Rozdział 8

Rozwiązywanie równań liniowych.

W tym rozdziale zajmiemy się algorytmami związanymi z algebrą liniową, a dokładnie rozwiązywaniem układów równań liniowych.

Interesuje nas zadanie znalezienia rozwiązania układu równań liniowych:

$$As = f \tag{8.1}$$

gdzie A to macierz nieosobliwa $n \times n$ i f to wektor prawej strony.

8.1 Operator backslash

Podstawowym operatorem rozwiązującym (8.1) jest operator: octave'a \. Jego zastosowanie jest bardzo proste

```
x=A\f
```

Przetestujmy ten operator dla kilku macierzy:

```
A=[1  1;-1  1];
s=[1;  3];
f=A*s;
x=A\f;
x-s
norm(x-s,2)
norm(A*x-f,2)
```

Weźmy macierz prawie osobliwą:

```
A=[1  1+2*eps;1  1];
s=[1;  3]
f=A*s;
x=A\f
```

```
x-s
norm(x-s,2)
norm(A*x-f,2)
```

Wywołanie operatora A\f zwróciło ostrzeżenie i dało niepoprawny wynik. Zauważmy, że norma residualna $\|A*x-f\|_2$ jest mała, podczas gdy rozwiązanie jest zupełnie złe: $\|x-s\|_2 > 1$.

Nasza macierz została specjalnie tak dobrana, aby odpowiedni algorytm używany przez octave'a nie zwrócił poprawnego wyniku - ale w praktycznych zastosowaniach istnieją takie macierze, dla których błędy zaokrągleń powodują, że rozwiązywanie układów równań liniowych z nimi jest obarczone ogromnym błędem.

Operator \ może też w jednym wywołaniu rozwiązać układ równań liniowych z prawą stroną będącą macierzą, tzn.:

$$AX = F$$

gdzie A to macierz nieosobliwa $n \times n$, X, F to macierze $n \times m$.

Wywołujemy po prostu

```
X=A\F
```

Taki układ jest równoważny m układom z macierzą A i prawymi stronami - kolumnami macierzy F, jako że:

$$AX = (Ax_1, Ax_2, \ldots, Ax_m) = (f_1, \ldots, f_m) = F.$$

Pętle w octave'ie działają wolno, więc - jeśli to możliwe - należy unikać ich wywoływania. Na poniższym przykładzie porównamy czas działania operatora \ w pętli i wywołania bezpośredniego. Rozpatrzmy układ z macierzą losową, ale silnie diagonalnie dominującą:

```
n=100;
m=100;
A=2*eye(n,n) + (1/n)*rand(n,n);
S=rand(n,m);X=XX=zeros(n,m);
F=A*S;
tic;X=A\F;t1=toc
#a teraz w petli
tic;
for k=1:m,
  XX(:,k)=A\F(:,k);
endfor
t2=toc
t2/t1
```

Na moim komputerze czas obliczeń w pętli był ponad 50 razy dłuższy.

Sprawdźmy, czy wyniki są poprawne:

```
norm(X-XX,1)
norm(X-S,1)
norm(A*X-F,1)
```

Wszystkie błędy są małe.

8.2 Macierz odwrotna

W octave'ie funkcja **inv**(A) zwraca macierz odwrotną (o ile ona istnieje.)

Sprawdźmy, jak to działa:

```
A=[1 1;-1 1];
iA=inv(A);
A*iA
iA*A
norm(A*iA- eye(2),1)
norm(iA*A- eye(2),1)
```

Sprawdźmy, jak działa ta funkcja dla macierzy prawie osobliwej:

```
A=[1 1+2*eps;1 1];
iA=inv(A);
A*iA
iA*A
norm(A*iA- eye(2),1)
norm(iA*A- eye(2),1)
```

Macierz odwrotna jest poprawna. Teraz sprawdźmy, czy możemy rozwiązać układ równań liniowych z wykorzystaniem macierzy odwrotnej:

```
s=[1;3];
f=A*s;
x=iA*f;
x-s
norm(x-s,2)
norm(A*x-f,2)
```

Wynik jest znów nieprawidłowy.

