

Parareal

Ein paralleler Lösungsalgorithmus für gewöhnliche
Differentialgleichungen

Johannes Reinhardt

Übersicht

Grundlagen

Gewöhnliche Differentialgleichungen

Numerische Methoden

Parareal

Der Algorithmus

Performanceanalyse

Grundlagen

Definition

- ▶ Abkürzung: gDGI
- ▶ Gesucht: $u : \mathbb{R} \rightarrow \mathbb{R}^n, t \in \mathbb{R}$ mit

$$F(t, u(t), u'(t), \dots, u^{(n)}(t)) = 0$$

- ▶ Hier: Explizite gDGI erster Ordnung:

$$u'(t) = F(t, u(t))$$

Beispiele

Physik

- ▶ Bewegungsgleichungen
- ▶ Radioaktiver Zerfall
- ▶ Elektrotechnik

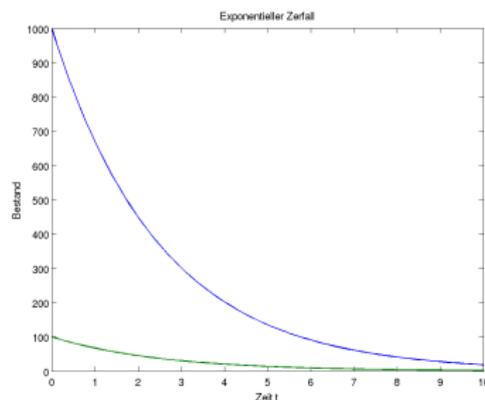
Mathematik

- ▶ Partielle Differentialgleichungen

sonstiges

- ▶ Volkswirtschaftliche Modelle
- ▶ Ökologie

Eindeutigkeit



Radioaktiver Zerfall

$$u'(t) = -\lambda u(t)$$

- ▶ nicht nur eine Lösung
- ▶ weitere Bedingungen an die Lösung
- ▶ von praktischem Interesse:
 - ▶ Anfangswertaufgaben
 - ▶ Randwertaufgaben

Anfangswertprobleme

- ▶ Anfangswert bekannt

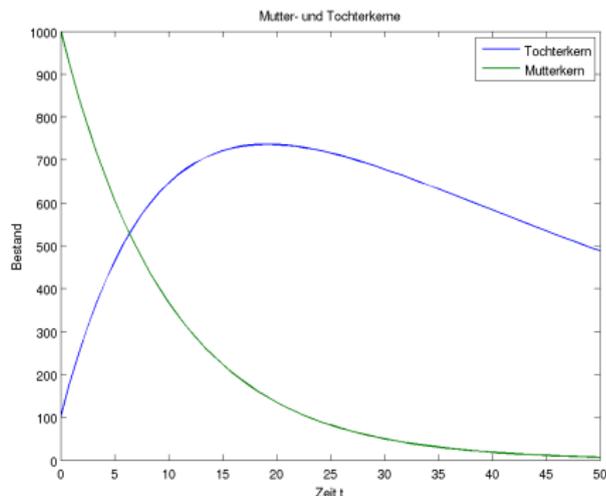
$$u(T_0) = u_0$$

- ▶ Gesucht:

- ▶ $u(T_{end})$
- ▶ u auf Intervall $[T_0, T_{end}]$

Mutter- und Tochterkerne

- ▶ 2 radioaktive Elemente
- ▶ eines zerfällt zu anderem
- ▶ unterschiedliche Halbwertszeiten
- ▶ Wann wieviel von Tochterelement?



