

MNUM-PROJEKT, zadanie 2.2

Student: Tomasz Jakubczyk

1. Wyznaczanie wartości własnych macierzy metodą rozkładu QR.

Rozkład QR to taki rozkład macierzy A , że $A=QR$, gdzie Q jest macierzą ortogonalną (czyli $Q^T Q=I$) i R jest górną macierzą trójkątną. Do wyznaczania rozkładu QR posłuży algorytm zmodyfikowany Grama-Shmidta. Przekształca on układ liniowo niezależnych wektorów w układ wektorów ortogonalnych.

```
function [ Q, R ] = ModGSQR( A )
%ModGSQR zmodyfikowany algorytm rozkładu QR Grama-Shmidta
% A - macierz wejściowa
% Q - macierz wyjściowa ortogonalna
% R - macierz wyjściowa górna trójkątna
[m,n]=size(A); %macierz A może nie być kwadratowa
Q=zeros(m,n);
R=zeros(n,n);
d=zeros(1,n);
for i=1:n %rozkład QR
    Q(:,i)=A(:,i);
    R(i,i)=1;
    d(i)=Q(:,i)'*Q(:,i);
    for j=i+1:n
        R(i,j)=(Q(:,i)'*A(:,j))/d(i);
        A(:,j)=A(:,j)-R(i,j)*Q(:,i);
    end
end
for i=1:n %normowanie układu
    dd=norm(Q(:,i));
    Q(:,i)=Q(:,i)/dd;
    R(i,i:n)=R(i,i:n)*dd;
end
end
```

Obliczanie wartości własnych metodą QR bez przesunięć polega na przekształcaniu macierzy $A^{(k)}$ w $A^{(k+1)}$ przez podobieństwo. (z podręcznika)

$$A^{(k)} = Q^{(k)} R^{(k)} \quad \text{-- faktoryzacja,}$$
$$A^{(k+1)} = R^{(k)} Q^{(k)},$$

$$R^{(k)} = Q^{(k)T} A^{(k)},$$

$$A^{(k+1)} = Q^{(k)T} A^{(k)} Q^{(k)},$$

```
function [ ww, iter, status ] = WWBP( A )
%WWBP obliczanie wartości własnych metodą rozkładu QR bez przesunięć
% A - macierz wejściowa symetryczna
% ww - wartości własne
% iter - liczba wykonanych iteracji
```

```

% status - sposób zakończenia
maxIter=1e5;
status=1;
for i=1:maxIter
    b=A-diag(diag(A));
    if max(max(abs(b)))<1e-6
        break;
    end
    [q, r]=ModGSQR(A);
    % [q, r]=qr(A);
    A=r*q;
    ww=diag(A);
    if i==maxIter
        disp('maxIter')
        status=0;
    end
    iter=i;
end

```

Metoda QR bez przesunięć jest wolno zbieżna jeśli wartości własne leżą blisko siebie, więc stosuje się metodę z przesunięciami.

Metoda QR z przesunięciami:

$$\begin{aligned}
 A^{(k)} - p_k I &= Q^{(k)} R^{(k)} \\
 A^{(k+1)} &= R^{(k)} Q^{(k)} + p_k I \\
 &= Q^{(k)T} (A^{(k)} - p_k I) Q^{(k)} + p_k I \\
 &= Q^{(k)T} A^{(k)} Q^{(k)},
 \end{aligned}$$

Za p_k przyjmujemy wartość własną podmacierzy 2×2 z prawego dolnego rogu macierzy $A^{(k)}$, aż wyzerują się wszystkie elementy ostatniego wiersza. Następnie zmniejszamy wymiar macierzy o 1 odrzucając ostatni wiersz i kolumnę (deflacja).

```

function [ ww, iter, status ] = WWZP( A )
%WWZP obliczanie wartości własnych metodą QR z przesunięciami
% A - macierz wejściowa
% ww - wartości własne
maxIter=1e5;
iter=0;
status=1;
n=size(A,1);
ww=diag(zeros(n));
AO=A;%oryginalna macierz wejściowa
for k=n:-1:2
    AK=AO;
    i=0;
    while i<=maxIter && max(abs(AK(k,1:k-1)))>1e-6
        AA=AK(k-1:k,k-1:k);%prawy dolny róg 2x2
        [ww1,ww2]=QPNR(1,-(AA(1,1)+AA(2,2)),AA(2,2)*AA(1,1)-AA(2,1)*AA(1,2));
        if abs(ww1-AA(2,2))<abs(ww2-AA(2,2))
            shift=ww1;
        else
            shift=ww2;
        end
    end