Macierz odwrotną można też obliczyć jednym wywołaniem operatora \:

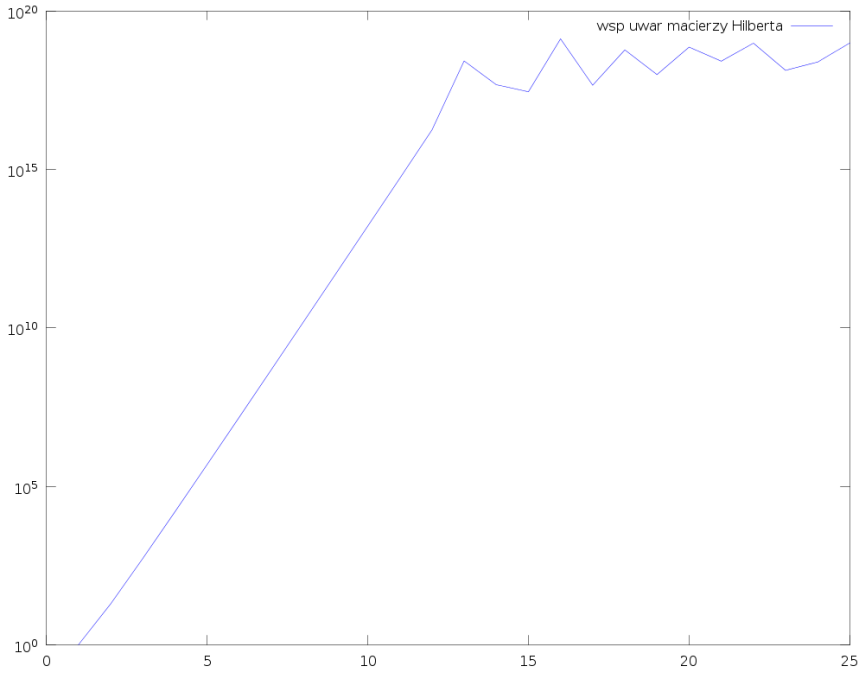
```
iA=A\eye(size(A))
```

Przetestujmy ten sposób:

```
A=[1  1;-1  1];
Id=eye(size(A));
iA=A\Id
iA2=inv(A)
norm(iA*A,1)
norm(iA-iA2,1)
```

Norma różnicy obu macierzy odwrotnych jest identyczna, czyli oba sposoby są równoważne.

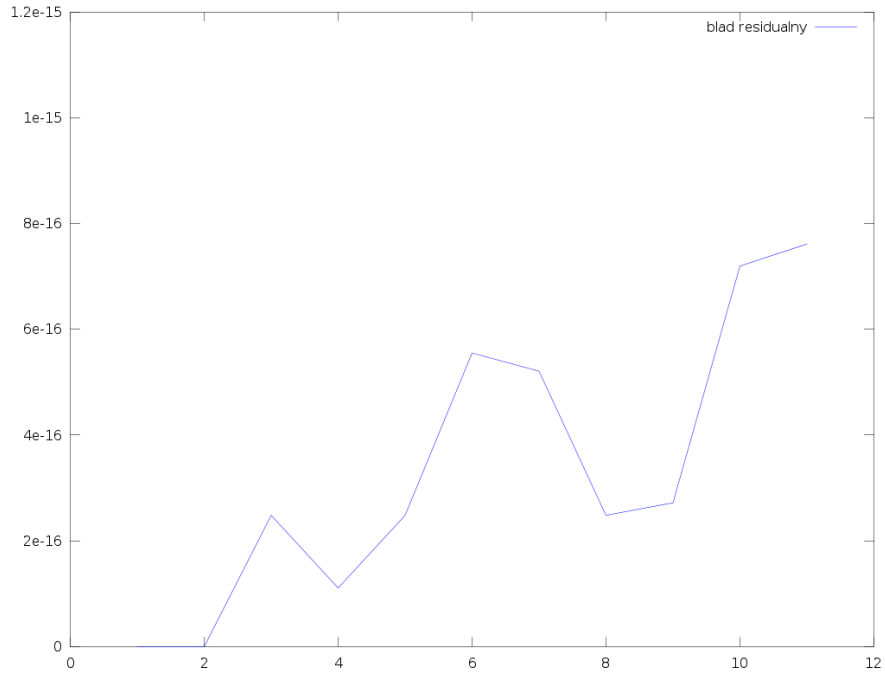
8.3 Współczynnik uwarunkowania macierzy



Rysunek 8.1: Wykres współczynnika uwarunkowania macierzy Hilberta w skali półlogarytmicznej.

