Numerische Lineare Algebra
mit einer Einführung in MATLAB

Gerhard Opfer WS 1998/1999

Bogen 3: Übungen am 12. November 1998

Lösungshinweise

Aufgaben und Lösungen auch in `http://www.math.uni-hamburg.de/home/opfer/aufgabenB.html`

Präsenzaufgaben:

Aufgabe P7. Beispiel: `j=3;x=pi;`
`disp(['Aufgabe ',int2str(j),' hat die Lsung x = ',num2str(x),'.']);`
Ergebnis: `Aufgabe 3 hat die Lösung x = 3.1416.`
Funtionsprozeduren benötigen eigene Dateien. Der Aufrufname ist der Dateiname. Sei er `trig.m`.

```
function [s,c]=trig(x);
s=sin(x);c=cos(x);
return; %return kann auch entfallen
```

Außerhalb von `trig.m`:

```
x=linspace(0,2*pi,51);
[si,co]=trig(x);
plot(x,[si;co]); % [si;co] ist eine zweizeilige Matrix
figure(1); plot(x,si);axis([0,2*pi,-1,1]);legend('Sinus');axis off
print -dps2 si.ps
figure(2); plot(x,co,'x');axis([0,2*pi,-1,1]);legend('Kosinus');axis off
print -dps2 cox.ps
figure(3); stem(x,co,'--');axis([0,2*pi,-1,1]);legend('Kosinus ');axis off
print -dps2 stem.ps
```

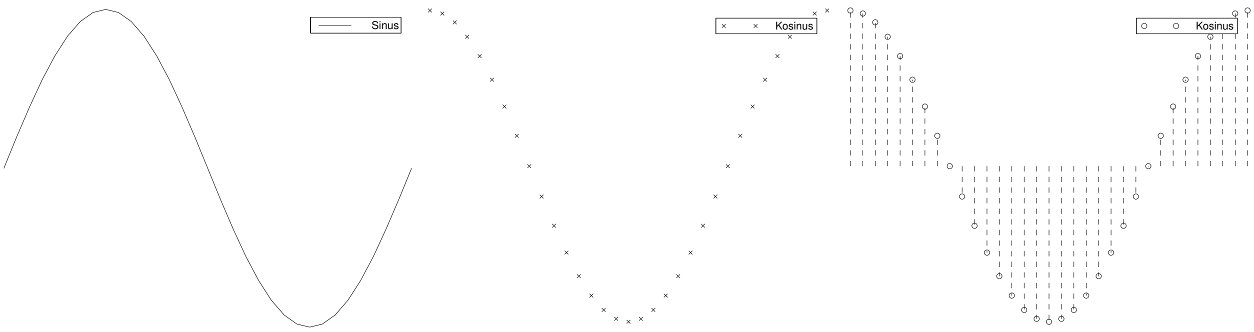


Abbildung L3.1: Die Wirkung der obigen Befehle mit `plot, stem, legend`

Aufgabe P8. `function u=Glocke(x,y,z)`

```
%Die Glockenkurve wird bis zu drei Argumenten programmiert
%wesentlicher Befehl ist:  nargin (number of arguments in)
if nargin==1, u=exp(-x.*x); end;
if nargin==2, u=exp(-x.*x-y.*y); end;
if nargin==3, u=exp(-x.*x-y.*y-z.*z); end;
```

Höhenlinien und Konturlinien sind mit `surfc` (und einigen Tricks) hinten gezeichnet in Abbildung L3.4.

