ulm university universität
uulm

Fakultät für Mathematik und Wirtschafts-wissenschaften

Institut für Numerische Mathematik

Cache-optimierte QR-Zerlegung

Bachelorarbeit an der Universität Ulm

Vorgelegt von:
Florian Krötz
florian.kroetz@uni-ulm.de

Gutachter:
Dr. Michael Lehn
Dr. Andreas Borchert

Betreuer:
Dr. Michael Lehn

2018

Fassung 23. April 2018

Satz: PDF-LaTeX 2ε

Inhaltsverzeichnis

1 Einleitung

Für was brauch ich die QR?

Warum muss die schnell sein?

Was soll die Arbeit?

1.1 Intel MKL

Kapitel über die wichtigkeit der Intel MKL.

1.1.1 QR Anwendung oder so was

-LGS -Ausgleichsprobleme -QR-Verfahren

2 QR factorisation

2.1 QR-Zerlegung

Definition

Eine Matrix $A \in \mathbb{R}^{m \times n}$, $m \geq n$ besitzt eine eindeutige QR-Zerlegung.

$$A = QR \tag{2.1}$$

mit einer orthogonalen Matrix $Q \in \mathbb{R}^{m \times n}$ und einer oberen Dreiecksmatrix $R \in \mathbb{R}^{n \times n}$

Eine QR Zerlegung kann mit einer Householder-Transformation bestimmt werden.

2.2 Householder-Transformation

Sei $v \in \mathbb{R}^n$ und $\tau \in \mathbb{R}$ dann wir die $n \times n$ Matrix

$$H = I - \tau v v^T \tag{2.2}$$

als Householder-Transformation und der Vektor v als Householder-Vektor bezeichnet.

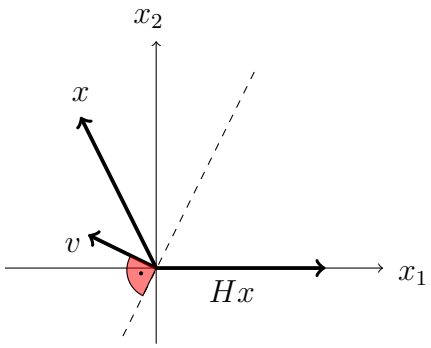

Abbildung 2.1: Spiegelung an der zu v orthogonalen Ebene

2.2.1 Householder Vector

2.2.2 Apply vector

$$H = I - \tau v v' \tag{2.3}$$
$$HA_2 = A_2 - \tau v v' A_2 \tag{2.4}$$
$$= A_2 - \tau v * (v' * A_2) \tag{2.5}$$

2.3 LAPACK QR

Der von LAPACK benutzte Algorithmus [2]

$$H = I - \tau \omega \omega^T \tag{2.6}$$
$$\tau = \frac{\alpha - \beta}{\beta} \tag{2.7}$$
$$\alpha = A(i,i) \tag{2.8}$$
$$\beta = \mathsf{sign}(\alpha) \left| \sqrt{\alpha^2 + \|x\|^2} \right| \tag{2.9}$$
$$x = A(i+1:m,i) \tag{2.10}$$
$$\omega = A(i+1:m,i) * \frac{1}{\alpha - \beta} \tag{2.11}$$

Algorithmus

2.4 NUM1 Urban QR

Algorithmus aus Numerik 1

Mathe

$$H = I - 2\frac{\omega\omega^T}{\omega^T\omega} \tag{2.12}$$

$$\omega_1 = \frac{x - \alpha e_1}{x_1 - \alpha} \tag{2.13}$$

$$\alpha^2 = \|x\|^2 \tag{2.14}$$

2.5 Unterschiede der Algorithmen

LAPCK hat das Tau
Vor und Nachteile oder so was

2.6 QR Blocked

Geblockte Alorighmus

$$H = I - VTV' \tag{2.15}$$

$$H' = I - VT'V' \tag{2.16}$$

$$H'A_2 = A_2 - VT'V'A_2 \tag{2.17}$$

Betrachte A geblockt

$$A = \left(\begin{array}{c|c} A_{0,0} & A_{0,\text{bs}} \\ \hline A_{\text{bs},0} & A_{\text{bs,bs}} \end{array}\right) \tag{2.18}$$

Berechne QR Zerlegung für Blöcke $A_{0,0}$ und $A_{\text{bs},0}$

$$\left(\frac{A_{0,0}}{A_{\text{bs},0}}\right) \leftarrow \left(\frac{Q_{0,0}\backslash R_{0,0}}{Q_{\text{bs},0}}\right) \tag{2.19}$$

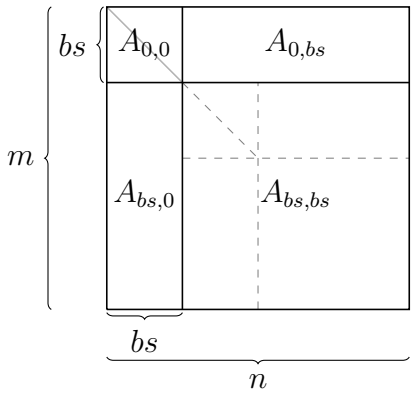

Abbildung 2.2: Partitionierung vom A

Berechne $H(0)...H(bs)$ aus $Q_{0,0}$ und $Q_{bs,0}$ mit $H = I - V * T * V^T$.
Wende H^T auf $A_{0,\text{bs}}$ und $A_{0,\text{bs}}$ an.

$$\left(\frac{A_{0,\text{bs}}}{A_{0,\text{bs}}}\right) \leftarrow H^T \left(\frac{A_{0,\text{bs}}}{A_{0,\text{bs}}}\right) \tag{2.20}$$

Fahre mit $A_{0,\text{bs}}$ fort.

