Weiterbildendes Studium Ingenieurmathematik – WS 98/99

Optimierung mit MATLAB

H.-J. Oberle, G. Opfer, S. Fredenhagen

Arbeitsblatt 1

1. Login-Prozedur

Die Übungen werden an PCs mit Windows NT (Raum 142) oder Sun-Workstations (Raum 144) durchgeführt. Zunächst müssen Sie sich auf dem jeweiligen Computer einloggen. Dies geschieht durch Eingabe der Kennung **am8...** und des Paßwortes. Bei den PCs darf man die Return-Taste erst nach Eingabe des Paßwortes verwenden. Beim ersten Einloggen verwendet man das Standard-Paßwort. Nach dem Einloggen wird man aufgefordert das Paßwort zu ändern.

Die PCs laufen unter Windows NT. Sie dürfen deshalb nicht einfach abgeschaltet werden. Zum Herunterfahren wählt man unter **Start** den Menupunkt **Beenden**.

2. Matlab-Aufruf

Die numerischen Übungen zu dieser Veranstaltung werden mit **Matlab** durchgeführt. Der Aufruf von Matlab erfolgt an den Workstations durch Eingabe von `matlab`.

Bei den PCs ruft man Matlab durch einen Doppelklick auf das entsprechende Symbol auf.

Um Matlab wieder zu verlassen, tippt man einfach

```
>> quit
```

Hierbei bezeichnet `>>` den Matlab-Prompt.

3. Matlab-Anweisungen

Matlab-Anweisungen können direkt eingeben werden. Matlab schreibt dann die Ausgabe direkt auf den Bildschirm, etwa:

```
>> a=sqrt(2)
a =
    1.4142
```

Die Ausgabe des Ergebnisses auf dem Bildschirm ist gelegentlich lästig. Man kann sie verhindern, indem man die Anweisung mit einem Semikolon abschließt.

```
>> a=sqrt(2);
```

In einer Zeile können mehrere Anweisungen gegeben werden, wenn diese jeweils durch ein Komma oder Semikolon getrennt werden.

4. Variablennamen

Variablennamen können beliebig lang sein. Matlab verwendet aber nur die ersten 19 Zeichen eines Namens. Groß- und Kleinbuchstaben werden unterschieden. Ein Variablenname muß mit einem Buchstaben beginnen, dahinter können Buchstaben, Ziffern und _ (Underscore) folgen.

5. Matrizen und Vektoren

Matrizen werden in Matlab mit eckigen Klammern beschrieben. Innerhalb der Klammern werden die Zeilen der Matrix aufgelistet und durch Semikolons voneinander getrennt.

```
>> M=[1,2,3;4,5,6;7,8,9]
M =
    1    2    3
    4    5    6
    7    8    9
```

Vektoren werden analog als einzeilige oder einspaltige Matrizen geschrieben.

```
>> x=[0.1 0.2 0.3]
x =
    0.1    0.2    0.3
>> y=[0.1 0.2 0.3]'
y =
     0.1
     0.2
     0.3
```

Beachten Sie hier den Unterschied zwischen Zeilen- und Spaltenvektoren.

6. Grafik in Matlab

Matlab besitzt sehr ausgeprägte Grafikfähigkeiten, die für den Benutzer relativ einfach zu nutzen sind. Der einfachste Fall ist, daß man gegebene Punkte

x	0	0.5	1	1.5	2	2.5	3
y	0	0.25	1	2.25	4	6.25	9

in ein Diagramm zeichnen möchte. Hierzu gibt muß man das folgende eingeben:

```
>> x=[0 0.5 1 1.5 2 2.5 3];
>> y=[0 0.25 1 2.25 4 6.25 9];
>> plot(x,y,'+')
```

Wenn man nur einfach

```
>> plot(x,y)
```

eingibt, so werden die Punkte geradlinig verbunden. Dabei wird die erste Zeichnung gelöscht. Möchte man beides in ein Diagramm einzeichnen, so muß man nach der ersten Zeichnung

```
>> hold on;
```

eingeben. Nun kann man zum Beispiel die Punkte wieder in das Diagramm einzeichnen. Mit dem Befehl

```
>> plot(x,y,'ro')
```

werden sie mit roten Kreisen gekennzeichnet. Welche anderen Farben und Strichmuster beim Befehl `plot` zu Verfügung stehen, erfährt man, indem man

```
>> help plot
```

eingibt. Mit dem Befehl

```
>> hold off;
```

wird der Grafikbildschirm wieder freigegeben, d.h. er wird von jedem `plot`-Befehl erneut gelöscht.