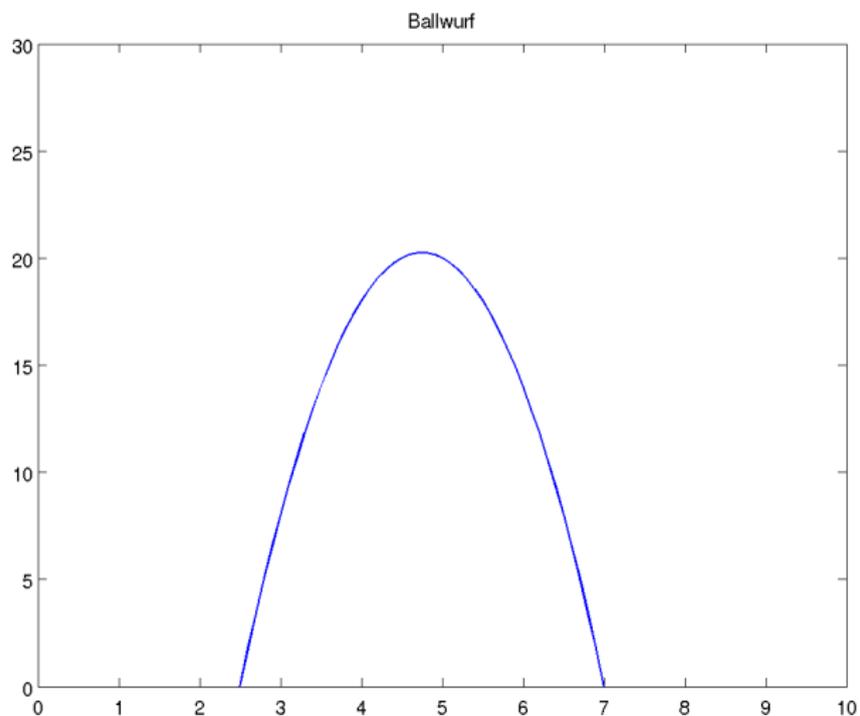
Randwertprobleme

- ▶ Bedingungen an u am Rand
- ▶ Randoperatoren R_1, R_2
- ▶ Gesucht: u auf Intervall $[T_a, T_b]$

$$R_1(u(T_a)) = r_a$$

$$R_2(u(T_b)) = r_b$$

Ballwurf



Ballwurf

- ▶ Komponenten von u :

$$u = (x, y, v_x, v_y)^T$$

- ▶ Randoperator:

$$R : \mathbb{R}^4 \rightarrow \mathbb{R}^2$$

$$u \rightarrow (x, y)$$

- ▶ Randbedingungen:

$$R(u(0)) = (2.5, 0)^T$$

$$R(u(12)) = (7, 0)^T$$

- ▶ Diskretisierung der Zeit:

$$[T_0, T_{end}] \rightarrow \{t_i: i = 0 \dots n, t_0 = T_0, t_n = T_{end}\}$$

- ▶ Meist equidistant, Schrittweite h :

$$t_j - t_{j-1} = h \tag{1}$$

- ▶ Näherungslösung $\hat{u}(t)$ an diskreten Zeitpunkten

Propagator

- ▶ Schreibweise für Lösung von AWP:

$$P : \mathbb{R} \times \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$$

$$P(T_{end}, T_0, u(T_0)) = u(T_{end})$$

- ▶ Propagator für Näherungslösung:

$$\hat{P}(T_{end}, T_0, u(T_0)) = \hat{u}(T_{end})$$

Einschrittverfahren

- ▶ Näherungsverfahren meist Einschrittverfahren
- ▶ Verfahrensfunktion

$$E : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$$

$$E(t_0, u(t_0), h) = \hat{u}(t_0 + h) \approx u(t_0 + h)$$

- ▶ Zugehöriger Propagator

$$\hat{P}_E(T_{end}, T_0, u(T_0)) = \hat{u}(T_{end}) \quad (2)$$

Beispiel: Euler Verfahren

- ▶ Einfachstes Verfahren
- ▶ Taylorentwicklung um t_j , abgebrochen nach linearem Term

$$\begin{aligned} E(t_j, u(t_j), h) &= u(t_j) + hu'(t_j) \\ &= u(t_j) + hF(t_j, u(t_j)) \end{aligned}$$

- ▶ **Aber:** „Niedrige Konvergenzordnung“

Beispiel: Runge-Kutta Verfahren

- ▶ Klasse von Verfahren, Stufe s
- ▶ Koeffizienten a_{ij}, b_i, c_i für $i, j = 1, \dots, s$
- ▶ Berechnung s approx. Steigungen

$$k_i = F \left(t_i + c_i h, u(t_i) + h \sum_{j=1}^s a_{ij} k_j \right)$$

- ▶ Geschickte Kombination der k_j

$$E(t_i, u(t_i), h) = u(t_i) + h \sum_{j=1}^s b_j k_j$$

Butcher Tableaus

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
c/b	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{1}{6}$