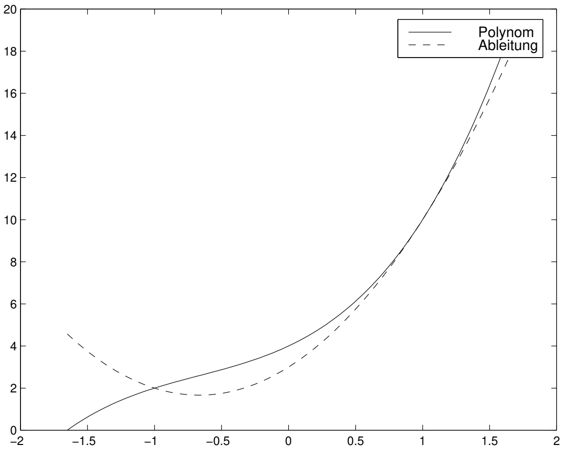
Aufgabe P9. Wir schreiben das Programm direkt ab:

```
%Polynom und Ableitung in einem Bild
p=[1 2 3 4]; %Polynomkoeffizienten, 1 ist Hchstkoeffizient
r=roots(p);  %Nullstellen, mindestens eine ist reell
J=find(r==real(r)); %J enthlt die Nummern der reellen Nullstellen
r=r(J); %die komplexen Nullstellen werden weggelassen
if length(J)==1
  rl=min([-1,-abs(r)]);rr=max([abs(r),1]);
  else, rl=min([r' -1]);rr=max([r' 1]);
end; %if
```

```
x=linspace(rl,rr);
y=polyval(p,x);  %das Polynom wird auf dem Intervall [rl,rr] ausgewertet
pabl=polyder(p); %die Koeffizienten der Ableitung werden berechnet
yabl=polyval(pabl,x); %die Ableitung wird ausgerechnet
figure(1); %Fenster mit Nr. 1
plot(x,y,x,yabl,'--'); legend(['Polynom  ';'Ableitung']);
print -dps2 pundabl.ps %Postscript level2 (keine Farben)
%======================================================================
%Polynom durch Nullstellen N gegeben
N=[0 1 1 3 4 5 6];
%N=[i -i];  %gibt langweiliges Bild x^2 + 1
p1=poly(N); %Koeffizienten werden berechnet
x1=0:0.05:6;
y1=polyval(p1,x1);
figure(2) %neues Fenster mit Nr. 2
plot(x1,y1,[0 6],[0 0]); %[] etc bewirkt Einzeichnen der x-Achse
text(1,-5,'| doppelte Nullstelle'); %Text an einer bestimmten Stelle plazieren
print -dps2 pmitwurzeln.ps
%die hier beschriebene Methode zum Zeichnen ist instabil:
%besser: y=(x-N(1))(x-N(2))...(x-N(6))

```

Das Programm erzeugt die folgenden Bilder:



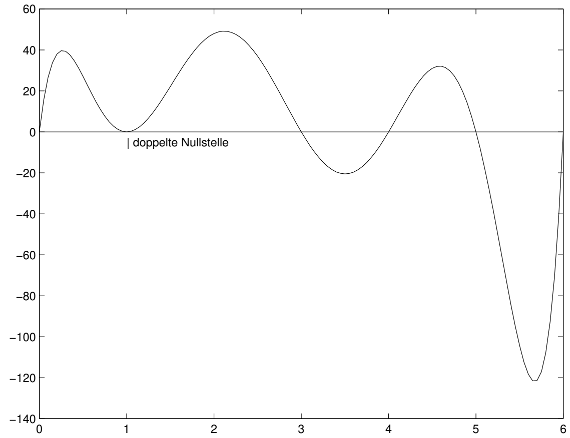


Abbildung L3.2: Polynom mit Ableitungen (links) und mit vorgegebenen Nullstellen

Hausaufgaben:

Aufgabe 5. Zum Beweis gucken Sie in der entsprechenden Literatur unter *Spektralnorm* nach, z. B. bei Opfer, loc. cit. S. 186.

Aufgabe 6. Das Programm könnte so aussehen.

```
%Wir wollen Funktionen mit den expliziten Namen f1,f2,...,f10
%auswerten. Eine Funktion sollen zuf\"allig ausgew\"ahlt und
%gezeichnet werden.
j=round(10*rand+0.5); if j==11,j=10,end; %if
x=linspace(-j,j,1001);
%Die zuf\"alligen Zahlen liegen in [0.5,10.5].
%Man k\"onnte theoretisch also auch eine elf erwischen.
%Es werden drei Techniken beschrieben, die alle das
%gleiche Ergebnis haben.
%1.
[y1]=eval(['f',int2str(j),'(x)']);
%2.
f=['f',int2str(j)];
[y2]=feval(f,x);
%3.
F=['f1 ';'f2 ';'f3 ';'f4 ';'f5 ';'f6 ';'f7 ';'f8 ';'f9 ';'f10'];
%F ist eine Matrix mit 10 Zeilen; F(j,:) liefert also 'fj '
[y3]=feval(F(j,:),x);
plot(x(1:500),y3(1:500),'r',x(501:1001),y3(501:1001),'b');
%das Bild ist links rot, rechts blau
eval(['print -dpsc2 bild',int2str(j),'.ps']);
```

