfache Beispiel das durch die Gleichung

$$\dot{y} = -1000y + 1000 \quad y(0) = 2$$

- Dieses AWP ist gut konditioniert: Die gestörte Anfangsbedingung $y(0) = 2 + \epsilon$ erzeugt eine nur minimal verfälschte Lösung
- Die exakte Lösung ist $y(t) = e^{-1000t} + 1$
- Probleme entstehen durch Instabilität bei zu großen Zeitschritten
- Beispiel - Explizites Euler Verfahren $\Delta t = 0.0021$
- Durch vollständige Induktion kann gezeigt werden $y_{i+1} = (1 - 1000\Delta t)^{i+1} + 1$



Problem bei steifen Differentialgleichungen

- Exakte Lösung konvergiert schnell gegen ihren Grenzwert
- ABER!! Die mathematische Folge durch das explizite Euler Verfahren divergiert, falls $\Delta t > 0.002$ wird
- Wir werden zu einer extrem feinen Schrittweite gezwungen
- Nicht erforderlich da exakte Lösung $y(t)$ sich langsam mit Zeit ändert (außer nahe bei $t=0$)

Definitionen von Steifigkeit

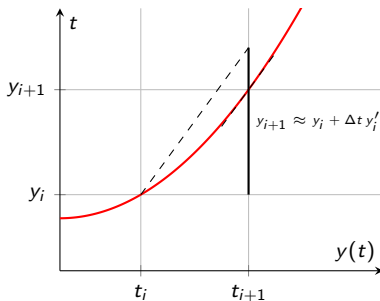
- Differentialgleichungen werden steif genannt, wenn sie einen Term der Form $e^{-\alpha t}$ enthalten, wobei α eine große, positive Konstante ist
- Beispiele: Schwingungen, chemischen Reaktionen und elektrischen Stromkreisen
- Steife Probleme haben ihren Namen von der Bewegung von Feder und Massensystemen, die große Federkonstanten besitzen
- Lösung: Numerische Methoden mit größerem Stabilitätsgebiet (implizite Methoden)

Implizites Euler Verfahren

- Differentialquotienten durch Differenzenquotienten ersetzen
- Für die rechte Seite der gewöhnlichen Differentialgleichung wird die Lösung aus dem **zukünftigen** Zeitschritt eingesetzt
- Da y_{i+1} nur implizit gegeben ist muss in jedem Zeitschritt ein Gleichungssystem gelöst werden

$$\frac{y_{i+1} - y_i}{\Delta t} = f(t_{i+1}, y_{i+1})$$

$$y_{i+1} = y_i + \Delta t f(t_{i+1}, y_{i+1})$$



Beispiel - Implizites Euler Verfahren

- Standard-Problem $\dot{y} = \lambda y$

$$y_{i+1} = y_i + h\lambda y_{i+1}$$

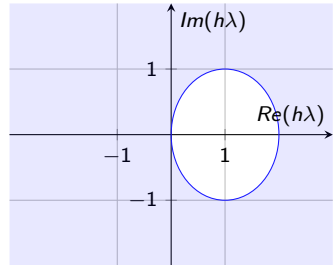
$$y_{i+1} = \underbrace{\frac{1}{1 - h\lambda}}_{R(h\lambda)} y_i$$

- Auflösen der Rekursion

$$y_i = R(h\lambda)^i y_0$$

- Stabilitätsbedingung

$$|R(h\lambda)| < 1$$



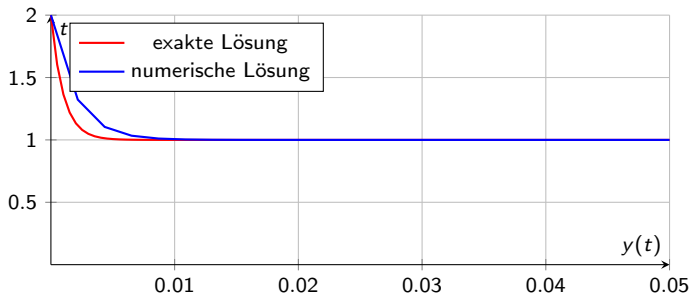
Steiße Differentialgleichungen (implizite Euler Methode)