Runge-Kutta 4

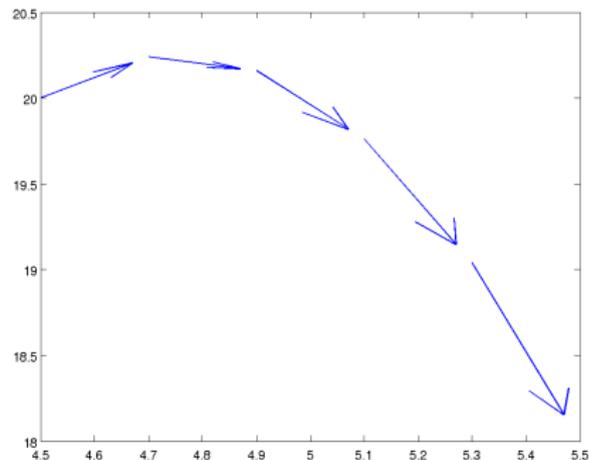
0	
c/b	1

Euler

Anfangswertprobleme

- ▶ Setze $u(t_0) = u_0$
- ▶ Iterativ bis T_{end} :

$$u(t_{i+1}) = E(t_i, u(t_i), h)$$



Randwertprobleme

- ▶ Idee: Umformulierung des RWP zu Gleichungssystem
- ▶ Definiere „Randdifferenz“:

$$D(u_a) := R_2(P(T_b, T_a, u_a)) - r_b$$

- ▶ Gleichungssystem

$$D(u_a) = 0$$

$$R_1(u_a) = r_a$$

- ▶ Aus Lösung $u(T_a)$ kann $u(T_b)$ bestimmt werden

$$u(T_b) = P(T_b, T_a, u(T_a))$$

Schießverfahren

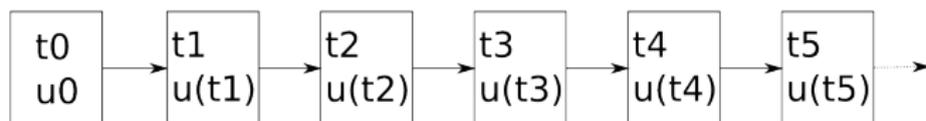
- ▶ Verfahren zur Lösung von Randwertproblemen
- ▶ Numerische Lösung des Gleichungssystems
 - ▶ Newton Verfahren
 - ▶ Bisektionsverfahren
 - ▶ ...
- ▶ Numerische Lösung der AWP's
 - ▶ Euler Verfahren
 - ▶ Runge Kutta Verfahren
 - ▶ ...

Zwischenbilanz

- ▶ Definition
- ▶ Anfangswert- und Randwertaufgaben
- ▶ numerischen Werkzeugkasten
- ▶ Lösung von Anfangswert und Randwertaufgaben
- ▶ Beispiele und Anwendungen

Probleme

- ▶ ODEs teuer und hochdimensional
- ▶ Hohe Genauigkeit nötig
- ▶ Kleine Schrittweiten, teure Verfahren nötig
- ▶ Großer Rechenaufwand, großer Zeitaufwand
- ▶ Echtzeitberechnung
- ▶ **Ausweg:** Parallelisierung
- ▶ **Aber:** Einschrittverfahren konzeptionell seriell



Parareal

Ziel

- ▶ Lösung eines Anfangswertproblems auf mehrere Processing Units (PU)

