

Zeitliche Dynamik Fortsetzung

Zeitabhängige Probleme können für unsere Beispiele in zwei Fälle klassifiziert werden (eigentlich Spezialfälle)

Lineare Probleme:

1) $\underline{\dot{u}} = \underline{A} \cdot \underline{u}$ (indem $\underline{u} = \begin{pmatrix} \underline{v} \\ \underline{p} \end{pmatrix}$,
 \uparrow \uparrow \uparrow
 Mat. Vektor \uparrow \uparrow
 können auch Problem

2) mit zeitabhängiger Matrix $\underline{A}(t)$ $\underline{v} = \underline{B} \cdot \underline{v} + \underline{f}$
 $\underline{\dot{u}} = \underline{A}(t) \cdot \underline{u}$ dabei kann es effizient sein
 in Form z.B. Beispiel:

$$\underline{A}(t) = \underline{A}_{\text{const}} + \underline{F}(t) \underline{A}_{\text{dyn}}$$

3) $\underline{u} = \underline{f}(\underline{u}, t)$ Allgemein Form, dadurch
 schwieriger in vektorisierter Fassung
 zu bringen.

Beispiel für die Umsetzung

1. Beispiel zeitabhängige Schrödingergl

$$i\hbar \partial_t \psi(\underline{r}, t) = \left(-\frac{\hbar^2}{2m} \Delta_{\underline{r}} + V_{\text{core}}(\underline{r}) \right) \psi(\underline{r}, t) + \delta(\underline{r}) dE(t)$$

Wir diskretisieren

$$\psi(\underline{r}, t) \text{ z. B. auf } \delta \text{ it/r} \rightarrow \underline{\psi}$$

$$\left(-\frac{\hbar^2}{2m} \Delta_{\underline{r}} + V_{\text{core}}(\underline{r}) \right) \rightarrow \underline{A}_{\text{const}}$$

$$\underline{v} = \begin{pmatrix} \underline{\psi} \\ \lambda \end{pmatrix}$$

$$\partial_t \underline{v} = \begin{pmatrix} \underline{A}_{\text{const}} & 0 \\ 0 & 0 \end{pmatrix} \cdot \underline{v} + \underline{F}(t) \cdot \begin{pmatrix} 0 & 0 \\ 0 & \underline{\delta} \end{pmatrix} \cdot \underline{v}$$

\Rightarrow Fall 2. (sehr leicht)
 bzw. Fall 1 falls $F(t) = 0$

2. Beispiel Blochgleichung

$$\partial_t \rho_{12} = -i(\epsilon_1 - \epsilon_2) \rho_{12} - i F d_{21} (\rho_{22} - \rho_{11})$$

$$\partial_t \rho_{11} / \rho_{22} = \mp \text{Im}(F \cdot d_{21} \rho_{12})$$

$$\underline{v} = (\rho_{12}, \rho_{21}, \rho_{11}, \rho_{22})$$

(komplex konjugierte Größen wie
 wie so leicht in Vektorform)

$$\partial_t \underline{v} = \begin{pmatrix} -i(\epsilon_1 - \epsilon_2) & -i F d_{21} & i F d_{21} & 0 \\ -i F d_{21} & +i(\epsilon_1 - \epsilon_2) & -i F d_{21} & 0 \\ -i F d_{21} & +i F d_{21} & 0 & 0 \\ +i F d_{21} & -i F d_{21} & 0 & 0 \end{pmatrix}$$

$$\Rightarrow \partial_t \underline{v} = \underline{A} \cdot \underline{v} \Rightarrow \text{wie Fall 1}$$

Achtung hier gibt es ein Problem $\epsilon_2 - \epsilon_1 \gg F d_{21}$
 daher wird der Zeitschritt sehr klein
 benötigt, um die Oszillationen aufzulösen!

Trick: $\underline{v} \rightarrow \underline{\tilde{v}} = (\tilde{\rho}_{12}, \tilde{\rho}_{21}, \rho_{11}, \rho_{22})$

Mit $\tilde{\rho}_{12} = \rho_{12} e^{i\omega t}$, Rotationsframe

$$\Rightarrow \partial_t \tilde{\rho}_{12} = -i \Delta \tilde{\rho}_{12} - i \hat{F} d_{21} (\rho_{22} - \rho_{11})$$

mit $\Delta = (\epsilon_1 - \epsilon_2) + \omega$

Aber die vektorisierte Form des Problems allerdings nicht.

