

**Fachhochschule Aachen**

**Campus Jülich**

Fachbereich 9: Medizintechnik und Technomathematik

Studiengang: Scientific Programming

Programm-Dokumentation

Berechnung des größten Eigenwertes einer reellen, symmetrischen und dünn besetzten Matrix

WS 2016/2017

Autoren:

Domenic Kühne

Martin Benndorf

Korrektur: Prof. Dr. Karola Merkel (FH Aachen)

Datum: 23. Januar 2017

**Eidesstattliche Erklärung, Kühne**

Hiermit versichere ich, dass ich die Hausarbeit mit dem Thema

„Berechnung des größten Eigenwertes einer reellen, symmetrischen und dünn besetzten Matrix“

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Name: Domenic Kühne

Matrikelnummer: 4012757

Köln, den 23. Januar 2017



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Unterschrift

**Eidesstattliche Erklärung, Benndorf**

Hiermit versichere ich, dass ich die Hausarbeit mit dem Thema

„Berechnung des größten Eigenwertes einer reellen, symmetrischen und dünn besetzten Matrix“

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Name: Martin Benndorf

Matrikelnummer:

Köln, den 23. Januar 2017

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Unterschrift

**Inhaltsverzeichnis**

[1 Aufgabenbeschreibung [KÜHNE] 1](#_Toc472782715)

[2 Ein-/Ausgabe [BENNDORF] 2](#_Toc472782716)

[3 Compressed Row Storage Verfahren [BENNDORF] 3](#_Toc472782717)

[3.1 Mathematische Beschreibung 3](#_Toc472782718)

[3.2 Nassi-Shneiderman-Diagramme 4](#_Toc472782719)

[4 Vektoriteration [KÜHNE] 5](#_Toc472782720)

[4.1 Grundlagen aus der linearen Algebra 5](#_Toc472782721)

[4.2 Mathematischer Beweis 5](#_Toc472782722)

[4.3 Nassi-Shneiderman-Diagramme 5](#_Toc472782723)

[4.3.1 Vektoriteration 5](#_Toc472782724)

[4.3.2 Matrix-Vektor-Multiplikation mit CRS 6](#_Toc472782725)

[4.3.3 Skalarprodukt 7](#_Toc472782726)

[4.3.4 normalisierter Vektor 8](#_Toc472782727)

[5 Testfälle 9](#_Toc472782728)

[Abkürzungsverzeichnis 10](#_Toc472782729)

[Quellenverzeichnis 11](#_Toc472782730)

[Anhang 12](#_Toc472782731)

# Aufgabenbeschreibung [KÜHNE]

Die Aufgabe bestand darin ein Programm in der Programmiersprache *Cobol* zu entwickeln, welches den betragsmäßig größten Eigenwert einer reellen, symmetrischen und dünn besetzten Matrix mittels des numerischen Verfahrens der Vektoriteration bestimmt.

Das Programm sollte auf dünn besetzte Matrizen optimiert werden, indem nur die von Null verschiedenen Matrixeinträge oberhalb der Diagonale gespeichert werden. Dazu sollte ein geeignetes Verfahren ausgewählt werden um die Einträge der Matrix abzuspeichern.

Aus einer Datei sollten die entsprechenden Eingabeparameter gelesen und anschließend validiert werden. Die Eingabeparameter setzten sich zusammen aus der Matrix für die der betragsmäßig größte Eigenwert berechnet werden soll, dem Startvektor x0, der maximalen Anzahl der Iterationen und einer Zahl **ε**, welche in der Abbruchbedingung **|𝜆𝑛+1− 𝜆𝑛| ≤ 𝜀 | 𝜆𝑛**| zum Einsatz kommen sollte. Das Ausgabemedium des Programms sollte eine Datei sein, in welche die Ergebnisse geschrieben werden.

Für die Realisierung des Programms sollte eine geeignete Modularisierung gewählt werden.

# Ein-/Ausgabe [BENNDORF]

mögliche Testfälle:

keine quadratische (n x n)-Matrix

Matrix ist nicht symmetrisch

Matrix ist nicht dünn besetzt

Nullmatrix -> sollte evtl. hier schon abgefangen werden..

n ist negativ

ε größer 1

n und/oder ε nicht gesetzt

Eingabe-Format beschreiben!