$$u : [T_0, T_{end}] \rightarrow \mathbb{R}^n$$

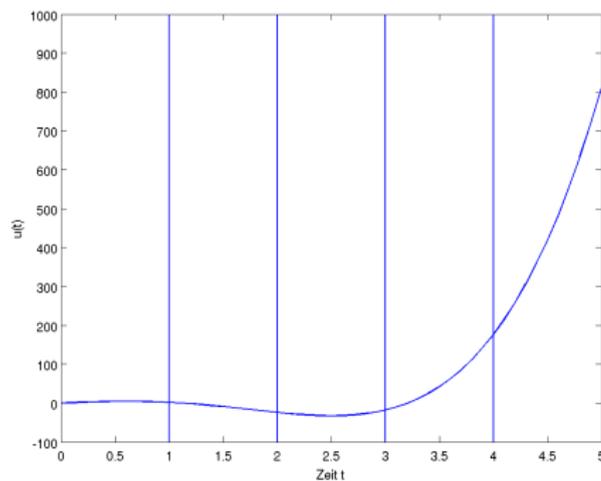
$$u'(t) = F(t, u(t))$$

$$u(T_0) = u_0$$

- ▶ Schneller als auf einzelner PU

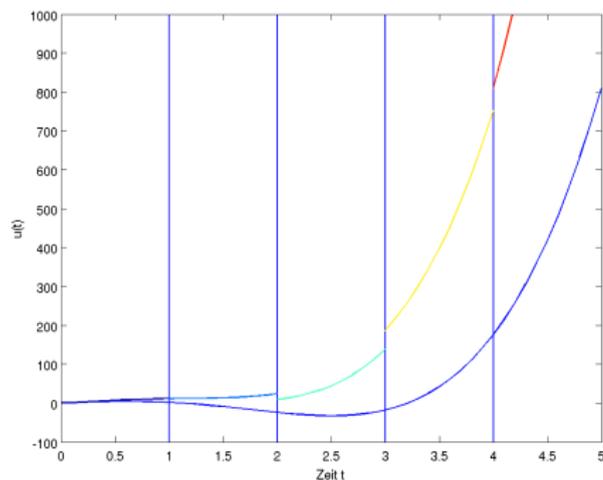
Idee

- ▶ Aufteilung in Teilintervalle



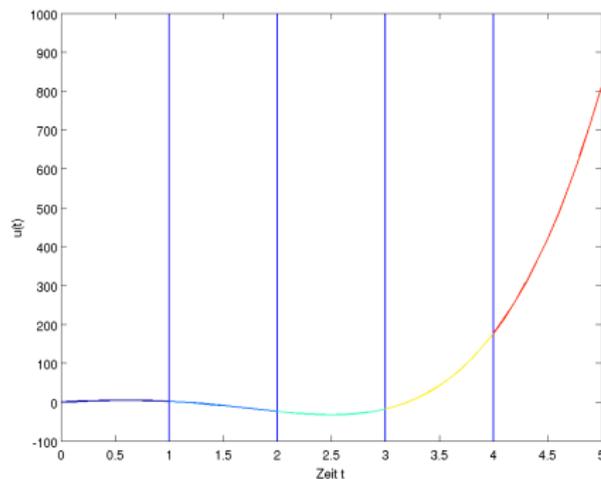
Idee

- ▶ Aufteilung in Teilintervalle
- ▶ AWP auf jedem Teilintervall



Idee

- ▶ Aufteilung in Teilintervalle
- ▶ AWP auf jedem Teilintervall
- ▶ Anschlussbedingungen durch Iteration



Vorgehen bei Herleitung

1. Formulierung des AWP als Gleichungssystem
2. Lösung mit Newtonverfahren

AWP als Gleichungssystem

Anfangswertprobleme

$$\begin{array}{lll}
 v_0'(t) = F(t, v_0(t)), & v_0(T_0) = U_0, & t \in [T_0, T_1] \\
 v_1'(t) = F(t, v_1(t)), & v_1(T_1) = U_1, & t \in [T_1, T_2] \\
 \dots & \dots & \dots \\
 v_N'(t) = F(t, v_N(t)), & v_N(T_N) = U_N, & t \in [T_N, T_{end}]
 \end{array}$$

Anschlussbedingungen

$$\begin{array}{l}
 U_1 = v_0(T_1) = P(T_1, T_0, U_0) \\
 U_2 = v_1(T_2) = P(T_2, T_1, U_1) \\
 \dots \\
 U_{end} = v_N(T_{end}) = P(T_{end}, T_N, U_N)
 \end{array}$$

- ▶ Notation:

$$\mathbf{U} = \begin{pmatrix} U_0 \\ U_1 \\ \dots \\ U_{end} \end{pmatrix} \qquad \mathbf{U}_- = \begin{pmatrix} u_0 \\ v_0(T_1) \\ \dots \\ v_N(T_{end}) \end{pmatrix}$$