Współczynnik uwarunkowania macierzy w danej normie definiujemy jako:

$$\mathrm{cond}(A, \|\cdot\|) = \|A\|\|A^{-1}\|.$$



Rysunek 8.2: Wykres błędu w normie drugiej pomiędzy rozwiązaniem dokładnym układu równań z macierzą Hilberta, a rozwiązaniem otrzymanym przez operator backslash octave'a.

Jak wiadomo, por. [11], jeśli macierz jest źle uwarunkowana, tzn. jeśli ten współczynnik jest duży względem dokładności arytmetyki, to rozwiązywanie układu równań liniowych z macierzą A może być obarczone dużym błędem.

Znaną źle uwarunkowaną macierzą jest tzw. macierz Hilberta:

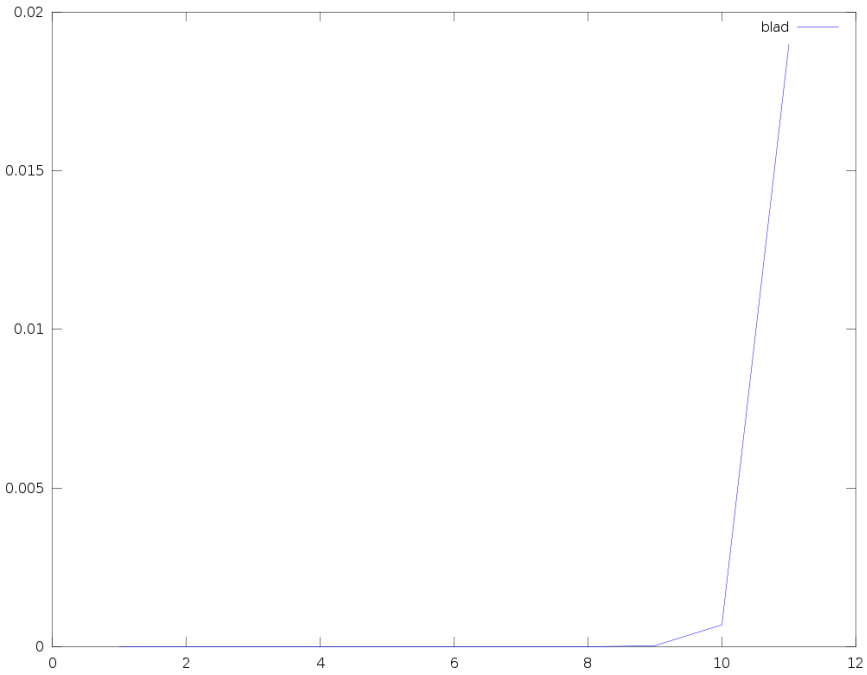
$$H_N = \left(\frac{1}{i+j-1} \right)_{i,j=1}^{N}$$

Jest to macierz symetryczna i dodatnio określona.

W octave'ie funkcja H=**hilb**(N) tworzy macierz Hilberta tylko z zamienioną kolejnością wierszy i kolumn (ale permutowanie wierszy i kolumn nie zmienia uwarunkowania macierzy).

Sprawdźmy jej współczynnik uwarunkowania w normie drugiej:

```
M=25;
c=zeros(M,1);
```



Rysunek 8.3: Wykres błędu w normie drugiej pomiędzy rozwiązaniem dokładnym układu równań z macierzą Hilberta, a rozwiązaniem otrzymanym przez operator backslash octave'a.

```
for  j=1:M,
  c(j)=cond(hilb(j));
endfor
semilogy(c,";wsp uwar macierzy Hilberta;")
```

Uwarunkowanie wzrasta wykładniczo - jak widać na wykresie w skali półlogarytmicznej, por. rysunek 8.1.

