Oplossingen NLA 4

***Nico Vervliet**, KU Leuven* *19 mei 2016*

1 Theorie

Opgave 1

Er zijn twee manieren om dit aan te tonen: via de oplossing van een kleinstekwadratenprobleem en via het argument dat de doelfunctie minimaliseert:

1. Het minimalisitieprobleem

$$\min_{\rho \in \mathbb{R}} ||\mathbf{A}\mathbf{x} - \rho\mathbf{x}||_2$$

kan gezien worden als een kleinstekwadratenprobleem

$$\min_{\bar{\mathbf{x}}} \left|\left|\bar{\mathbf{A}}\bar{\mathbf{x}} - \bar{\mathbf{b}}\right|\right|_2$$

dat als oplossing

$$\bar{\mathbf{x}} = \bar{\mathbf{A}}^{\dagger}\bar{\mathbf{b}}$$

heeft, door te stellen dat $\bar{\mathbf{x}} = \rho$, $\bar{\mathbf{A}} = \mathbf{x}$ en $\bar{\mathbf{b}} = \mathbf{A}\mathbf{x}$.

2. Het optimum van een minimalizatieprobleem kan gevonden worden door de gradiënt gelijk te stellen aan nul. Het minimalisitieprobleem

$$\min_{\rho \in \mathbb{R}} ||\mathbf{A}\mathbf{x} - \rho\mathbf{x}||_2$$

is equivalent aan

$$\min_{\rho \in \mathbb{R}} \frac{1}{2} ||\mathbf{A}\mathbf{x} - \rho\mathbf{x}||_2^2 = \min_{\rho} f(\rho).$$

Dus

$$
\begin{aligned}
\frac{\mathrm{d}f}{\mathrm{d}\rho} &= \frac{\mathrm{d}}{\mathrm{d}\rho}\frac{1}{2}(\mathbf{Ax}-\rho\mathbf{x})^T\mathbf{Ax}-\rho\mathbf{x} \\
&= \frac{\mathrm{d}}{\mathrm{d}\rho}\left(\frac{1}{2}\mathbf{x}^T\mathbf{A}^T\mathbf{Ax}-\rho\mathbf{x}^T\mathbf{Ax}+\frac{1}{2}\mathbf{x}^T\mathbf{x}\right) \\
&= -\mathbf{x}^T\mathbf{Ax}+\rho\mathbf{x}^T\mathbf{x}=0.
\end{aligned}
$$

In beide gevallen vinden we als oplossing het Rayleigh-quotiënt:

$$\rho = \frac{\mathbf{x}^\top\mathbf{Ax}}{\mathbf{xAx}}.$$

Opgave 2

In beide gevallen zijn inverse iteratie en Rayleigh quotiënt iteratie geschikt. Geef een bespreking van de voor- en nadelen van de methoden, rekening houdend met

- De kost om de matrix tridiagonaal te maken (aangezien de matrix symmetrisch is).
- De kost per iteratie eens de matrix tridiagonaal is.
- De kost om een stelsel op te lossen (bij inverse iteratie kan het stelsel eerst gefactorizeerd worden aangezien μ constant is).
- De convergentie (lineair versus kubisch).
- Hoe de schatting voor de eigenwaarde μ nodig voor inverse iteratie bepaald wordt.

2 Eigenwaarden bepalen met de Arnoldi iteratie

Opgave 3

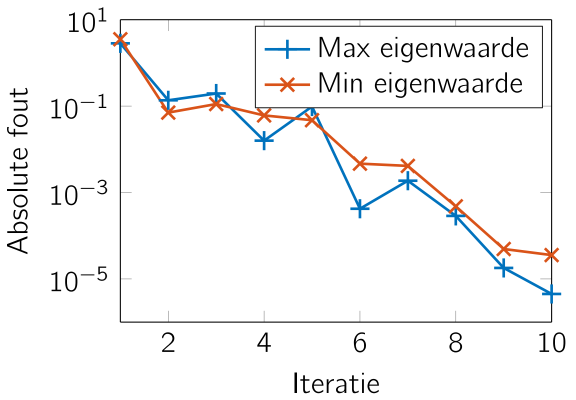
De volgende code berekent het verschil tussen grootste (kleinste) eigenwaarde en de grootste (kleinste) Ritz waarde.

```
function [H, Q, maxd, mind] = arnoldiew(A, b, N)
% function [H, Q, maxd, mind] = arnoldiew(A, b, N)
%
% maximum en minimum eigenwaarden met Arnoldi iteratie
% maxd(:) absolute waarde van verschil tussen
%         grootste eigenwaarde en grootste Ritz waarde
```