- ▶ Anschlussbedingungen bilden Gleichungssystem

$$g : \mathbb{R}^{N \times n} \rightarrow \mathbb{R}^{N \times n}$$

$$g(\mathbf{U}) = \mathbf{U} - \mathbf{U}_-$$

Äquivalente Formulierung des AWP

- ▶ $g(\mathbf{U}) = 0 \Leftrightarrow U_i = u(T_i)$
- ▶ Lösung des AWP \Leftrightarrow Lösung eines Gleichungssystem
- ▶ Newton Verfahren

$$\mathbf{x}^{k+1} = \mathbf{x}^k - J_g^{-1}(\mathbf{x}^k)g(\mathbf{x}^k)$$

J_g Jacobimatrix

Jacobi Matrix

- ▶ Newton für Anschlussbedingungen:

$$\mathbf{U}^{k+1} = \mathbf{U}^k - J^{-1}(\mathbf{U}^k)(\mathbf{U}^k - \mathbf{U}_-^k)$$

- ▶ mit J durchmultiplizieren

$$J(\mathbf{U}^k)(\mathbf{U}^{k+1} - \mathbf{U}^k) = \mathbf{U}^k - \mathbf{U}_-^k$$

Jacobi Matrix

► Jacobi Matrix

$$J = \begin{pmatrix} I & 0 & 0 & \dots \\ -\frac{\partial v_0(T_1)}{\partial U_0} \Big|_{U_0^k} & I & 0 & \\ 0 & -\frac{\partial v_1(T_2)}{\partial U_1} \Big|_{U_1^k} & I & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{pmatrix}$$

Parareal

- ▶ Einsetzen, auflösen liefert

$$U_{i+1}^{k+1} = v_i(T_{i+1}, U_i^k) + \left. \frac{\partial v_i(T_{i+1})}{\partial U_i} \right|_{U_n^k} (U_i^{k+1} - U_i^k)$$

- ▶ Update erfordert Lösung von Anfangswertaufgaben
- ▶ Approximation durch unterschiedlich teure Verfahren
- ▶ Dadurch Parallelisierung sinnvoll

- ▶ Teures, feines Verfahren für

$$u_n(T_{i+1}, U_i^k) \approx P_F(T_{i+1}, T_i, U_i^k)$$

- ▶ Schnelles, grobes Verfahren für

$$\begin{aligned} \frac{\partial v_i(T_{i+1})}{\partial U_i} \Big|_{U_n^k} (U_i^{k+1} - U_i^k) &\approx P_G(T_{i+1}, T_i, U_i^{k+1}) \\ &\quad - P_G(T_{i+1}, T_i, U_i^k) \end{aligned}$$

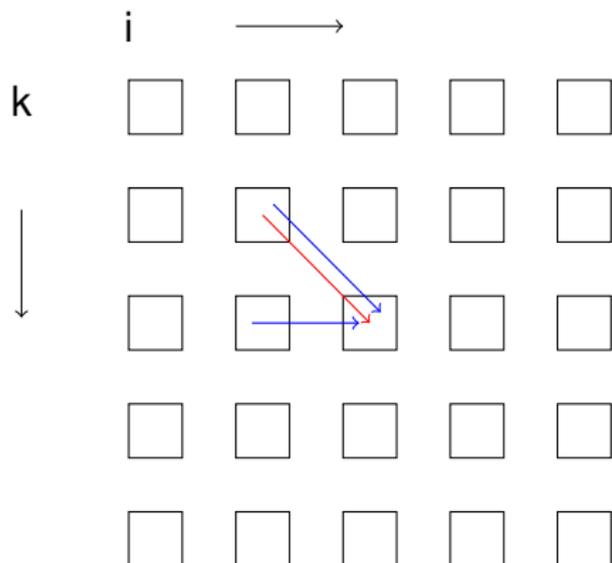
Parareal

- ▶ Beginne mit Approximation für U_i^0
- ▶ Iterative Verfeinerung

$$U_0^{k+1} = u_0$$

$$U_{i+1}^{k+1} = P_G(T_{i+1}, T_i, U_i^{k+1}) + P_F(T_{i+1}, T_i, U_i^k) \\ + P_G(T_{i+1}, T_i, U_i^k)$$

