

Lernziele

In diesem Praktikum sollen Sie üben und lernen:

- Einlesen von Matrizen aus Dateien
- Übergabe von Funktionen als Parameter
- Verschiedene Speicherformate für Matrizen
- Grafische Darstellung der Konvergenz des (konjugierten) Gradientenverfahrens

Am Anfang wollen wir Ihre Kenntnisse über das Lösen von linearen Gleichungssystemen und die Programmierung in Matlab etwas auffrischen. Dazu stellen wir Ihnen einige Fragen bzw. einige off-line Aufgaben.

Machen Sie bitte zuerst die folgenden off-line Übungen bevor Sie sich einloggen!

Off-line Aktivitäten

Kurz und knapp

Geben Sie bitte eine kurze Antwort zu jeder der folgenden Fragen. Ihre Antwort sollte so kurz und präzise wie möglich sein; versuchen Sie es mit zwei bis drei Sätzen.

1. Worin unterscheidet sich in Matlab eine Funktion von einem Funktions-handle?

Ihre Antwort:

2. Wie kann man in Matlab eine Funktion durch ihren Funktions-handle ausführen? (Geht es nur mit dem Befehl `feval`?)

Ihre Antwort:

3. Was versteht man unter einer namenlosen (anonymous) Funktion?

Ihre Antwort (geben Sie ein Beispiel an):

4. Was bewirkt der Befehl `fid = fopen('daten.dat')` in Matlab? Und was bedeutet es, wenn `fid` den Wert `-1` bekommt?

Ihre Antwort:

5. Wofür verwendet man den Befehl `fclose(fid)` in Matlab?

Ihre Antwort:

6. Wofür verwendet man den Befehl `fscanf` und `fgetl` in Matlab?

Ihre Antwort:

Programmausgaben

Für jedes der folgenden Programmsegmente, lesen Sie zuerst die Zeilen und schreiben Sie die Ausgabe an die dafür vorgesehene Stelle.

7. Wie lautet die Ausgabe des folgenden Programms, wenn in der Datei `daten.dat` nur eine Zeile mit dem Inhalt `'1 2 3 4'` steht?

```
1 fid=fopen('daten.dat');
2 while 1
3     tline = fgetl(fid);
4     if ~ischar(tline), break, end
5     disp(tline)
6 end
7 fclose(fid);
```

Ihre Antwort:

8. Vorausgesetzt die Textdatei `exp.txt` sieht wie folgt aus

```
0.00  1.00000
0.10  1.10517
0.20  1.23145
0.30  1.31828
```

Was bedeutet der Parameter `[2 inf]` in der 2. Zeile?
Wie lautet die Ausgabe des folgenden Programms?

```
1 fopen('exp.txt');
2 a = fscanf(fid,'%g %g',[2 inf])
3 fclose(fid)
```

Ihre Antwort:

Korrigieren Sie den Code

Für jedes der folgenden Codesegmente sollen Sie feststellen, ob ein Fehler enthalten ist. Falls ein Fehler vorliegt, markieren Sie diesen und spezifizieren Sie, ob es sich dabei um einen logischen oder Syntaxfehler handelt. Schreiben Sie die korrigierten Anweisungen jeweils in jeden dafür vorgesehenen Bereich unter der Problemstellung. Falls das Segment keinen Fehler enthält, schreiben Sie einfach „kein Fehler“. [Bemerkung: Es kann sein, dass ein Programm mehrere Fehler enthält.]

9. Die folgende Matlab-Funktion soll das Produkt der Matrix A mit einem Vektor x zurückgeben, wobei die Matrix A im sogenannten Koordinatenformat gespeichert ist, dem einfachsten Speicherformat für schwach besetzte Matrizen.

Das Format sei an einem Beispiel erklärt:

$$A := \begin{pmatrix} 11 & 12 & 0 & 14 & 0 & 0 \\ 0 & 22 & 23 & 0 & 0 & 0 \\ 31 & 0 & 33 & 34 & 0 & 0 \\ 0 & 42 & 0 & 44 & 45 & 46 \\ 0 & 0 & 0 & 0 & 55 & 56 \\ 0 & 0 & 0 & 0 & 65 & 66 \end{pmatrix} \dots \text{wird zu } \dots$$

value	65	56	46	34	44	45	22	66	11	12	31	55	23	33	14	42
row-index	6	5	4	3	4	4	2	6	1	1	3	5	2	3	1	4
col-index	5	6	6	4	4	5	2	6	1	2	1	5	3	3	4	2

```

1     function Ax = MatVek(value, row, col, x);
2     Ax = zeros(max(row), 1);
3     for k = 1 : length(row)
4         Ax(row(k)) = value(k) * x(row(k));
5     end

```

Ihre Antwort:

Praktikumsaufgabe 13 – Das Gradientenverfahren

Lesen Sie die Aufgabenstellung, studieren Sie dann die vorgegebenen Programmzeilen. Ersetzen Sie dann die %% Kommentare im vorgegebenen Code durch Matlab-Anweisungen und führen Sie das Programm aus.

• Problembeschreibung

- (1) Sie sollen eine Matlab-Funktion schreiben, die zu einer gegebenen Matrix $A \in \mathbb{R}^{n \times n}$, einer rechten Seite b , einem Startvektor x_0 und einer natürlichen Zahl k , die ersten $k-1$ Iterierten (und x_0) des Gradientenverfahrens (als Matrix $n \times k$) zurückgibt.