- Betrachten wir das gleiche Beispiel wie zuvor

$$\dot{y} = -1000y + 1000 \quad y(0) = 2$$

- Die exakte Lösung ist $y(t) = e^{-1000t} + 1$
- Beispiel - Implizites Euler Verfahren $h = 0.0021$

$$y_{i+1} = \frac{y_i + 1000\Delta t}{1 + 1000\Delta t} = \left(\frac{1}{1 + 1000\Delta t} \right)^{i+1} + 1$$



- Allgemeine Stability-Function für Runge-Kutta Verfahren

$$R(z) = 1 + z\mathbf{b}^T (\mathbb{1} - z\mathbf{A})^{-1} \mathbb{1}$$

Stabilitätseigenschaften

0-Stability $|R(z)| < 1$ für $z \rightarrow 0$

Für infinitesimal kleine Zeitschritte ist Stabilität gegeben. Das Äquivalenztheorem von Dahlquist beweist daher auch Konvergenz

A-Stability $|R(z)| < 1 \quad \forall \operatorname{Re}(z) < 0$

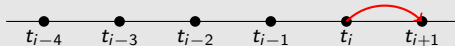
Die komplette linke Halbebene ist stabil. Wünschenswert für die Lösung steifer Probleme. Ähnlich ist $A(\alpha)$ -Stabilität, wo das Stabilitätsgebiet zumindest einen Winkel α zur x -Achse einschließt.

L-Stability $|R(z)| \rightarrow 0$ für $z \rightarrow \infty$

Stellt sicher, dass hochfrequente Eigenmoden stark gedämpft werden.

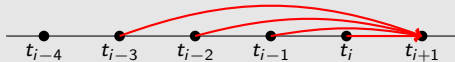
Wiederholung Einschrittverfahren

- Die bisherigen Verfahren sind allesamt so genannte Einschrittverfahren
- Für die Berechnung von y_{i+1} werden keine weiter als t_i zurückliegenden Zeitpunkte herangezogen, sondern neue Auswertestellen gewählt
- **Nachteil:**
 - Rechte Seite der Differentialgleichung muss oft ausgewertet werden
 - Kann zu erheblichem Rechenaufwand führen (Bei vielen Anwendung müssen für die Berechnung der rechten Seite partielle Differentialgleichungen gelöst werden)



Idee Mehrschrittverfahren

- **Mehrschrittverfahren:** Keine zusätzlichen Auswertestellen von f verwenden, sondern ältere (und schon berechnete) Funktionswerte wiederverwerten
- Zum Beispiel in t_{i-1} beim Adams-Bashforth-Verfahren zweiter Ordnung



Adams-Bashforth-Verfahren zweiter Ordnung

$$y_{i+1} = y_i + \frac{\Delta t}{2} (3f(t_i, y_i) - f(t_{i-1}, y_{i-1}))$$

- Adams-Bashforth Verfahren sind explizite Mehrschritt-Verfahren, bei denen frühere Funktionswerte wiederverwendet werden
- Es gibt auch implizite Verfahren, die dann bessere Stabilitätseigenschaften aufweisen (diese werden **Adams-Moulton Verfahren** genannt)
- **Idee:** Ersetze f durch ein Polynom p von passendem Grad. Diese Polynom ist dann einfach zu integrieren

$$y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} \dot{y}(t) dt = y_i + \int_{t_i}^{t_{i+1}} f(t, y(t)) dt \approx \int_{t_i}^{t_{i+1}} p(t) dt$$

- **Beginn:** solange es noch nicht genügend alte Werte gibt, benutzt man in der Regel ein passendes Einschrittverfahren

Mittelpunktsregel

$$y_{i+1} = y_{i-1} + 2\Delta t f(t_i, y_i)$$

- Die Mittelpunktsregel ist offensichtlich ein 2-Schritt-Verfahren
- Ihre Konsistenz sieht man leicht daran, dass der Differenzenquotient für $\Delta t \rightarrow 0$ tatsächlich gegen die erste Ableitung von y in t_k und somit gegen f konvergiert

Mittelpunktsregel - Stabilitätsproblem

- Wir wenden die Mittelpunktsregel auf folgendes AWP an

$$\dot{y} = -2y(t) + 1, \quad y(0) = 1 \quad \Rightarrow \quad y(t) = \frac{1}{2} (e^{-2t} + 1)$$

- Die Mittelpunktsregel gibt

$$y_{i+1} = y_{i-1} + 2\Delta t(-2y_i + 1) = y_{i-1} - 4\Delta t y_i + 2\Delta t$$

- Mit den exakten Werten y_0 und y_1 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