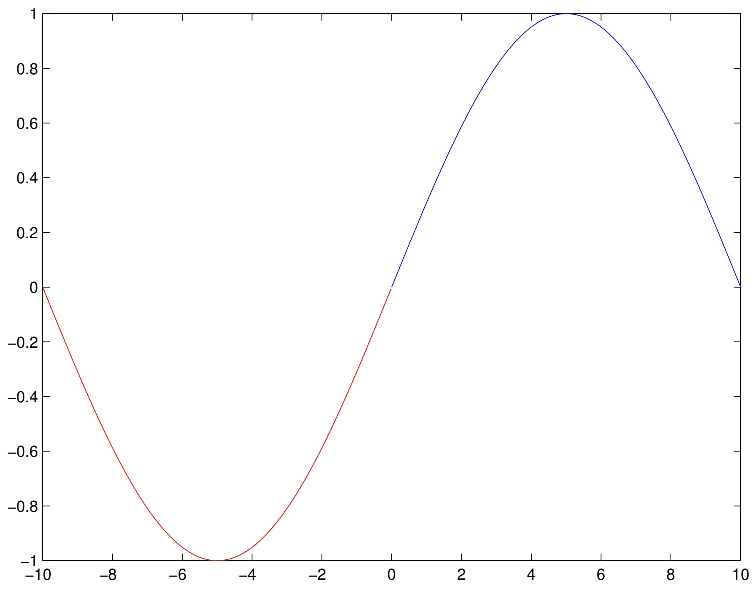


Abbildung L3.3: Graph mit unterschiedlicher Färbung (nur auf einem Farbdrucker zu sehen)

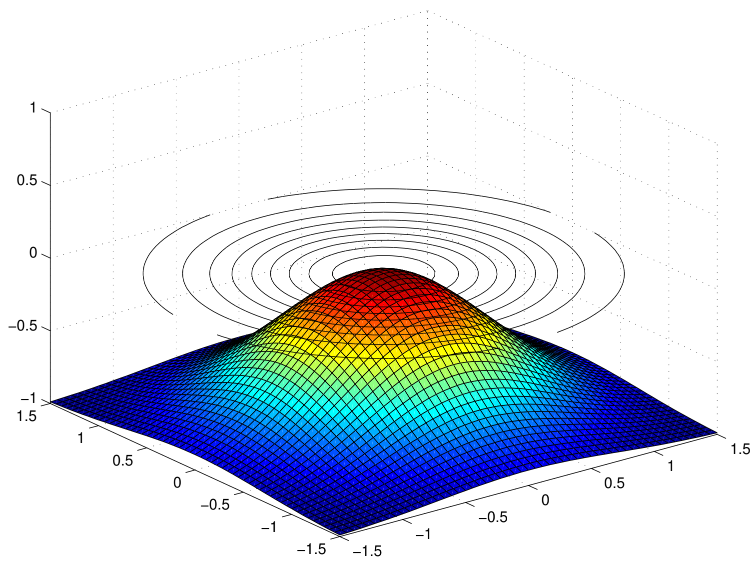


Abbildung L3.4: Glockenkurve mit Höhenlinien (Aufgabe P8)

Hinweis: In der auch in diesem Semester stattfindenden Vorlesung *Weiterbildendes Studium Ingenieurmathematik* mit dem Untertitel *Optimierung mit MATLAB* gibt es viele weitere Hinweise und Beispielprogramme, zu finden unter `http://www.math.uni-hamburg.de/cm/wbing/aufgaben.html`

`(Version of 11. 11. 1998 at 891 minutes since midnight, machine used: ultra)`