Przetestujmy, jak działa operator octave'a \ zastosowany do rozwiązywania układu równań liniowych z macierzą Hilberta.

```
M=11;
er=err=zeros(M,1);
for  j=1:M,
  H=hilb(j);
  s=ones(j,1);
  f=H*s;
  x=H\f;
```

```
  er(j)=norm(x-s,2);
  err(j)=norm(H*x-f,2);
endfor
plot(1:M,er,";blad;")
plot(1:M,err,";blad residualny;")
```

Błąd residualny, tzn. $\|H * x - f\|_2$ jest na poziomie błędu zaokrągleń, por. rysunek 8.2, ale błąd w normie drugiej gwałtownie rośnie dla $N > 9$, por. rysunek 8.3. Dla wymiaru $N \geq 12$ operator octave'a zwraca ostrzeżenie, że macierz jest numerycznie osobliwa, i że nie da się rozwiązać tego układu równań. Octave wtedy rozwiązuje to równanie w sensie LZNK, por. Rozdział 9.

Proszę zauważyć, że w praktyce nie znamy rozwiązania, więc możemy obliczyć tylko błąd residualny, tzn. normę $H*x-f$. Norma błędu residualnego może być na poziomie błędu arytmetyki zmiennopozycyjnej, podczas gdy realny błąd może być o kilka rzędów, czy nawet kilkanaście rzędów wielkości większy.

8.4 Rozkłady macierzy

Podstawowymi algorytmami bezpośrednimi rozwiązywania układu (8.1) są tzw. rozkłady macierzy: rozkład LU, rozkład L^TL - o ile A jest symetryczna, dodatnio określona i rozkład QR. Tutaj L oznacza macierz dolnotrójkątną, U, R - macierze górnotrójkątne, Q - macierz ortogonalną ($Q^TQ = QQ^T = I$).

8.4.1 Rozkład LU

Podstawowym algorytmem jest tzw. eliminacja Gaussa, która dla dowolnej macierzy nieosobliwej zwraca - w wersji z częściowym wyborem elementu głównego (czyli inaczej z częściowym osiowaniem) - następujący rozkład typu LU:

$$PA = LU$$

gdzie P to macierz permutacji, U to macierz nieosobliwa górnotrójkątna, L to macierz dolnotrójkątna z jedynkami na diagonali.

Znając ten rozkład możemy rozwiązać układ $Ax = b$ w trzech krokach:

1. Policz $f = Pb$
2. Rozwiąż układ z macierzą dolnotrójkątną $Ly = f$ ($PAx = LUx = Pb$)
3. Rozwiąż układ z macierzą górnotrójkątną $Ux = y$

Możemy też obliczyć wyznacznik A korzystając z tego, że $\det(A) = \det(P)\det(U) = \det(P)\,\Pi_k u_{kk}$. Wyznacznik P wynosi jeden albo minus jeden, w zależności od tego, czy permutacja jest parzysta, czy nieparzysta. Jeśli macierz A jest symetryczna i dodatnio określona, tzn.

$$A = A^T \qquad x^T A x > 0 \quad x \neq 0$$

to algorytm Choleskiego zwraca macierz dolnotrójkątną L taką, że

$$A = LL^T.$$

Z kolei jeśli macierz $n \times n$ A jest nieosobliwa, to istnieje rozkład QR tej macierzy, tzn. istnieją macierze tego samego wymiaru co A: ortogonalna Q (tzn. $Q^T Q = I$) oraz macierz nieosobliwa górnotrójkątna R takie, że

$$A = QR.$$

Aby rozwiązać $Ax = b$ wykorzystując ten rozkład należy rozwiązać równoważny układ:

$$Rx = Q^T b$$

z macierzą górnotrójkątną.