## Eingabeformat

Das Eingabeformat in der Eingabedatei ist sehr simpel aufgebaut:

dim=10

n=1000

e=0.001

x=1 -2 3 -4 5 -5 4 -3 2 -1

2 1 0 0 0 0 0 0 0 0

1 2 1 0 0 0 0 0 0 0

0 1 2 1 0 0 0 0 0 0

0 0 1 2 1 0 0 0 0 0

0 0 0 1 2 1 0 0 0 0

0 0 0 0 1 2 1 0 0 0

0 0 0 0 0 1 2 1 0 0

0 0 0 0 0 0 1 2 1 0

0 0 0 0 0 0 0 1 2 1

0 0 0 0 0 0 0 0 1 2

Zuerst werden nacheinander Dimension, der Maximalwert für N und Epsilon eingelesen. Dies geschieht indem man <dim/n/e>=<wert> in eine Zeile schreibt.

Der Startvektor X wird ähnlich eingelesen. Alle Werte des Vektors werden in eine Zeile geschrieben und durch Leerzeichen getrennt.

Alle anderen Zeilen werden zu der Inputmatrix hinzugefügt. Dies funktionert wie beim Startvektor, alle Elemente einer Reihe in eine Zeile, getrennt durch Leerzeichen.

## Ausgabeformat

Das Ausgabeformat ist ähnlich simpel gestaltet:

==========

Matrix:

2.0000 1.0000 0.0000 0.0000

1.0000 2.0000 1.0000 0.0000

0.0000 1.0000 2.0000 1.0000

0.0000 0.0000 1.0000 2.0000

0.0000 0.0000 0.0000 1.0000

Start Vektor:

1.0000 - 2.0000 3.0000 - 4.0000

Erwartungswert:

3.6813

==========

(Matrix und Startvektor sind gekürzt)

Da mehrere Ausgaben in der Ausgabedatei auftreten können (neue Ausgaben werden zur besseren Übersicht einfach am Ende der Datei angehängt) werden die einzelnen Ausgaben durch mit Gleichzeichen gefüllten Reihen getrennt.

Die Ausgabe besteht im Normalfall aus der Startmatrix, dem Startvektor und dem errechnetem Erwartungswert.

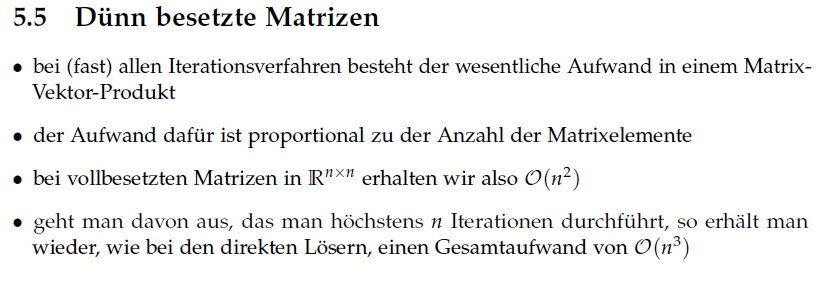
Falls während der Ausführung ein Fehler auftritt (zum Beispiel Syntaktisch Fehlerhafte Eingaben oder die Eingabe einer nicht unterbesetzen Matrix) wird nur dieser in der Ausgabedatei ausgegeben.

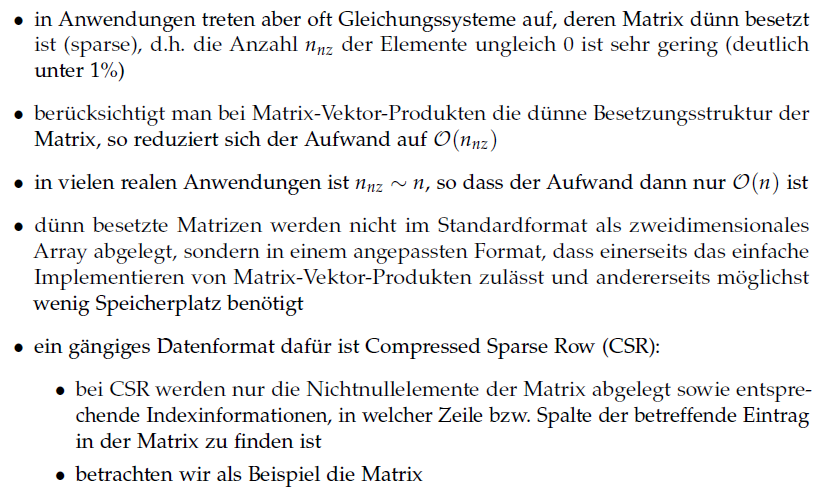
# Compressed Row Storage Verfahren [BENNDORF]

