

Probeklausur zu Algorithmen der Numerischen Mathematik SS 2013

ID Nummer: **1**

Name:

Geburtsdatum:

Matrikelnummer:

Übungsgruppenleiter:

Abschluss (D,B,L neu, L alt):

Probeklausur zu Algorithmen der Numerischen Mathematik — 03.07.2013

Aufgabe	1	2	3	4	5	Σ
max. Punkte	4	4	4	4	4	16
err. Punkte						

Hinweise: WICHTIG !

- Schreiben Sie auf jedes Blatt Ihre ID Nummer OHNE NAMEN, schreiben Sie mit Kugelschreiber oder Füller.
- Wichtig: Bearbeiten Sie nur 4 Aufgaben. Sollten Sie dennoch zu allen Aufgaben Lösungen abgeben, wird Aufgabe 2 nicht gewertet.
- Es werden nur Lösungsvorschläge gewertet, bei denen der Lösungsweg klar zu erkennen ist. Die Angabe des Endresultats alleine gibt keine Punkte.
- Berechnete Lösungen sind exakt anzugeben und nicht als gerundete Zahlen.
- Verwenden Sie für jede Aufgabe ein neues Blatt.
- Zum Bestehen dieser Klausur sind **8** Punkte hinreichend.
- Es sind keine elektronischen Geräte als Hilfsmittel zugelassen.

Aufgabe 1:**ID: 1**

Gegeben sei folgendes kleines Matlab-Programm

```
1 function y = kleinesprogramm(x)
2 N=length(x);
3 L=log2(N);
4 if(mod(L,2)~=1 & mod(L,2)~=0)
5     fprintf("Falsche Dimension!");
6 end
7 y(1)=x(1);
8 for k=2:N
9     index=0;
10    zahl=k-1;
11    for j=1:L
12        m=floor(zahl/2);
13        rest= mod(zahl,2);
14        zahl=m;
15        index=index+rest*2^(L-j);
16    end
17    y(k)=x(index+1);
18 end
19 end
```

wobei die Funktion `floor(z)` die Nachkommastellen der Zahl z abschneidet (z.B. `floor(2.14)=2`).

- Welche Aufgabe erfüllt obiges Programm? Welches Ergebnis liefert das Programm angewendet auf den Vektor $x = (2, 0, 3, 1, 6, 4, 5, 7)^T$?
- Erläutern Sie, wie sich die Variable `index` jeweils zwischen den Zeilen 6 und 13 ändert.
- Wie groß ist der Rechenaufwand des Programms für einen Eingabevektor der Länge $N = 2^L$, $L \in \mathbb{N}$?
- An welcher(n) Stelle(n) müssen Sie obiges Programm in nachfolgender Funktion `moechtegern_fft(x)` einfügen, damit dieses tatsächlich die diskrete Fouriertransformation eines $N = 2^L$ -dimensionalen Vektors berechnet?

```
1 function y = moechtegern_fft(x)
2 N=length(x);
3 L=log2(N);
4 A=zeros(N,1);
5 for k=1:L;
6     for j=1:2^(L-k)
7         for l=1:2^(k-1)
8             factor = exp((l-1)*2*pi*1i/(2^k)) * y(2^k*(j-1)+2^(k-1)+1);
9             A(2^k*(j-1)+1) = y(2^k*(j-1)+1) + factor;
10            A(2^k*(j-1)+2^(k-1)+1) = y(2^k*(j-1)+1) - factor;
11        end
12    end
13    y=A;
14 end
15 end
```

Aufgabe 2:**ID: 1**

Gegeben Sei eine Prozedur $\mathbf{y}=\mathbf{fft}(\mathbf{x},N,d)$ welche als erstes Argument einen Vektor $x \in \mathbb{R}^N$ und als zweites Argument dessen Länge übernimmt. Für die k -te Komponente des Ausgabevektors $\mathbf{y} \in \mathbb{R}^N$ gilt dabei

$$y_k = \begin{cases} \sum_{j=0}^{N-1} e^{i\frac{2\pi}{N}jk} x_j & \text{falls } d = 1 \\ \sum_{j=0}^{N-1} e^{-i\frac{2\pi}{N}jk} x_j & \text{falls } d = -1. \end{cases}$$

Sie können davon ausgehen, dass die Prozedur $\mathbf{fft}(\mathbf{x},N,d)$ die ihr gestellte Aufgabe so effizient wie möglich ausführt.

- a) Welche Ausgabe liefert die Eingabe

$$\begin{aligned} \mathbf{x} &= [0 \ 1 \ 5 \ 4]'; \\ \mathbf{y} &= \mathbf{fft}(\mathbf{fft}(\mathbf{x},4,1),4,-1) \end{aligned}$$

für $\mathbf{y}$?

- b) Gegeben Sei nun weiterhin eine glatte, 2π -periodische Funktion $f : [0, 2\pi) \rightarrow \mathbb{R}$.

Erläutern Sie, wie sie obige Prozedur verwenden können, um näherungsweise die zweite Ableitung von f an einem Punkt $x \in [0, 2\pi)$ zu bestimmen ohne explizit eine Ableitung zu berechnen. Geben Sie einen kurzen Pseudo-Code an. Wieviele Operationen als Funktion von N werden insgesamt benötigt? Wie groß müssen Sie N mindestens wählen, um mit diesem Programm für das spezielle $f(x) = \sin(2x)$ die zweite Ableitung für beliebiges $x \in [0, 2\pi)$ bis auf einen Fehler in der Größenordnung der Maschinengenauigkeit zu berechnen?

Aufgabe 3:**ID: 1**

Eine Matlabfunktion `qr_fuer_tridiag(A)` überschreibe die tridiagonale Matrix $A \in \mathbb{R}^{n \times n}$ entsprechend der Vorschrift für QR-Zerlegungen und gebe zusätzlich einen Vektor $d = (\alpha_1, \dots, \alpha_n)^T$ aus.