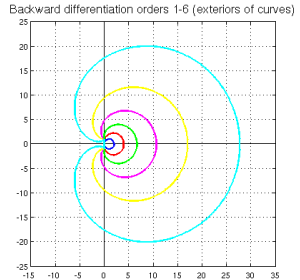
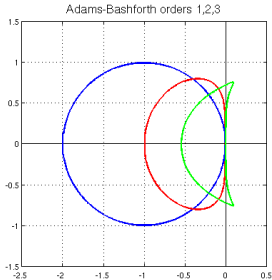
Für jede (noch so kleine) Schrittweite Δt oszilliert die Folge der berechneten y_k . Somit haben wir zwar Konsistenz, aber offenkundig keine Konvergenz. Die Mittelpunktsregel ist kein stabiler Algorithmus (für dieses Problem).

Allgemeines Mehrschritt Verfahren

$$\sum_{i=0}^{K_1} \alpha_{n,i} y^{n-i} + h \sum_{j=0}^{K_2} \beta_{n,j} \underbrace{\dot{y}^{n-j}}_{f_{n-j}} = 0$$

Adams-Moulton $K_1 = 1, K_2 = q \quad (q = 1 \dots 12)$

BDF $K_1 = q, K_2 = 0 \quad (q = 1 \dots 5)$



Symplektische Integratoren

- gut geeignet für die Beschreibung von Hamilton Systemen

$$\dot{q} = \frac{\partial H(p, q)}{\partial p} \qquad \dot{p} = -\frac{\partial H(p, q)}{\partial q}$$

- Energie-Erhaltung $E = \frac{1}{2}(p^2 + q^2) = \text{const}$ soll auch im Diskreten gelten
- Explizite Euler-Methode liefert $p_{n+1}^2 + q_{n+1}^2 = (p_n^2 + q_n^2) \cdot (1 + \Delta t)^2$
- Symplektischer Euler:

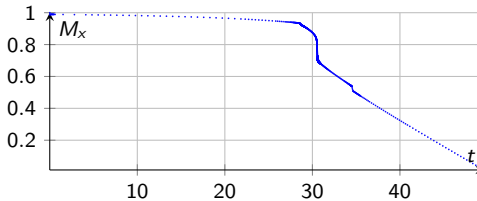
$$\begin{aligned} q_{n+1} &= q_n + h \left(\frac{\partial H(p, q)}{\partial p} \right)_{p=p_n} \\ p_{n+1} &= p_n - h \left(\frac{\partial H(p, q)}{\partial q} \right)_{q=q_{n+1}} \end{aligned}$$

- Diskreter Erhaltungssatz

$$p_n^2 + q_n^2 + \Delta t \, p_n q_n = \text{const}$$

Fehlerschätzer (1)

- Abschätzung des lokalen Fehlers
- Automatische Bestimmung der Schrittweite
- Adaptive Schrittweitensteuerung
- $h-\frac{h}{2}$ -Fehlerschätzer
- eingebettete Runge-Kutta-Verfahren



h - $\frac{h}{2}$ -Fehlerschätzer

Vergleiche Ergebnisse mit verschiedenen Schrittweiten (Richardson extrapolation)

$$u_{i+1}^{(0)} = u_i + \phi(t_i, u_i, h) + c_i h^{P+1} + O(h^{P+2})$$

$$u_{i+\frac{1}{2}}^{(1)} = u_i + \phi(t_i, u_i, \frac{h}{2}) + c_i \frac{h^{P+1}}{2} + O(h^{P+2})$$

$$u_{i+1}^{(1)} = u_{i+\frac{1}{2}} + \phi(t_{i+\frac{1}{2}}, u_{i+\frac{1}{2}}, \frac{h}{2}) + c_{i+\frac{1}{2}} \frac{h^{P+1}}{2} + O(h^{P+2})$$

$$= u_i + \tilde{\phi}(t_i, u_i, h) + \frac{1}{2} c_i h^{P+1} + O(h^{P+2})$$

$$\Rightarrow c_i \propto u_{i+1}^{(1)} - u_{i+1}^{(0)}$$

Eingebettete Runge-Kutta-Verfahren

- Vergleiche Ergebnisse mit verschiedenen Ordnungen (z.B: $p, p - 1$)
- Eingebettete Runge-Kutta-Verfahren
- Vorteil: keine zusätzlichen Funktionsauswertungen notwendig