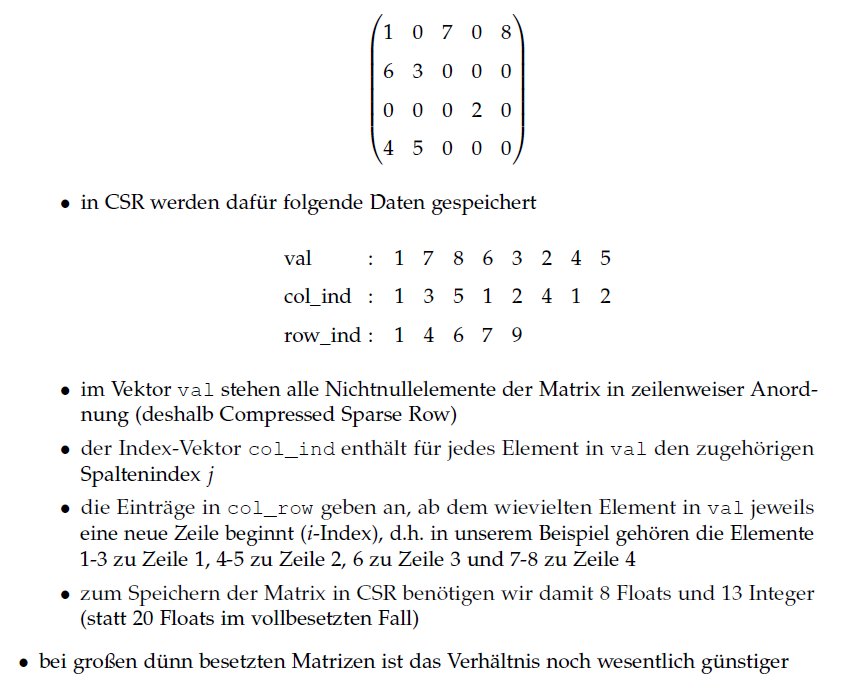
## Mathematische Beschreibung

Hier nur CRS beschreiben, in der Präsentation aber auch Compressed Column Storage und sagen wieso wir uns für CRS entschieden haben

Auszug aus Numerik-Skript:







