



ulm university universität  
**uulm**

**Fakultät für  
Mathematik und  
Wirtschafts-  
wissenschaften**

— Institut im Quellcode  
anpassen nicht verges-  
sen! —

# High Performance implementation of QR factorization

Bachelorarbeit an der Universität Ulm

**Vorgelegt von:**

Florian Krötz  
florian.kroetz@uni-ulm.de

**Gutachter:**

Prof. Dr. Streng Geheim  
Prof. Dr. Un Leserlich

**Betreuer:**

Betreuername

2018

Fassung 4. Februar 2018

© 2018 Florian Krötz

Satz: PDF- $\text{\LaTeX}$  2 <sub>$\varepsilon$</sub>

# Inhaltsverzeichnis

<b>1</b>	<b>QR factorisation</b>	<b>1</b>
1.1	LAPACK . . . . .	1
1.2	NUM3 Urban . . . . .	2
1.3	Zeug . . . . .	3
<b>A</b>	<b>Quelltexte</b>	<b>4</b>
	<b>Literaturverzeichnis</b>	<b>5</b>

# 1 QR factorisation

## 1.1 LAPACK

Mathe [1]

$$H = I - \tau \omega \omega^T \quad (1.1)$$

$$\tau = \frac{\alpha - \beta}{\beta} \quad (1.2)$$

$$\alpha = A(i, i) \quad (1.3)$$

$$\beta = \text{sign}(\alpha) \left| \sqrt{\alpha^2 + \|x\|^2} \right| \quad (1.4)$$

$$x = A(i+1:m, i) \quad (1.5)$$

$$\omega = A(i+1:m, i) * \frac{1}{\alpha - \beta} \quad (1.6)$$

Algorithmus

```
1  householderVektor(Vektor v, alpha, tau)
2    beta = sign(sqrt(alpha ^2 + norm(x)^2), alpha)
3    tau = (alpha - beta) / beta
4    scal(1/(alpha - beta), v)
```

```
1  tau=zeros(min(m,n))
2  for i = 0 : min(m,n)
3    householderVektor(A(i+1:m,i), A(i,i), tau(i))
4    if (i < n && tau != 0)
5      AII = A(i,i)
6      A(i,i)= 1
7      A = A - tau *w(w'*A) // MV und rank1
8      A(i,i) = AII
```

## 1.2 NUM3 Urban

Mathe

$$H = I - 2 \frac{\omega \omega^T}{\omega^T \omega} \quad (1.7)$$

$$\omega_1 = \frac{x - \alpha e_1}{x_1 - \alpha} \quad (1.8)$$

$$\alpha^2 = \|x\|^2 \quad (1.9)$$

Algorithmus

```
1  householderVektor(Vektor x, omega, beta)
2      n = length(x)
3      if n > 1
4          sigma = x(2:end)'*x(2:end);
5          if sigma==0
6              beta = 0;
7          else
8              mu = sqrt(x(1)^2+sigma);
9              if x(1)<=0
10                 tmp = x(1) - mu;
11             else
12                 tmp = -sigma / (x(1) + mu);
13             end
14             beta = 2*tmp^2/(sigma + tmp^2);
15             x(2:end) = x(2:end)/tmp;
16         end
17         v = [1;x(2:end)];
18     else
19         beta = 0;
20         v = 1;
21     end
```

```
1  for i = 1:n
2      housevector(A(i:m, i), w, beta)
3      A(i:m,i:n) = (I(m-i+1) - beta * w * w')*A(i:m,i:n)
4      if i < m
5          A(i + 1 : m, i) = w(2:m-i+1)
```

## 1.3 Zeug

Fortan Col-Major

# 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] 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