Parallelisierung

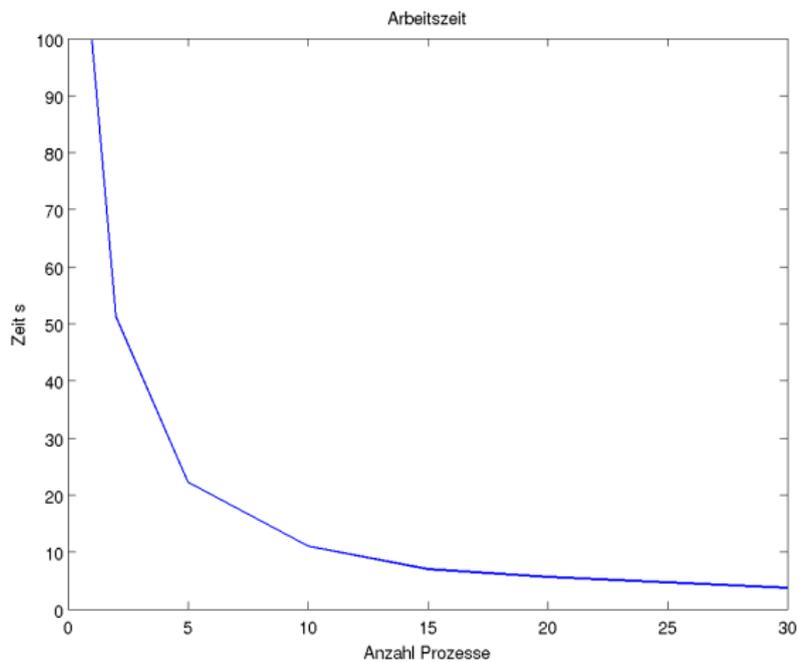


- ▶ Nur noch eine grobe Lösung seriell
- ▶ Rest kann parallel berechnet werden

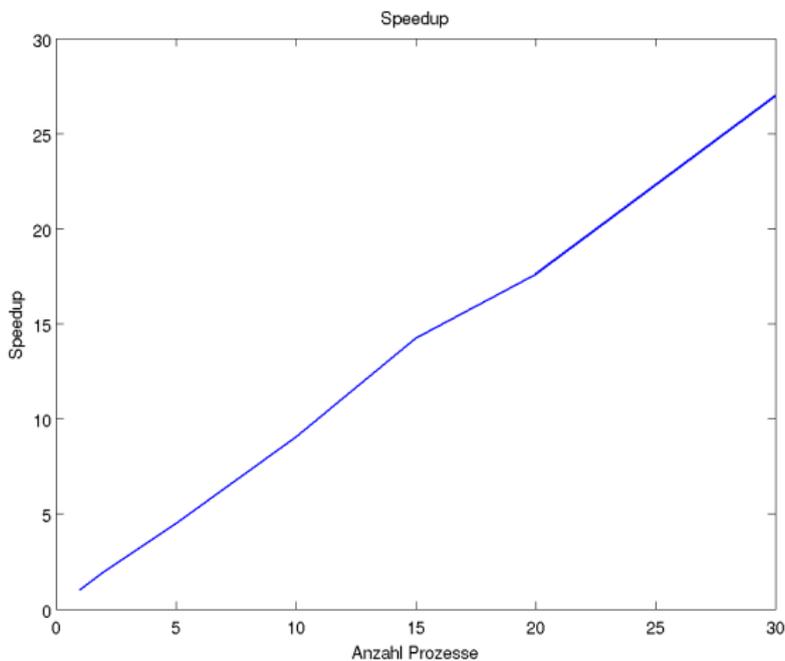
Messmethodik

- ▶ Testproblem: Wärmeleitgleichung aus Übung mit 100 Gitterpunkten in jede Richtung
- ▶ Schrittweite: 0.01
- ▶ Intervall: $[0, 0.02]$
- ▶ Iterationen: 4
- ▶ Zeit gemessen mit `MPI_Wtime`
- ▶ **Achtung:** Nicht aussagekräftig, massiv beschummelt

Zeit



Speedup



Effizienz