2.6.1 Calc Factor T larft

[1]

$$\begin{aligned} H_2 H_1 x &= (I - \tau_2 v_2 v_2^T)(I - \tau_1 v_1 v_1^T)x \\ &= (I - \tau_1 v_1 v_1^T - \tau_2 v_2 v_2^T - \tau_2 v_2 v_2^T \tau_1 v_1 v_2^T)x \\ &= x - \tau_1 v_1 v_1^T x - \tau_2 v_2 v_2^T x - \tau_1 \tau_2 v_2 (v_2^T v_1) v_2^T x \\ &= x - \tau_1 v_1 v_1^T x - \tau_2 v_2 v_2^T x - \tau_1 \tau_2 (v_2^T v_1) v_2 v_2^T x \end{aligned}$$

$$
\begin{aligned}
H_{1,2}x = (I - VTV^T)x &= x - VTV^T x \\
&= x - (v_1, v_2)\begin{pmatrix} a & b \\ 0 & c \end{pmatrix}\begin{pmatrix} v_1^T \\ v_2^T \end{pmatrix} x \\
&= x - (v_1, v_2)\begin{pmatrix} a & b \\ 0 & c \end{pmatrix}\begin{pmatrix} v_1^T x \\ v_2^T x \end{pmatrix} \\
&= x - (v_1, v_2)\begin{pmatrix} av_1^T x + bv_2^T x \\ cv_2^T x \end{pmatrix} \\
&= x - v_1(av_1^T x + bv_2^T x) - v_2(cv_2^T x) \\
&= x - av_1 v_1^T x - bv_1 v_2^T x - cv_2 v_2^T x
\end{aligned}
$$

2.6.2 Apply H larfb

Die Funktion larfb berechnet.

$$H^T A = A - VT^T V^T A \tag{2.21}$$

$$
\begin{aligned}
W = C' * V &= (C1' * V1 + C2' * V2)(storedinWORK) \\
W &= C1' \\
W &= W * V1 \\
IF(M.GT.K)THEN& \\
W &= W + C2' * V2 \\
W &= W * T' or W * T \\
C &= C - V * W' \\
IF(M.GT.K)THEN& \\
C2 &= C2 - V2 * W' \\
W &= W * V1' \\
C1 &= C1 - W'
\end{aligned}
$$

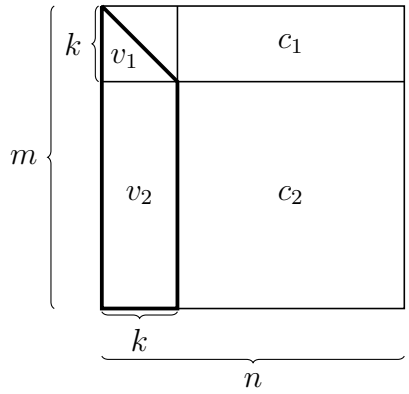

Abbildung 2.3: Partitionierung vom A

2.6.3 Iterativer Algorithmus

for i = 0 : n **do**
    QR = A;
    if i + ib > n **then**
        Calc T: H=I-VTV'
        Apply H: A=H'A
    end if
end for

2.6.4 Rekursiver Algorithmus

e

3 Implementierung und Benchmarks

Irgend was über die HPC Bibliothek

3.1 MKL Wraper

3.2 Benchmarks

A Quelltexte

In diesem Anhang sind einige wichtige Quelltexte aufgeführt.

```
1 #include<stdio.h>
2 int main(int argc, char ** argv) {
3   printf("Hallo HPC \n");
4   return 0;
5 }
```

Literaturverzeichnis

[1] JOFFRAIN, Thierry ; LOW, Tze M. ; QUINTANA-ORTÍ, Enrique S. ; GEIJN, Robert van d. ; ZEE, Field G. V.: Accumulating Householder Transformations, Revisited. In: *ACM Trans. Math. Softw.* 32 (2006), Juni, Nr. 2, 169–179. http://dx.doi.org/10.1145/1141885.1141886. – DOI 10.1145/1141885.1141886. – ISSN 0098–3500

[2] TENNESSEE, Univ. of California B. o. ; LTD.., NAG: *LAPACK unblocked QR*. http://www.netlib.org/lapack/explore-3.1.1-html/dgeqr2.f.html, 2006. – [Online; zugegriffen 31-01-2018]

Name: Florian Krötz Matrikelnummer: 884948

Erklärung

Ich erkläre, dass ich die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den .....

Florian Krötz