

Wiederholung (Finite Differenzen)

$$\left. \frac{d^m f}{dx^n} \right|_{x=x_0} \approx \sum_{\nu=0}^n \delta_{n,\nu}^m f(x_\nu)$$

$n \leftarrow$ grad der Approximation

Fortsetzung
auf dem Gitter

Benachbarte Punkte auf dem Gitter

Wie bestimmt man $\delta_{n,\nu}^m$ Koeffizienten.

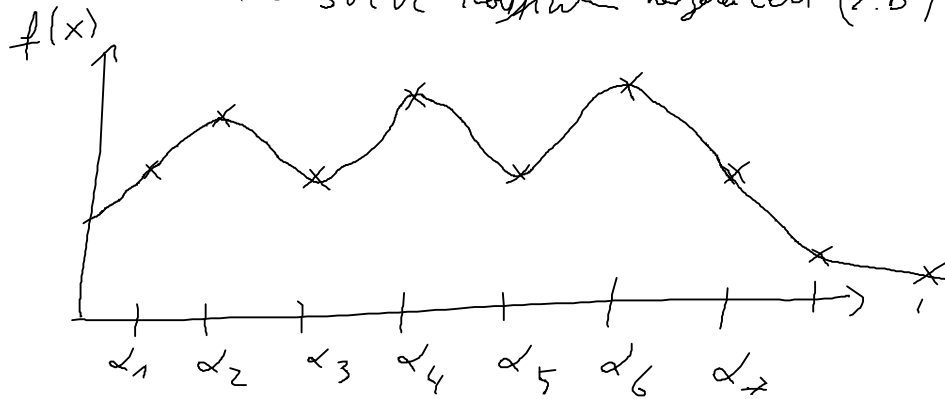
a) Tabelliert für verschiedene Situationen z.B. symmetrischer Stencil

Grad der Ableitung	Order	x-Koordinat									
		-4	-3	-2	-1	0	1	2	3	4	
0	∞					1					
1	2				$-\frac{1}{2}$	0	$+\frac{1}{2}$				
	4			$\frac{1}{12}$	$-\frac{2}{3}$	0	$\frac{2}{3}$	$-\frac{1}{12}$			
2	2				1	-2	1				
	4		$-\frac{1}{12}$	$\frac{4}{3}$	$-\frac{5}{2}$	$\frac{4}{3}$	$-\frac{1}{12}$				
8	2	$\frac{1}{280}$	$-\frac{4}{105}$	$\frac{1}{5}$	$-\frac{4}{5}$	0	$\frac{4}{5}$	$-\frac{1}{5}$	$\frac{4}{105}$	$-\frac{1}{280}$	
	4										$\frac{1}{4}$

usw

b) Es gibt auch iterative Formel zum berechnen der Koeffizienten (s. Formung...)

Wie werden solche Koeffizienten hergeleitet (z.B.)?



Idee: Finde ein Polynom, das die Punkte interpoliert?

Lagrange interpolations Polynom (Wronski (1779), Euler (1783)
Lagrange (1795))

$$p(x) = \sum_{\nu=0}^n F_{n,\nu}(x) f(\alpha_\nu)$$

$$F_{n,\nu}(x) := \frac{w_n(x)}{w_n'(\alpha_\nu) (x - \alpha_\nu)}$$

$$\text{mit } w_n(x) := \prod_{k=0}^n (x - \alpha_k)$$

Das $F_{n,\nu}(x)$ ist ein Polynom der Ordnung n , das 1 bei $x = \alpha_\nu$ ist und 0 bei $x = \alpha_k$ ($k \neq \nu$).

$$\text{Zur Erinnerung } \frac{d^m f}{dx^m} \Big|_{x=x_0} \approx \sum_{\nu=0}^n \delta_{n,\nu}^m f(\alpha_\nu)$$

Die entsprechenden Gewichte ergeben sich, dann über

$$\delta_{n,\nu}^m = \left[\frac{d^m}{dx^m} F_{n,\nu}(x) \right]_{x=0}$$

Zurück zum Problem

Laplace

Der Term war

$$-\frac{\hbar^2}{2m_V} \Delta_V = -\frac{\hbar^2}{2m_V} (\partial_{x_V}^2 + \partial_{y_V}^2)$$