Allgemein:

c_1	a_{11}	\dots	a_{1s}
\vdots	\vdots		\vdots
c_s	a_{s1}	\dots	a_{ss}
<hr/>			
	b_1	\dots	b_s
	\tilde{b}_1	\dots	\tilde{b}_s

z.B: Euler-Heun-Method (Ordnung 2/1)

0	.	.
1	1	.
<hr/>		
	$\frac{1}{2}$	$\frac{1}{2}$
	1	0

$$u_{i+1}^{(0)} = u_i + \phi(t_i, u_i, h) + c_i h^{P+1} + O(h^{P+2})$$

$$u_{i+1}^{(1)} = u_i + \tilde{\phi}(t_i, u_i, h) + O(h^{P+2})$$

$$\Rightarrow c_i \propto u_{i+1}^{(1)} - u_{i+1}^{(0)}$$

Wahl der Integrationsmethode

- Jedes physikalisches Problem verlangt nach einer passenden Zeitintegrationsmethode
- Mehrschritt Verfahren sind zu wählen, wenn die Berechnung der Funktionswerte aufwendig ist (z.B. CVODE Package)
- Runge-Kutta Methoden eignen sich, wenn Funktionsauswertung mit wenig Aufwand verbunden ist (RK-Suite)
- Falls Speicherplatz ein Problem ist wird man Runge-Kutta Methoden verwenden
- Bei steifen Problemen sind implizite Zeitintegrationsmethoden zu verwenden (z.B. CVODE Package)

Beispiel unterschiedliche Effizienz

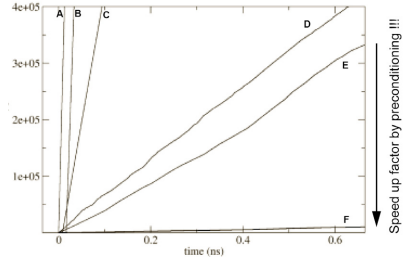
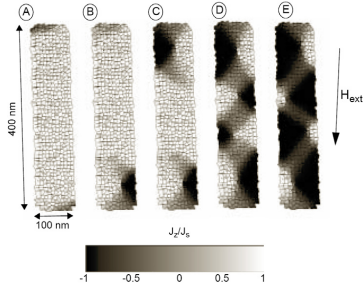


Abbildung 1: (Links) Ummagnetisierung eines magnetischen Nanoteilchens. (Rechts) Zeitgewinn bei verschiedenen Integrationsmethoden: (A) Explizite Runge-Kutta Methode. (B-E) verschiedene implizite Methoden. Im speziellen Backward Differentiation Formula. (F) Neben der ersten Ableitung wird die zweite Ableitung zur Verfügung gestellt (sogenanntes Preconditioning).

CVODE

- hervorragende Implementierungen von verschiedenen Mehrschrittverfahren
- basiert auf **Adams-Moulton** Formeln und **Backward-Difference-Formulas (BDF)**
- automatische Wahl der Anfangsschrittweite
- Fehlerkontrolle wird vorgenommen
- Ordnung und Schrittweite werden automatisch angepasst
- siehe <http://computation.llnl.gov/casc/sundials/>

RK-Suite

- Runge - Kutta Verfahren verschiedener Ordnung
- Fehlerkontrolle durch Vergleich der Rechnung mit Rechnung höherer Ordnung
- automatische Wahl der Anfangsschrittweite
- Siehe z.B: <http://www.netlib.org/ode/rksuite/>

Vorlesung:

- Numerik von Differentialgleichungen (Joachim Schöberl)
- Numerische Methoden und Simulation (K. Held, H. Leeb, C. Lemell, H. Müller)

Literatur:

- E. Hairer, S.P. Norsett und G. Wanner: *Solving Ordinary Differential Equations I: Nonstiff Problems*
- E. Hairer, G. Wanner: *Solving Ordinary Differential Equations II: Stiff and Differential Algebraic Problems*