3. Beispiel Halbleiter-Blockgleichung

Bewegungsgl.:

$$\partial_t P_{11} = -\frac{i}{\hbar} \underbrace{(\Sigma_{V,11} - \Sigma_{C,11} + i\gamma)}_{\text{linear}} P_{11} + \frac{i}{\hbar} \left(\sum_{\xi \neq 0} V(\xi) \right) (f_{C,1\xi} - f_{V,1\xi}) P_{\xi}$$

\Rightarrow aber nicht $A \cdot v$ darstellen
 zwei Stück in v
 Produkt an
 \Rightarrow allgemeine Form Fall 3, gesucht $\delta(v, t)$

\Rightarrow per Hand programmieren

z.B.
$$\sum_{\xi \neq 0} V(\xi) (f_{C,1\xi} - f_{V,1\xi})$$

Faktor jeder

Schwierig bei Parallelisierung, nicht alle Fluss sind auf dem richtigen Prozessor.

Nächster Schritt: Algorithmen finden für die 3. Falle.

Bemerkung: Bei PDGL gibt es noch weitere Vorgehens-

Lösungsverfahren.

Es gibt viele Verfahren, hier diskutieren wir die beliebteste Familie der Runge-Kutta Algorithmen (Achtung: Hier, nicht der spezielle RK4, Order.)

Allgemein ist unser Problem

$$\underline{u} = f(\underline{u}, t)$$

Wir diskretisieren die Zeit!

$$a = t_0 < t_1 < \dots < t_n = b$$

$\underline{u}(t_0)$ sei gegeben.

Speziell sind jetzt Runge-Kutta-Formeln der Form

$$\underline{u}_{n+1} = \underline{u}_n + h \sum_{i=1}^s \textcircled{b_i} k_i$$

ist die Schrittweite
 $h = t_{n+1} - t_n$

mit $\underline{u}(t_n) = \underline{u}_n$

$$k_1 = f(t_n, \underline{u}_n)$$

$$k_i = f(t_n + \textcircled{c_i} h, \underline{u}_n + h \sum_{j=1}^{i-1} \textcircled{a_{i,j}} k_j)$$

$i = 2, \dots, s$

Das Schema verwendet die Ableitung an s verschiedenen Punkten für die weitere Approximation
 $(n_i = \{1, 1, 2, 4, 9, 20, \dots\})$

Ausdrückung des Fehlers

$$\underline{u}(t_n + h) - \underline{u}_{n+1} = \sum_{i=1}^{\infty} h^i \sum_{j=1}^{n_i} \alpha_j^{(i)} D_j^{(i)}$$

Taylor Entwicklung für glatte Fkt.
 Koeffizienten, die hängen von $a_{i,j}$ und b_j
 sind von pot. h. Abh.

Die Ordnung kann jetzt abgelesen werden über die Koeffizienten.

Sind alle $\alpha_j^{(i)} = 0$ für $i \leq k$, dann ist es ein Schema k -Ordnung.

Ein Verfahren wird über die Koeffizienten

c_i, b_i und a_{ij} gesetzt.

Für jede Ordnung gibt es verschiedene Koeffizientenkombinationen, die andere Vor- und Nachteile bzgl. Anzahl an Rechenschritten, Stabilität, Schrittweite, etc. haben.

Beispiele Bogacki-Shampine (Appl. Math. Lett. 2(4), 321-325 (1988))

$$\begin{array}{c|ccc} c_1 & 0 & & \\ c_2 & \frac{1}{2} & & \\ c_3 & \frac{3}{4} & & \\ \hline & 1 & \frac{2}{3} & \frac{4}{9} \\ & & b_1 & \dots & b_3 \end{array} = a_{21}$$

Verfahren 3. Ordnung

Verfahren 3. Ordnung haben ein lokales Fehler von h^3 .
Aufsummiert ist es h^2 (1. Ordnung Verfahren Fehler mit h ,
sehr viel schlechter \Rightarrow hohe Diskretisierung.)

Vorteil von Bogacki-Shampine es reicht auch Verfahren 2. Ordnung aus!

Das erlaubt den Fehler abzuschätzen (steht ja anders nicht (Potenz)) und Schrittweite h bei Rechnen anzupassen!
(embedded Runge Kutta)

Der Fehler ist dann:

$$e_{n+1} = \underbrace{u_{n+1}^{(Method 1)}}_{(Method 1)} - \underbrace{u_{n+1}^{(Method 2)}}_{(Method 2)} = h \sum_{i=1}^5 (b_i^{(Method 1)} - b_i^{(Method 2)}) k_i$$

Das ist $O(h^p)$
Der Fehler muss dann in bestimmter Toleranz sein.
Falls nicht wird die Schrittweite verringert.

Beispiele

Fuhr

$$\frac{0}{1} \bigg| \frac{0}{1}$$

Ham Method

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Original Range Kette

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Embedder Methoden

$$\begin{array}{c|c} 0 & 1 \\ 1 & \frac{1}{2} \quad \frac{1}{2} \\ & 1 \quad 0 \end{array}$$

Ham und Fuhr

Fehlberg RK 1(2)

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ 1 & \frac{1}{256} & \frac{255}{256} \\ \hline & \frac{1}{256} & \frac{255}{256} \quad 0 \\ & \frac{1}{512} & \frac{255}{256} \quad \frac{1}{512} \end{array}$$