Also $\partial_{x_V}^2 \psi(x[n_1], y[n_2])$

$$= \sum_{\nu=0}^n \delta_{n,\nu}^2 \psi(x[n_1 + \nu - \frac{n}{2}], y[n_2])$$

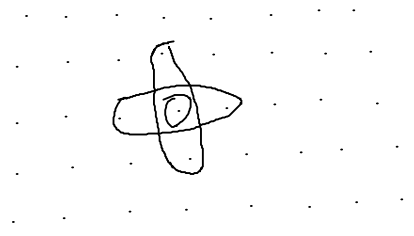
bzw.

$$\partial_{y_V}^2 \psi(x[n_1], y[n_2])$$

$$\rightarrow \sum_{\nu=0}^n \delta_{n,\nu}^2 \psi(x[n_1], y[n_2 + \nu - \frac{n}{2}])$$

Gibt es eine Konsequenz aus dem Zugriffverhalten, für die Vektoren?

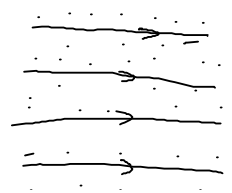
Gitter



Naive Bredy des Vektorindies

$$i[n_1, n_2] = n_1 + n_2 \cdot M$$

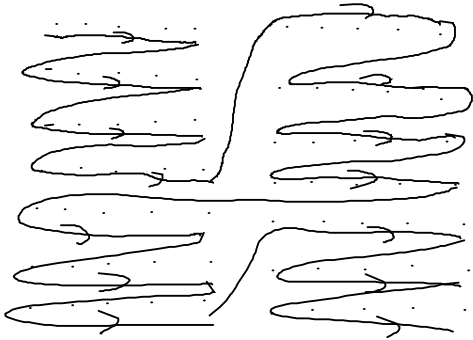
↑
Anzahl der Punkte



Weit entfernte all diagonale
 Anteile (in Speicher nicht in Cache)
 Großen Transfer bei MPI
 Lösung anderer Indizes

3x3 Spalten

$$\begin{array}{c}
 1 \\
 2 \\
 3 \\
 4 \\
 5 \\
 6 \\
 7 \\
 8 \\
 9
 \end{array}
 \begin{pmatrix}
 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
 -4 & 1 & & & & & & & \\
 1 & -4 & 1 & & & & & & \\
 & 1 & -4 & & & & & & \\
 1 & & & -4 & 1 & & 1 & & \\
 & 1 & & 1 & -4 & 1 & & 1 & \\
 & & & & 1 & -4 & & & 1 \\
 & & & & & 1 & -4 & & \\
 & & & & & & 1 & -4 & 1 \\
 & & & & & & & 1 & -4
 \end{pmatrix}$$



⇒ Vorteil Die meisten Zugriffe erfolgen auf Vektorpositionen, die sehr nah beieinander sind. Weniger Zugriffe sind weit entfernt im Speicher.

Nachteil: Programmierung ist aufwendig

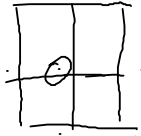
Wichtig: Vorteil, auf verschiedene CPUs oder Kernen entlang der Blockgrenzen, sonst wird der Vorteil nicht ausgespielt.

Es fehlt der Term (Trigon Problem)
 $\frac{h^2}{2m\omega} D_{x_1} \cdot D_{x_2}$ bei Trigon.