Möchte man eine Funktion genauer zeichnen, so ist es etwas aufwendig, die Vektoren wie oben komponentenweise einzugeben. Den Vektor `x` hätte man auch mit

```
>> x=0:0.5:3;
```

erzeugen können.

7. Komponentenweise Matrix- und Vektoroperationen

Die Operatoren `+` und `-` bezeichnen die komponentenweise Vektor- und Matrixaddition bzw. Subtraktion. In Matlab können auch alle anderen Operationen komponentenweise durchgeführt werden. Hierbei wird dem Operator ein Punkt vorangestellt. So wird mit `.*` die komponentenweise Multiplikation und mit `./` die komponentenweise Division gekennzeichnet. Um die oben betrachteten Funktionswerte, die zu der quadratischen Funktion $y = x^2$ gehören, einzugeben, hätte man auch

```
>> y=x.^2;
```

schreiben können. Auch die mathematischen Funktionen wie zu Beispiel $\mathrm{sqrt}(x)$, $\sin(x)$ und $\exp(x)$ werden in Matlab komponentenweise auf Vektoren oder Matrizen angewendet. Um den Sinus auf dem Intervall $[-\pi, \pi]$ zu zeichnen muß man

```
>> a=-pi:0.01:pi;
>> b=sin(a);
>> plot(a,b)
```

eingeben.

8. Aufgabe 1.1: Funktionen ($f : \mathbb{R} \to \mathbb{R}$)

Zeichnen Sie die Funktionen

a) $y = \sin(x)$ für $x \in [-\pi, 0]$

b) $y = \sin(x)$ für $x \in [-\pi, \pi]$

c) $y = 3x^2 - \frac{17}{3}x^3 + \frac{17}{4}x^4 - \frac{7}{5}x^5 + \frac{1}{6}x^6$

und beantworten Sie anhand der Zeichnungen die nachfolgenden Fragen:
Ist die Funktion konvex, bzw. in welchen Intervallen ist sie konvex?
Besitzt sie lokale/globale Extrema?

9. Musterlösungen

Zu jeder Aufgabe auf diesem Blatt gibt es ein Programm mit einer Musterlösung, das sich in einem Unterverzeichnis ihres Heimatverzeichnisses befindet. Matlab-Programme haben die Endung `.m`. Um die Programme nutzen zu können, müssen Sie zunächst einmal in das entsprechende Verzeichnis wechseln.

An einem PC gibt man hierzu

```
>> cd h:\blatt1
```

ein. Um zu sehen, welche Dateien es in dem Verzeichnis gibt, braucht man nur

```
>> dir
```

einzutippen.

Auf den Workstations kann man mit

```
>> cd blatt1
```

in das entsprechende Verzeichnis wechseln. Mit

```
>> dir
```

kann man sich die Dateien anzeigen lassen.

Um ein Programm auszuführen, gibt man einfach seinen Namen ohne Endung ein. Für das Programm zu der Aufgabe 1a, muß man

```
>> aufg1a
```

eintippen. Zu Aufgabe 1.1c gibt es ein zweites Programm `aufg1c2.m`, das zusätzlich zu der Funktion auch ihre Ableitung zeichnet.