## Nassi-Shneiderman-Diagramme

# Vektoriteration [KÜHNE]

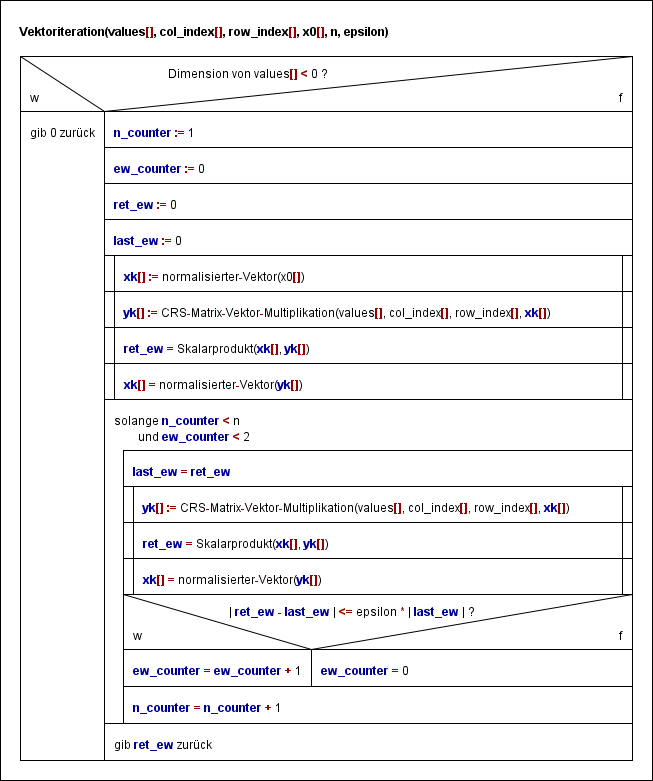
## Grundlagen aus der linearen Algebra

## Mathematischer Beweis

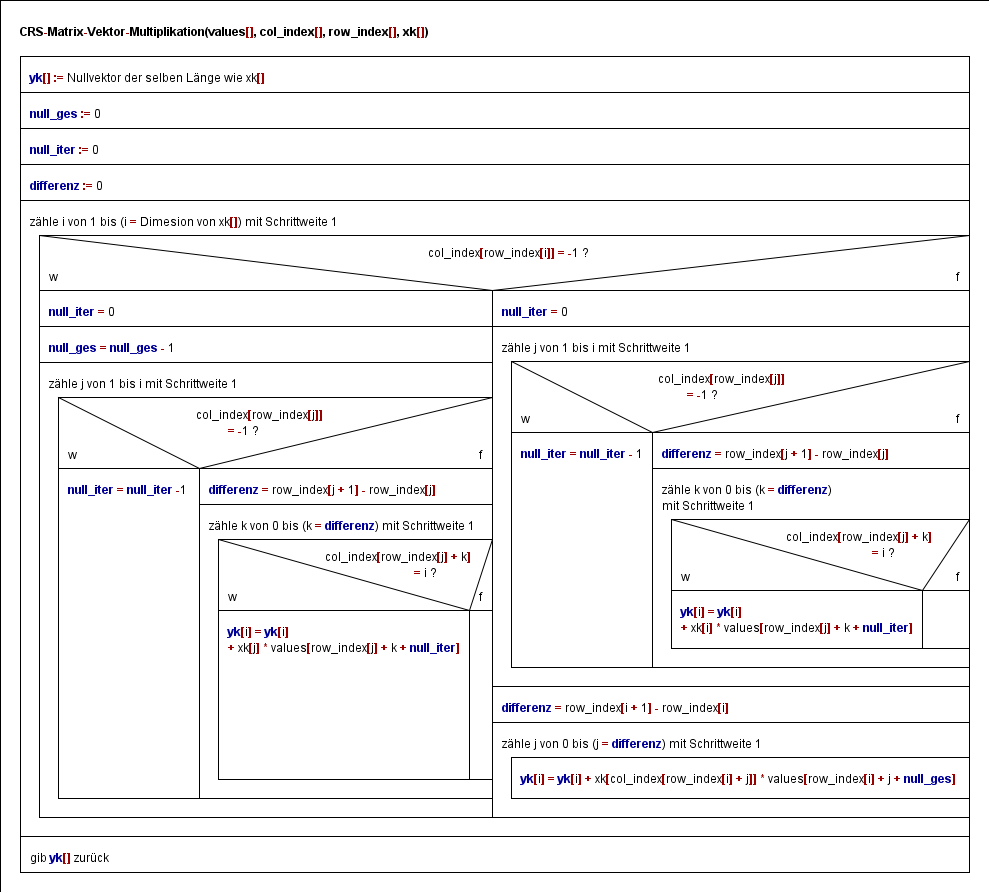
## Nassi-Shneiderman-Diagramme

Dieser Abschnitt beinhaltet den Algorithmus der Vektoriteration in Form von Nassi-Shneiderman-Diagrammen. Dazu werden die benötigten Funktionen der Matrix-Vektor-Multiplikation, des euklidische Skalarpodukt und der Normalisierung eines Vektors ebenfalls in Struktogrammen abgebildet.

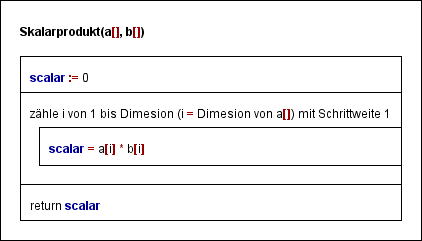
### Vektoriteration



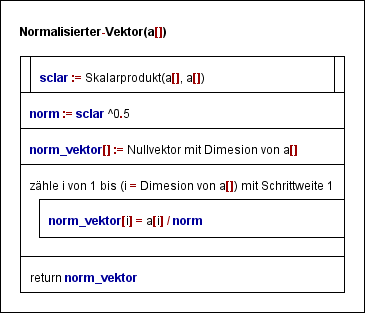
### Matrix-Vektor-Multiplikation mit CRS



### Skalarprodukt



### normalisierter Vektor



# Testfälle

# Abkürzungsverzeichnis

|  |  |
| --- | --- |
| CRS | Compressed Row Storage |
| EVA | Eingabe, Verarbeitung, Ausgabe |
|  |  |
|  |  |
|  |  |
|  |  |

# Quellenverzeichnis

# Anhang