Vorgeh
 $D_{x_1} \cdot D_{x_2} = \partial_{x_1} \partial_{x_2} + \partial_{y_1} \partial_{y_2}$

$$\begin{aligned}
 & \partial_{x_1} \partial_{x_2} \psi(x_1[n_1], y_1[n_2], x_2[n_3], y_2[n_4]) \\
 & \approx \sum_{\nu_1=0}^n \delta_{n_1, \nu_1 - \frac{h}{2}} \psi(x_1[n_1 + \nu_1 - \frac{h}{2}], y_1[n_2], x_2[n_3], y_2[n_4]) \\
 & \approx \sum_{\nu_1=0}^n \sum_{\nu_2=0}^n \delta_{n_1, \nu_1} \delta_{n_2, \nu_2} \psi(x_1[n_1 + \nu_1 - \frac{h}{2}], y_1[n_2], x_2[n_3 + \nu_2 - \frac{h}{2}], y_2[n_4])
 \end{aligned}$$

Form des Stencil



Damit ist alles klar für Exziton und Trian, aber ich kann es jetzt Matrixform approximieren:

$$\underline{H} \cdot \underline{v} = \underline{E} \underline{v}$$

\uparrow
 Matrix

II.5 Iterative Lösungsverfahren für das Eigenwertproblem

Ziel ist Grundlagen für iterativer Lösung für:

$$\underline{H} \cdot \underline{v} = \underline{E} \underline{v}$$

wobei H z. B. aus einer partiellen DGL kommt.

Das ist ein Überblick mit Beispielen über wichtige Methoden (unvollständig).

Literatur: z. B. Technical Report 2. Sklep
<http://sklep.upv.es>

Erstmal Grundlagen der numerischen linearen Algebra.

QR-Faktorisierung und Gram-Schmidt Verfahren

$$\underline{X} = \underline{Q} \cdot \underline{R} \quad \begin{pmatrix} r_{11} & \dots & r_{1n} \\ & r_{22} & \dots \\ & & \dots \\ 0 & & & r_{nn} \end{pmatrix}$$

Spalte orthogonale Matrix obere Dreiecksmatrix

Dreiecksmatrizen haben gewisse Vorteile:

- Determinante ist das Produkt der Diagonalelemente
- $\Rightarrow (\underline{R} - \lambda \underline{I}) = 0$ Eigenwerte stehen auf der Diagonalen

Wie bekommt man \underline{Q} & \underline{R} meist (3 typische Verfahren):

- Householder Reflektion
- Givens Drehungen
- Gauß-Schmidt Verfahren (in der einfachen Form numerisch instabil)

Hier Diskutiere Gauß-Schmidt Verfahren
+ Modifikation für bessere Stabilität!

Zur Erinnerung, wie funktioniert Gauß-Schmidt?

Wir haben N Vektoren $\underline{x}_1, \dots, \underline{x}_N$, deren Wollen wir orthonormieren.

Wir erhalten die orthogonalen Vektoren $\underline{q}_1, \dots, \underline{q}_N$ mit der Norm $r_{jj} = \|\underline{q}_j\|_2$ mit

$$\underline{q}_j = \underline{x}_j - \sum_{i=1}^{j-1} \frac{\underline{q}_i^T \underline{x}_j}{r_{ii}} \underline{q}_i$$

Anteil von \underline{x}_j in Richtung von $\text{span}\{\underline{q}_1, \dots, \underline{q}_{j-1}\}$

In Form des Algorithmus (nach Strang Repeat)

$$\underline{q}_1 = \underline{x}_1, \quad \underline{q}_j = \underline{x}_j$$

for $i=1, \dots, j-1$

$$r_{ij} = q_i^T \cdot x_j$$

$$q_j = q_j - r_{ij} q_i$$

$$\text{and } r_{jj} = \|q_j\|^2$$

Man kann auch in Matrix

$$q_j = (I - Q_{j-1} Q_{j-1}^T) q_j$$

Matrix mit
j-1 Spalten
mit q

Projekte auf den
Span $\{q_1, \dots, q_{j-1}\}$

\Rightarrow Was hat das
mit der
QR Faktorisierung
zu tun?

Algorithmus

Eingangs Matrix X

$$r_{11} = \|x_1\|_2$$

$$q_1 = \frac{x_1}{r_{11}}$$

for $j = 2, \dots, n$

Erhalte die Werte $[r_{ij}, r_{jj}]$ aus dem
Gram-Schmidt Verfahren für
 q_1, \dots, q_{j-1} und x_j

$$q_j = q_j / r_{jj}$$

and

Ausgabe die Matrizen Q und R