```

```

        AK=AK-eye(k)*shift;%macierz przesunięta
        [q, r]=ModGSQR(AK);
        AK=r*q+eye(k)*shift;%macierz przekształcona
        i=i+1;
        iter=iter+1;
    end
    if i>maxIter
        disp('maxIter')
        status=0;
    end
    ww(k)=AK(k,k);
    if k>2
        AO=AK(1:k-1,1:k-1);%deflacja macierzy
    else
        ww(1)=AK(1,1);
    end
end

function [ x1, x2 ] = QPNR( a, b, c )
%QPNR obliczanie pierwiastków równania kwadratowego
% a,b,c - współczynniki trójmianu kwadratowego
% x1,x2 - pierwiastki
l1=-b+sqrt(b^2-4*a*c);
l2=-b-sqrt(b^2-4*a*c);
if abs(l1)>abs(l2)
    l=l1;
else
    l=l2;
end
x1=l/(2*a);
x2=(-b/a)-x1;
end

```

Wyniki:

Algorytm i rodzaj macierzy	Rozmiar macierzy	Średnia liczba iteracji
Bez przesunięć	5x5	135.23
	10x10	696.33
	20x20	4.78E+003
Z przesunięciami macierz symetryczna	5x5	7.9
	10x10	15.13
	20x20	29.83
Z przesunięciami macierz nie symetryczna	5x5	9.83
	10x10	22.27
	20x20	47.3

W przeprowadzonych próbach z maksymalną liczbą iteracji ustawioną na 100000 wszystkie wywołania zakończyły się przed osiągnięciem tej liczby, ale dla większych macierzy mogło by taki nie być. Oczywiście zwiększenie czułości na zero też miało by

podobny efekt. W praktyce metoda bez przesunięć często może okazywać się niewystarczająca (niezbieżna). Metoda z przesunięciami jest pod tym względem lepsza (szybciej zbieżna), wymaga mniejszego nakładu obliczeń i obsługuje macierze niesymetryczne.

```
A=rand(5)
B=A+A'
[ww, iter, status]=WWBP(B)
ww=sort(ww)
wwg=eig(B)
eWWBP=norm(ww-wwg)
[ww, iter, status]=WWZP(B)
ww=sort(ww)
eWWZP=norm(ww-wwg)
wwg=eig(A)
wwg=sort(wwg)
[ww, iter, status]=WWZP(A)
ww=sort(ww)
eWWZP=norm(ww-wwg)
```

Dla wybranych macierzy porównanie wyników z eig:

Macierz niesymetryczna: A =

0.7196	0.7647	0.6023	0.3033	0.8781
0.5819	0.2899	0.4332	0.7718	0.5617
0.8937	0.7680	0.3445	0.9689	0.1462
0.6131	0.2011	0.0534	0.1006	0.7232
0.8340	0.8671	0.3186	0.9540	0.0958

macierz symetryczna: B =

0.9490	1.2743	0.6972	1.6705	0.4604
1.2743	1.2381	1.1486	0.6574	1.5105
0.6972	1.1486	1.3810	0.2198	1.3639
1.6705	0.6574	0.2198	1.8481	1.9128
0.4604	1.5105	1.3639	1.9128	1.6691

Błędy jako norma różnic między wartościami własnymi:

Bez przesunięć: 2.9959e-13

Z przesunięciami macierz symetryczna: 2.2021e-15

Z przesunięciami macierz niesymetryczna: 6.8768e-09

Są to małe błędy zważając, na to, że dokładność porównywania z zerem ustawiłem na 1e-6.