■ Adams-Bashforth-Methode:

$$y_{n+1} = y_n + hf(t_n, y_n), \quad (\text{This is the Euler method})$$

$$y_{n+2} = y_{n+1} + h \left(\frac{3}{2}f(t_{n+1}, y_{n+1}) - \frac{1}{2}f(t_n, y_n) \right),$$

$$y_{n+3} = y_{n+2} + h \left(\frac{23}{12}f(t_{n+2}, y_{n+2}) - \frac{4}{3}f(t_{n+1}, y_{n+1}) + \frac{5}{12}f(t_n, y_n) \right),$$

$$y_{n+4} = y_{n+3} + h \left(\frac{55}{24}f(t_{n+3}, y_{n+3}) - \frac{59}{24}f(t_{n+2}, y_{n+2}) + \frac{37}{24}f(t_{n+1}, y_{n+1}) - \frac{3}{8}f(t_n, y_n) \right),$$

$$y_{n+5} = y_{n+4} + h \left(\frac{1901}{720}f(t_{n+4}, y_{n+4}) - \frac{1387}{360}f(t_{n+3}, y_{n+3}) + \frac{109}{30}f(t_{n+2}, y_{n+2}) - \frac{637}{360}f(t_{n+1}, y_{n+1}) + \frac{251}{720}f(t_n, y_n) \right).$$

■ Adams-Moulton-Methode:

$$y_n = y_{n-1} + hf(t_n, y_n), \quad (\text{This is the backward Euler method})$$

$$y_{n+1} = y_n + \frac{1}{2}h(f(t_{n+1}, y_{n+1}) + f(t_n, y_n)), \quad (\text{This is the trapezoidal rule})$$

$$y_{n+2} = y_{n+1} + h \left(\frac{5}{12}f(t_{n+2}, y_{n+2}) + \frac{2}{3}f(t_{n+1}, y_{n+1}) - \frac{1}{12}f(t_n, y_n) \right),$$

$$y_{n+3} = y_{n+2} + h \left(\frac{3}{8}f(t_{n+3}, y_{n+3}) + \frac{19}{24}f(t_{n+2}, y_{n+2}) - \frac{5}{24}f(t_{n+1}, y_{n+1}) + \frac{1}{24}f(t_n, y_n) \right),$$

$$y_{n+4} = y_{n+3} + h \left(\frac{251}{720}f(t_{n+4}, y_{n+4}) + \frac{646}{720}f(t_{n+3}, y_{n+3}) - \frac{264}{720}f(t_{n+2}, y_{n+2}) + \frac{106}{720}f(t_{n+1}, y_{n+1}) - \frac{19}{720}f(t_n, y_n) \right).$$

■ Backward-Differentiation-Formula (BDF):

$$\text{BDF1: } y_{n+1} - y_n = hf(t_{n+1}, y_{n+1}); \text{ (this is the backward Euler method)}$$

$$\text{BDF2: } y_{n+2} - \frac{4}{3}y_{n+1} + \frac{1}{3}y_n = \frac{2}{3}hf(t_{n+2}, y_{n+2});$$

$$\text{BDF3: } y_{n+3} - \frac{18}{11}y_{n+2} + \frac{9}{11}y_{n+1} - \frac{2}{11}y_n = \frac{6}{11}hf(t_{n+3}, y_{n+3})$$

$$\text{BDF4: } y_{n+4} - \frac{48}{25}y_{n+3} + \frac{36}{25}y_{n+2} - \frac{16}{25}y_{n+1} + \frac{3}{25}y_n = \frac{12}{25}hf(t_{n+4}, y_{n+4})$$

$$\text{BDF5: } y_{n+5} - \frac{300}{137}y_{n+4} + \frac{300}{137}y_{n+3} - \frac{200}{137}y_{n+2} + \frac{75}{137}y_{n+1} - \frac{12}{137}y_n = \frac{60}{137}hf(t_{n+5}, y_{n+5})$$

$$\text{BDF6: } y_{n+6} - \frac{360}{147}y_{n+5} + \frac{450}{147}y_{n+4} - \frac{400}{147}y_{n+3} + \frac{225}{147}y_{n+2} - \frac{72}{147}y_{n+1} + \frac{10}{147}y_n = \frac{60}{147}hf(t_{n+6}, y_{n+6})$$

1 ODE-Methoden:

Leite eine explizite Einschritt-Methode 2. Ordnung her, die sich von dem Heun-Verfahren unterscheidet.

- Wie sieht das Butcher-Tableau aus?
- Wodurch unterscheiden sich die Methoden?
- Welche Vorteile könnten sich ergeben?

2 Federpendel:

- Stelle die Gleichung für ein ungedämpftes System auf (harmonischer Oszillator)
- Löse die Gleichung anhand der vorgestellten Verfahren
- Stelle die Lösung im Phasenraum dar (v - x -Plot)
- Wie unterscheiden sich die Ergebnisse der einzelnen Methoden?