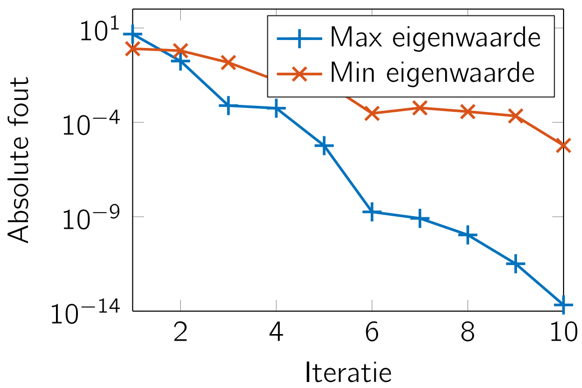
```
% mind(:) absolute waarde van verschil tussen
%         kleinste eigenwaarde en kleinste Ritz waarde

% Eigenwaarden van volle matrix
ewA = eig(full(A));
maxewA = max(ewA);
minewA = min(ewA);

% preallocation
n = size(A,1);
Q = zeros(n,N+1);
H = zeros(N+1,N);

Q(:,1) = b(:)/norm(b);
for n = 1:N
    v = A*Q(:,n);
    for j = 1:n
        H(j,n) = Q(:,j)'*v;
        v = v - H(j,n)*Q(:,j);
    end
    H(n+1,n) = norm(v);
    if H(n+1,n) <= 0, break; end
    Q(:,n+1) = v/H(n+1,n);

    % bereken Ritz waarden en fouten
    ewH = eig(full(H(1:n,1:n)));
    maxewH = max(ewH);
    minewH = min(ewH);
    maxd(n) = abs(maxewA - maxewH);
    mind(n) = abs(minewA - minewH);
end
```

```
m = 10;
L = eigint(4,5,m);
L(1) = 8;
%L(2) = 2;
[A,V] = willglv(L);
b = rand(m,1);

[H, Q, maxd, mind] = arnoldiew(A, b, m);

t = 1:m;
semilogy(t, maxd, '+-', t, mind,'x-');
xlabel('Iteratie');
```

```
ylabel('Absolute fout');
legend('Max eigenwaarde','Min eigenwaarde')
```



Figuur 1: Fouten voor eigenwaarden in $(4,5)$.



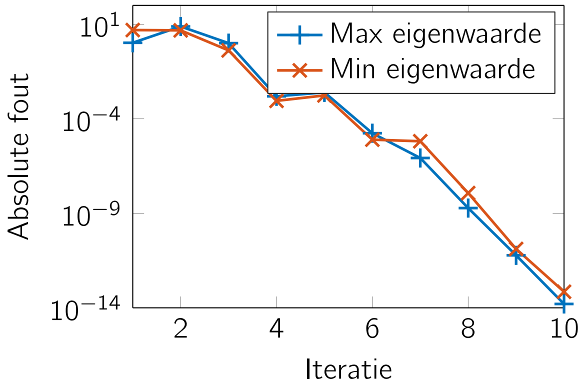
Figuur 2: Fouten voor eigenwaarden in $(4,5)$ en 8.

De resultaten voor de drie gevallen worden getoond in Figuren 1, 2 en 3. Extreme eigenwaarden worden sneller goed benaderd.

Opgave 4

We gebruiken de volgende routine om de Ritz waarden te berekenen:

```
function [H, Q, rw] = arnoldiritz(A, b, N)
% function [H, Q, rw] = arnoldiritz(A, b, N)
%
% rw: Ritswaarden

rw = nan(N,N);
```

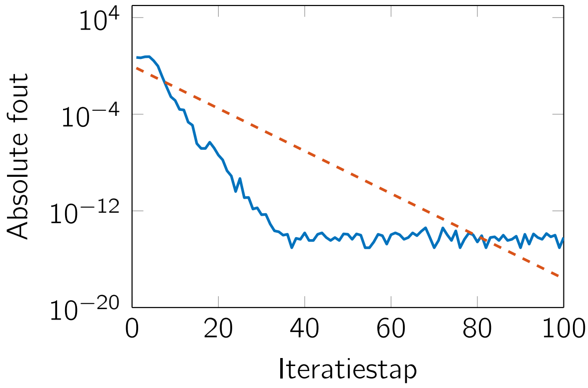


Figuur 3: Fouten voor eigenwaarden in $(4, 5)$, 8 en 2.

```
% preallocation
n = size(A,1);
Q = zeros(n,N+1);
H = zeros(N+1,N);

Q(:,1) = b(:)/norm(b);
for n = 1:N
    v = A*Q(:,n);
    for j = 1:n
        H(j,n) = Q(:,j)'*v;
        v = v - H(j,n)*Q(:,j);
    end
    H(n+1,n) = norm(v);
    if H(n+1,n) <= 0, break; end
    Q(:,n+1) = v/H(n+1,n);

    % Bereken Ritz waarden
    ewH = eig(full(H(1:n,1:n)));
    rw(1:n,n) = ewH;
end
```

```
N = 100;
A = sprand(1000,1000, 0.01);
b = randn(size(A,2),1);
eigs(A)
[H, Q, rw] = arnoldiritz(A,b,N);
rwmax = real(max(rw));
eigmax = real(max(eigs(A)));

clf;
```

```
semilogy(abs(rwmax-eigmax))
hold all
n = 1:N;
semilogy(n,(2/3).^n,'--');
xlabel('Iteratiestap');
ylabel('Absolute fout');
```



Figuur 4: Convergentie van Ritz waarden.

De convergentie van de Ritz waarden wordt getoond in Figuur 4. De waarden convergeren sneller dan de voorspelde convergentie met een factor $\left(\frac{2}{3}\right)^n$ met n de iteratiestap. De snellere convergentie is te verklaren door de geïsoleerde eigenwaarde (plot hiervoor de eigenwaarden van `A`.)