2. Rozwiązanie układów n równań liniowych $Ax=b$ w sensie minimalizacji sumy kwadratów niespełnienia równań.

Rozwiązanie przy pomocy układów równań normalnych opiera się na tym, że jest przestrzeń błędów rozwiązań układu, która ma minimum.

$$\forall \mathbf{x} \in \mathbb{R}^n \quad \|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}\|_2 \leq \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2.$$

Układy równań liniowych pozwalają na ładne zapisanie analitycznego gradientu pozwalającego natychmiastowo zminimalizować błąd rozwiązania.

$$J(\mathbf{x}) = (\mathbf{b} - \mathbf{A}\mathbf{x})^T(\mathbf{b} - \mathbf{A}\mathbf{x}) = \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x} - 2\mathbf{x}^T \mathbf{A}^T \mathbf{b} + \mathbf{b}^T \mathbf{b}.$$

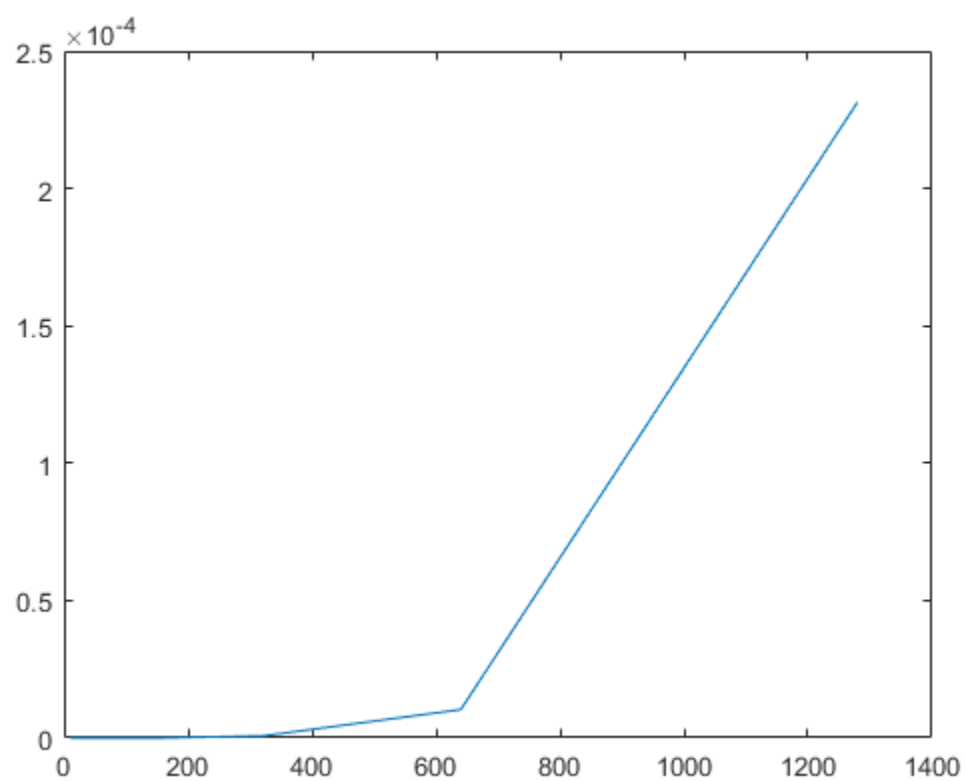
Minimum jest tam gdzie gradient się zeruje.

$$J'(\mathbf{x})^T = 2\mathbf{A}^T \mathbf{A} \mathbf{x} - 2\mathbf{A}^T \mathbf{b} = \mathbf{0}.$$