10. 3D-Grafikbefehle

In Matlab stehen auch verschiedene dreidimensionale Grafikbefehle zur Verfügung. Analog zu den Vektoren, die im zweidimensionalen Fall das Gitter auf der x-Achse und die dazugehörigen Funktionswerte bezeichnen, werden hier Matrizen verwendet, um das Gitter in der xy-Ebene darzustellen. Ein solches Gitter erstellt man mit

```
>> x=-1:0.1:3;
>> y=-1.5:0.05:2;
```

```
>> [X,Y]=meshgrid(x,y);
```

oder kürzer mit

```
>> [X,Y]=meshgrid(-1:0.1:3,-1.5:0.05:2);
```

Mit

```
>> Z=X.^2+Y.^2;
>> mesh(X,Y,Z)
```

kann man die Funkion $z = x^2 + y^2$ auf dem obigen Gitter zeichnen. Die Höhenlinien der Funktion kann man mit

```
>> contour(X,Y,Z)
```

zeichnen. Dabei werden die Höhenlinien von Matlab aus den in `Z` gespeicherten Funktionswerten berechnet. Es ist möglich die Zahl der Höhenlinien festzulegen. Zum Beispiel werden mit

```
>> contour(X,Y,Z,3)
```

nur drei Höhenlinien eingezeichnet. Auch die Höhe der Höhenlinien kann man festlegen. Mit

```
>> contour(X,Y,Z,[0.01,0.1,1,2.5,4,9])
```

werden nur die Höhenlinien für $f(x,y) = 0.01$, $f(x,y) = 0.1$, $\ldots$ und $f(x,y) = 9$ eingezeichnet. Dabei kann es passieren, daß man eine Höhenlinie zeichnen möchte, für die gar nicht ausreichende Informationen in der Matrix `Z` gespeichert sind. Diese wird dann von Matlab weggelassen.

11. Aufgabe 1.2: Funktionen $(f : \mathbb{R}^2 \to \mathbb{R})$

Zeichnen Sie die folgenden Funktionen und ihre Höhenlinien

a) $f(x,y) = x^2y^2 + (x^2-1)^2$

b) $f(x,y) = 100(y-x^2)^2 + (1-x)^2$ *(Rosenbrock-Funktion)*

Schätzen Sie die Lage der Maxima, Minima und Sattelpunkte der Funktionen. Probieren Sie die lokalen Maxima und Minima durch Höhenlinien zu kennzeichnen.

12. Berechnen und Zeichnen von Gradienten

Es ist in Matlab mit dem Befehl `quiver` auch möglich die zu einer Funktion berechneten Gradienten als Vektorfeld zu zeichnen. Wenn man sich nicht die Mühe machen will, die Funktion mit der Hand abzuleiten, kann man ihren Gradienten mit Matlab numerisch berechnen. Hierzu müssen wieder die Funktionswerte auf einem Gitter bestimmt werden.

```
>> dx=0.4;
>> dy=0.4;
>> [Xg,Yg]=meshgrid(-0.8:dx:2,-1.6:dy:2);
```

```
>> Zg=Xg.^2+Yg.^2;
```

Für die Approximation des Gradienten

```
>> [Fx,Fy]=gradient(Zg,dx,dy);
```

ist es wichtig, daß man sich die Knotenabstände im Gitter merkt. Dieses geht am einfachsten, indem man wie oben hierfür neue Variablen `dx` und `dy` einführt. Das Gradientenfeld kann man dann mit

```
>> quiver(Xg,Yg,Fx,Fy);
```

zeichnen.

13. Aufgabe 1.3: Funktionen ($f : \mathbb{R}^2 \to \mathbb{R}$)

Zeichnen Sie für die Funktionen aus Aufgabe 1.2 Gradienten und Höhenlinien in ein Bild. Versuchen Sie die Lage der Maxima, Minima und Sattelpunkte der Funktionen durch Wahl eines geeigneten Gitters zu kennzeichnen.

Bemerkung: Bei der obigen Vorgehensweise werden die berechneten Gradienten entsprechend ihrer Länge eingezeichnet. Interessiert einen vor allem ihre Richtung, so ist es günstig sie alle auf Länge eins zu normieren. Dieses wird in den Programmen `bsp3n.m`, `aufg3an.m` und `aufg3bn.m` gemacht.