- a) Schreiben Sie eine Funktion `rq_fuer_tridiag(A,d)`, welche die von `qr_fuer_tridiag(A)` generierten Matrizen als Eingabe übernimmt und effizient das Produkt RQ berechnet, wobei Q und R durch $A = QR$ definiert sind.
- b) Verwenden Sie die Funktionen `qr_fuer_tridiag(A)` und `rq_fuer_tridiag(A,d)`, um nachfolgendes Programm `EW_von_tridiag(H)` zur Berechnung der Eigenwerte einer Tridiagonalmatrix $H \in \mathbb{R}^{n \times n}$ mit rein reellem Spektrum zu vervollständigen.

```
1 function lambda = EW_von_tridiag(H)
2 n=length(H(:,1));
3 for k=n:-1:2
4     hk=H(k,k);
5     hkminuseins=H(k-1,k-1);
6     while abs(H(k,k-1)) > eps(abs(hk)+abs(hkminuseins))
7         d=(hkminuseins-hk)/2;
8         if d==0
9             mu=hk+sqrt(H_1(k-1,k)^2);
10        else
11            mu=hk+d-sign(d)*sqrt(d^2+H(k-1,k)^2);
12        end
13        hk=H(k,k);
14        hkminuseins=H(k-1,k-1);
15    end
16    lambda(k)=hk;
17    H=H(1:k-1, 1:k-1);
18 end
19 lambda(1)=H(1,1);
20 end
```

Aufgabe 4:**ID: 1**

- a) Führen Sie von Hand die ersten zwei Iterationen des QR-Algorithmus mit einfachem Shift aus der Vorlesung für die Matrix

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

durch ohne die Matrizen Q explizit zu berechnen. Was können Sie über die Konvergenz des Algorithmus in diesem speziellen Fall aussagen?

- b) Ersetzen Sie nun den einfachen Shift aus der Vorlesung durch den Wilkinson-Shift aus PP2 d) und wiederholen Sie obige Rechnungen.

Aufgabe 5:**ID: 1**

Bestimmen Sie Vektoren $v_1, v_2, v_3 \in \mathbb{R}^4$ und $w_1, w_2, w_3 \in \mathbb{R}^4$ sowie Skalare $a, b, c, d, e \in \mathbb{R}$ so, dass für

$$T := \begin{pmatrix} a & d & 0 \\ 1 & b & e \\ 0 & 1 & c \end{pmatrix}, A := \begin{pmatrix} 3 & 0 & 1 & 1 \\ 0 & 4 & 1 & 0 \\ 1 & 0 & 5 & 0 \\ 0 & 1 & 0 & 6 \end{pmatrix}, V := \begin{pmatrix} | & | & | \\ v_1 & v_2 & v_3 \\ | & | & | \end{pmatrix} \text{ und } W := \begin{pmatrix} | & | & | \\ w_1 & w_2 & w_3 \\ | & | & | \end{pmatrix}$$

gilt

$$W^T V = D := \text{diag}(w_1^T v_1, w_2^T v_2, w_3^T v_3), \\ W^T A V = D T.$$

Aufgabe 6:**ID: 1** (freiwilliger Zusatz als Verständnistest)

a) Beantworten Sie kurz folgende Verständnisfragen:

- Was hat die Methode des steilsten Abstieges mit Tasmania Berlin zu tun?
- Was ist die Suchrichtung beim Verfahren des steilsten Abstieges?
- Sind beim Verfahren des steilsten Abstieges alle Suchrichtung aufeinander orthogonal?
- Das Verfahren des steilsten Abstieges werde zur Lösung eines n -dimensionalen LGS verwendet. Stimmt es, dass das Verfahren nach höchstens n Schritten die Lösung liefert?
- Ist das cg-Verfahren für beliebige invertierbare Matrizen sinnvoll?
- Bezüglich welches Skalarproduktes sind beim cg-Verfahren die Suchrichtungen orthogonal?
- In welcher Relation stehen die Suchrichtungen beim Verfahren des steilsten Abstieges und die Gradienten beim cg-Verfahren?
- Welchen Aufwand hat ein Schritt des cg-Verfahrens?
- Was ist die Idee des vorkonditionierten cg-Verfahrens?
- Bezüglich welches Skalarproduktes sind die Basisvektoren v_k im Arnoldi-Verfahren orthogonal?
- Wieviele Operationen sind zur Berechnung der Arnoldi-Basis des Krylovraumes $\mathcal{K}_k(A, b)$ für allgemeine Matrizen erforderlich?
- Um welchen Faktor ist die Berechnung der Lanczos-Basis im allgemeinen Fall schneller als die Berechnung der Arnoldi-Basis?
- Was ist der Unterschied zwischen FOM und GMRES?
- BiCG und QMR kommen der Lösung des LGS in jedem Schritt näher

b Verstehen Sie einen Simplex-Schritt am Beispiel der linearen Optimierungsaufgabe

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 12 \\ x \geq 0 \\ x_1 + x_2 + x_3 = \min! \end{cases}$$

Dabei werde $x = (12, 0, 0)^T$ als Anfangsecke gewählt. Überlegen Sie sich dazu:

- Wie ändern sich die Kosten $x_1 + x_2 + x_3$ bei wachsender zweiten Komponente von x ?
- Wie ändern sich die Kosten $x_1 + x_2 + x_3$ bei wachsender dritten Komponente von x ?
- Welche Komponente von x wird im ersten Schritt des Simplex-Verfahrens erhöht und warum?
- Wie sieht dann die neue Ecke aus?
- Warum ist diese Ecke optimal?