Der Aufruf sollte somit die folgende Form haben:

```
iterierte = ex13x01(A,b,x,k)
```

- (2) Schreiben Sie eine Funktion `ex13x02.m`, die zu einer Matrix $A \in \mathbb{R}^{n \times n}$ und Vektoren b, x den Wert $f(x) := \frac{1}{2} x^T A x - b^T x$ zurückgibt.
- (3) Modifizieren Sie die Funktion `ex13x03.m` so, dass der Farbverlauf „glatt“ genug ist und Sie etwa 20 Höhenlinien sehen, wenn Sie die Funktion wie folgt aufrufen `ex13x03([2,1;1,2],[2;3],[-10,10,-10,10])`.

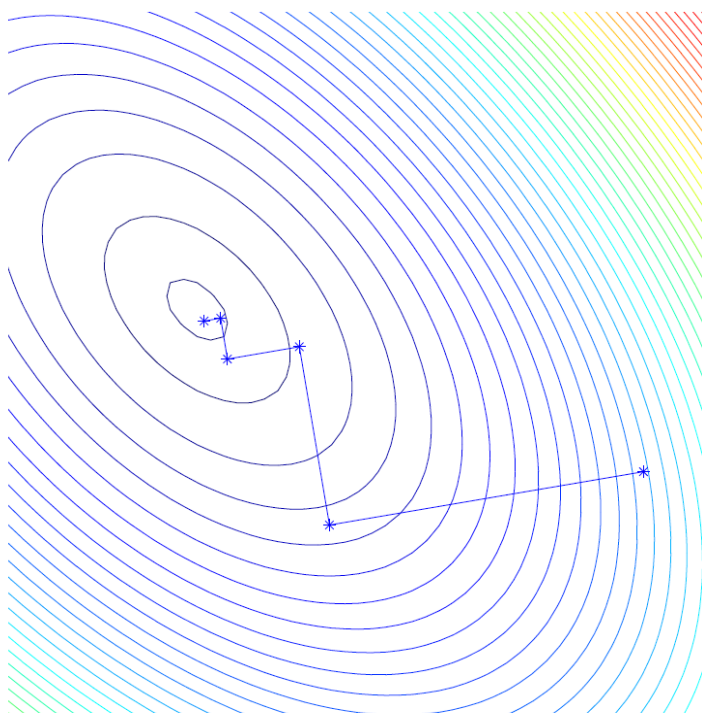
```
% ex13x03.m
% visualize the functional f(x) = 1/2 * x^T A x - b^T x
% on a proper grid

function ex13x03(A,b,xmin,xmax,ymin,ymax)

[x,y] = meshgrid(linspace(xmin,xmax,8), linspace(ymin,ymax,8));
z = zeros(size(x));
for j = 1:size(x,1)
    for k = 1:size(x,2)
        z(j,k) = ex13x02(A,b,[x(j,k);y(j,k)]);
    end
end
z = z-max(max(z)); % lower the z-value to plot
surf(x,y,z,'edgecolor','none'),view(2) % surf and contour in 1 figure
hold on
contour(x,y,z,10,'EdgeColor','w');
hold off
axis equal, grid off, axis off
```

- (4) Plotten Sie nun die mittels `ex13x01` gewonnenen 4 Iterierten in die in (3) erstellte Grafik und verbinden Sie dabei zwei aufeinanderfolgende Iterierte durch eine Linie.

Das Ergebnis sieht bei $b = [-1; 1]$, $x_0 = [8; -3]$, obiger Matrix A und entsprechender Anpassung von $xmin$, $xmax$, $ymin$, $ymax$ wie folgt aus.



Zur Erinnerung

Steepest Descent

Input: Initial guess x^0

$$r^0 := b - Ax^0$$

Iteration: $k = 0, 1, \dots$

$$a^k := Ar^k$$

$$\lambda_{opt} := \frac{\langle r^k, r^k \rangle}{\langle r^k, a^k \rangle}$$

$$x^{k+1} := x^k + \lambda_{opt} r^k$$

$$r^{k+1} := r^k - \lambda_{opt} a^k$$

1 matrix-vector-products, 2 inner products, and 2 saxpy's per iteration

Praktikumsaufgabe 14 – cg-Verfahren

Lesen Sie die Aufgabenstellung, studieren Sie dann die vorgegebenen Programmzeilen. Ersetzen Sie dann die %% Kommentare im vorgegebenen Code durch Matlab-Anweisungen und führen Sie das Programm aus.

• Problembeschreibung

Realisieren Sie den folgenden C-Code in Matlab. Der Aufruf der Matlab-Funktion soll lauten `[x,iterierte] = ex14x01(A, b, x0, tol)`, dabei sollen in `iterierte` die Iterierten `[x0,x1,x2,x3,...]` gespeichert werden. (Dies kann man z.B. durch die Zeilen `iterierte = x0;` und `iterierte = [iterierte,x];` an entsprechender Stelle realisieren.) Legen Sie k_{\max} innerhalb der Routine fest, ggf. müssen Sie den Wert $1e-16$ etwas größer wählen.

Testen Sie das Beispiel an den Daten aus Praktikumsaufgabe 13.

Algorithmus cg-Verfahren: **Function** $x \leftarrow \text{cg}(A, b, x, \text{tol})$

```
r := b - Ax
p := r
ρ := (r, r)
k := 1
while ||r|| > max{tol · ||b||, 10-16} and k < kmax
    k := k + 1
    v := Ap, λ :=  $\frac{\rho}{(v,r)}$ 
    x := x + λp
    r := r - λv, ρold := ρ, ρ := (r, r)
    p := r +  $\frac{\rho}{\rho_{\text{old}}}$ 
end
```

Visualisieren Sie, ähnlich wie in Aufgabe 13, die Konvergenz des Verfahrens. Nach zwei Iterationen sollten Sie am Ziel sein!