Universität Konstanz
FB Mathematik & Statistik
Prof. Dr. M. Junk
Dipl.-Phys. M. Rheinländer

Analysis II

Ergänzungsblatt zu Übung 7

Extras:

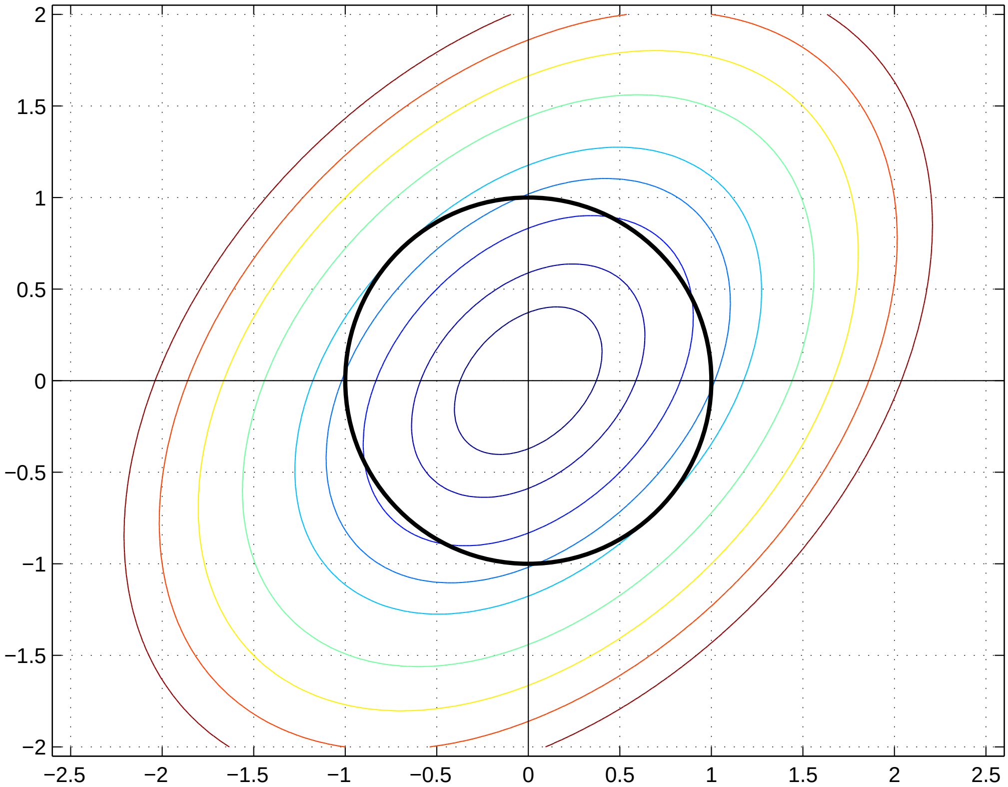
Demo Matlab-Programm `powermethod.m` zu Aufgabe 2 (herunterladbar von der Vorlesungs-Homepage).

```
%==========================================================================================
% Illustration of the power method and the Rayleigh quotient to approximate the largest
% eigenvector of a symmetric matrix.
%
% M.Rh. 06/06/06
%==========================================================================================
clear all;
format long;

N = 10;                        % number of iteration to be performed
k = 1;                         % index of component used to compute eigenvalue
S = 1/8 * [13 -5; -5  13];     % given symmetric matrix

if(0)                          % inactive for if(0), active for if(1)
  A = rand(8);                 % generates random test matrix
  S = (A+A');                  % symmetrizes A
end;

eigval     = eig(S);           % internal matlab function to compute eigenvalue
[m,mindex] = max(abs(eig(S))); % searches index of eigenvalue with largest modulus
lambda     = eigval(mindex);   % searched eigenvalue for comparison (taken as exact value)


% definition of the initialization vector
dim = size(S);
x   = zeros(dim(1),1); x(1)=1;

% iteration
for i=1:N,
  y = x;                               % old iterate
  x = S*x;                             % new iterate
  lambda_1(i) = x(k)/y(k);             % approximated eigenvalue by standard power method
  lambda_2(i) = x'*S*x/norm(x,2)^2;    % approximated eigenvalue by Rayleigh quotient
end;
[[1:N]', (lambda_1-lambda)', (lambda_2-lambda)']  % prints errors associated to each iteration



% level-line plot of x-> <x,Sx> if S is a 2x2 matrix
if(dim(1)==2)
  xmin  = -2.6; xmax = 2.6; dx = 0.02;
  ymin  = -2.0; ymax = 2.0; dy = 0.02;
  [X,Y] = meshgrid([xmin:dx:xmax],[ymin:dy:ymax]);
   Z    = S(1,1)*X.^2 + (S(1,2)+S(2,1))*X.*Y + S(2,2)*Y.^2;
  figure(1); clf; hold on; box on; grid on; axis equal;
  V = [0.10*lambda, 0.25*lambda, 0.50*lambda, 0.75*lambda, lambda, ...
       1.50*lambda, 2.00*lambda, 2.50*lambda, 3.00*lambda];
  contour(X,Y,Z,V);
  rectangle('Curvature',[1,1],'Position',[-1 -1 2 2],'LineWidth',2);
  line([xmin xmax],[0 0],'Color','k');
  line([0 0],[ymin ymax],'Color','k');
end;
```

Konturlinien einer altbekannten quadratischen Funktion (siehe Aufgabe 4.3, Figur 5)

$$x \mapsto \left\langle \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \frac{1}{8}\begin{pmatrix} 13 & -5 \\ -5 & 13 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \right\rangle$$

zu den Niveaus $(0.1, 0.25, 0.5, 0.75, 1.0, 1.50, 2.00, 2.50, 3.00) * \lambda_{max}$, wobei λ_{max} dem größeren der beiden Eigenwerte $\frac{9}{4}, 1$ entspricht.



Beispiel Output:

```
>> powermethod

ans =

    1.00000000000000  -0.62500000000000  -0.20618556701031
    2.00000000000000  -0.38461538461538  -0.04694146985477
    3.00000000000000  -0.20618556701031  -0.00956049722055
    4.00000000000000  -0.10088272383354  -0.00190015567962
    5.00000000000000  -0.04694146985477  -0.00037579781161
    6.00000000000000  -0.02130740931866  -0.00007424957942
    7.00000000000000  -0.00956049722055  -0.00001466728272
    8.00000000000000  -0.00426724185531  -0.00000289726831
    9.00000000000000  -0.00190015567962  -0.00000057230098
   10.00000000000000  -0.00084522744149  -0.00000011304715
```