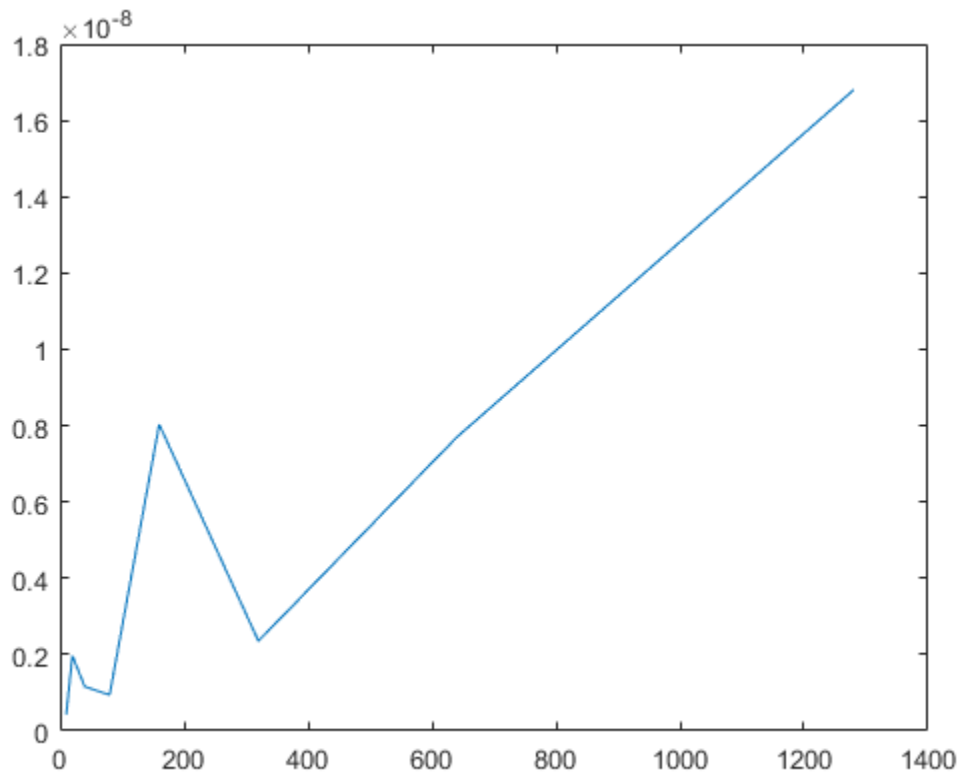
Wynikający z tego układ równań normalnych ma jednoznaczne rozwiązanie.

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}$$

```
t1=tic;
n=5;
k=0;
while toc(t1)<2
    k=k+1;
    n=n*2
    A=zeros(n);
    b=zeros(n,1);
    for i=1:2*n
        for j=1:n
            if i==j
                A(i,j)=1/5;
            else
                A(i,j)=2*(i-j)+1;
            end
        end
        b(i)=1+0.4*i;
    end
    ata=A'*A;
    atb=A'*b;
    a=ata\atb;
    res(k)=norm(atb-ata*a);
    ns(k)=n;
end
plot(ns(1:k),res(1:k));
```



Błędy dla układów równań normalnych 1



Błędy dla układów równań normalnych 2

$$\text{cond}_2(\mathbf{A}^T \mathbf{A}) = (\text{cond}_2 \mathbf{A})^2 = \sigma_1^2 / \sigma_n^2,$$

Jeśli macierz \mathbf{A} jest słabo uwarunkowana pogarsza się wskaźnik uwarunkowania dla układu równań normalnych i powinno się wtedy użyć rozkładu QR.

$$\mathbf{R}^T \mathbf{Q}^T \mathbf{Q} \mathbf{R} \mathbf{x} = \mathbf{R}^T \mathbf{Q}^T \mathbf{b}.$$

$$\mathbf{R} \mathbf{x} = \mathbf{Q}^T \mathbf{b}.$$

```

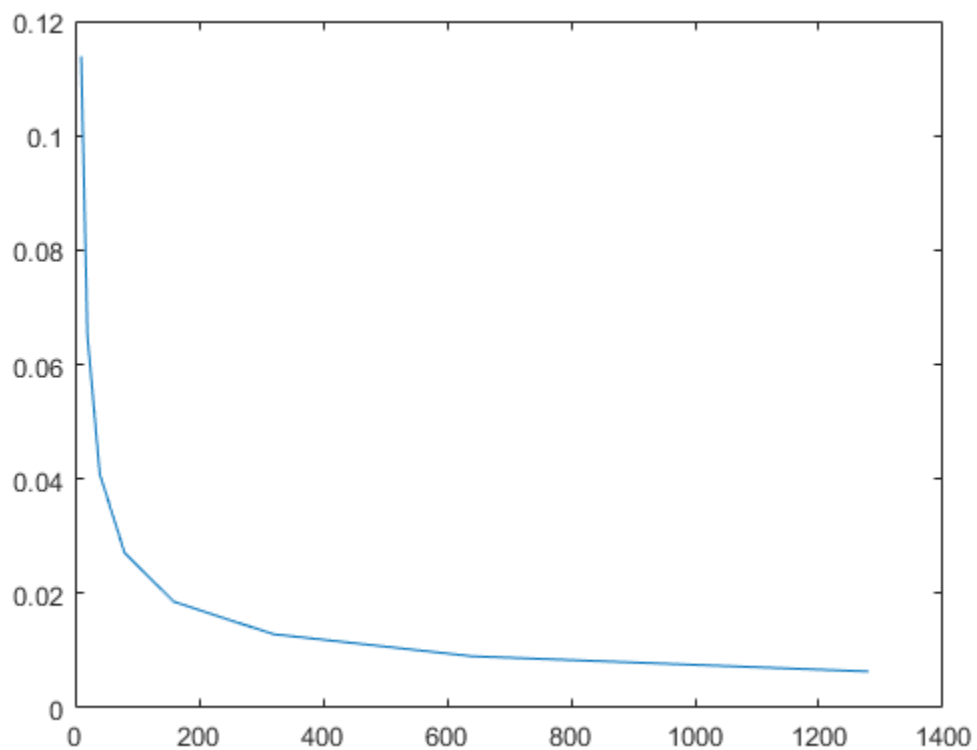
t1=tic;
n=5;
k=0;
while toc(t1)<2
    k=k+1;
    n=n*2
    A=zeros(n);
    b=zeros(n,1);
    for i=1:2*n
        for j=1:n
            A(i,j)=7/(9*(i+j+1));
        end
        if mod(i,2)==0
            b(i)=7/(5*i);
        else

```

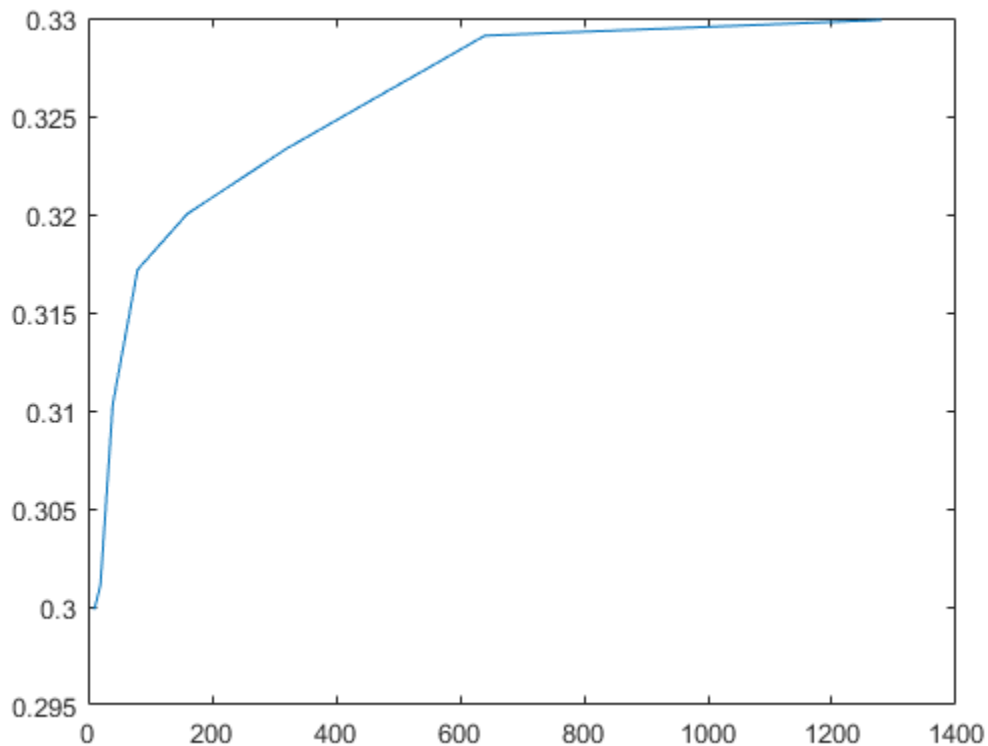
```

        b(i)=0;
    end
end
[Q,R]=qr(A);
a=R\Q'*b;
res(k)=norm(R*a-Q'*b);
ns(k)=n;
end
plot(ns(1:k),res(1:k));

```



Błędy dla układów I rozwiązywanych minimalizacją najmniejszych kwadratów z rozkładu QR



Błędy dla układów 2 rozwiązywanych minimalizacją najmniejszych kwadratów z rozkładu QR

Dla danych układów o rozsądnym rozmiarze układ równań normalnych daje mniejszy błąd, ale widać, że szybko on rośnie. Dla układów równań wynikających z rozkładu QR widać, że błędy nie rosną tak szybko. Układ równań z rozkładu QR dłużej zachowuje uwarunkowanie.