Do znalezienia rozkładu LU macierzy służy funkcja octave'a:

```
[L,U,P]=lu(A);
```

Argumentem jest macierz kwadratowa A, a funkcja zwraca L, U, P - odpowiednie macierze rozkładu LU.

Przetestujmy te rozkłady na kilku prostych macierzach. Na początek rozpatrzmy macierz symetryczną i znajdźmy jej rozkład LU:

```
 A=[3  1;1  3]
[L,U,P]=lu(A)
 L*U-P*A
norm(L*U-P*A,1)
```

Wynik jest prawidłowy. Zastosujmy do rozwiązania układu równań liniowych z macierzą A i znanym rozwiązaniem:

```
 s=[1;-2];
 b=A*s;
 f=P*b;
 y=L\f;
 x=U\y
 x-s
 norm(x-s,1)
```

```
xx=A\f
xx-s
norm(xx-s,1)
```

Do znalezienia rozkładu Choleskiego $A = LL^T$ macierzy symetrycznej dodatnio określonej służy funkcja octave'a R=**chol**(A), której argumentem jest macierz A, a funkcja zwraca macierz górnotrójkątną R taką, że $R^T * R = A$, czyli R to transponowana macierz L z naszego rozkładu Choleskiego.

Porównajmy rozkład Choleskiego tej samej macierzy co w poprzednim przypadku.

```
R = chol (A)
Er=R'*R-A
norm(Er,1)
```

Przetestujmy ten rozkład do znalezienia rozwiązania równania.

```
y=R'\b;
xc=R\y
xc-s
norm(xc-s,1)
```

Otrzymaliśmy poprawne wyniki.

8.4.2 Rozkład QR

Ostatni typ rozkładu macierzy, służący rozwiązaniu układu równań liniowych, to rozkład QR. Do otrzymania takiego rozkładu w octave'ie służy funkcja:

```
[Q,R]=qr(A)
```

Argumentem jest macierz A, a zwracanymi wartościami macierze rozkładu.

Przetestujmy tę funkcję na przykładzie tej samej macierzy A, co w powyższym przykładzie:

```
[Q,R]= qr (A)
Er=Q*R-A
norm(Er,1)
norm(Q*Q'-eye(size(A)),1)
```

Zastosujmy ten rozkład do rozwiązania układu równań liniowych $Ax = b$.

```
f=Q'*b;
xq=R\f
xq-s
norm(xq-s,1)
```

Metoda, jak widać, działa.

Przetestujmy rozkłady QR i LU dla macierzy nieosobliwej niesymetrycznej, lub symetrycznej ale nieokreślonej. Rozpatrzmy macierz niesymetryczną nieosobliwą i jej rozkład LU:

```
 A=[1  1;2  1]
[L,U,P]=lu(A)
 L*U-P*A
norm(L*U-P*A,1)
```

Przetestujmy rozkład QR:

```
 [Q,R]= qr (A)
 Er=Q*R-A
 norm(Er,1)
 norm(Q*Q'-eye(size(A)),1)
 norm(Q'*Q-eye(size(A)),1)
```

Co się stanie jeśli zastosujemy metodę Choleskiego?

```
 clear R
 R = chol (A)
```

Powtórzymy to samo dla macierzy symetrycznej ale nieokreślonej:

Najpierw metoda Choleskiego pozwoli nam sprawdzić, czy macierz rzeczywiście nie jest określona:

```
 A=[-4  1; 1  5]
 R = chol (A)
```

Czyli macierz nie jest dodatnio określona. Zauważmy, że algorytm Choleskiego jest najprostszą obliczeniowo metodą sprawdzenia, czy macierz jest dodatni określona.

Teraz znajdźmy rozkład LU macierzy A:

```
 [L,U,P]=lu(A)
 L*U-P*A
norm(L*U-P*A,1)
```

a następnie rozkład QR:

```
 [Q,R]= qr (A)
 Er=Q*R-A
 norm(Er,1)
 norm(Q*Q'-eye(size(A)),1)
 norm(Q'*Q-eye(size(A)),1)
```

Rozpatrzmy teraz przykład źle uwarunkowanej macierzy Hilberta dla większego $N = 5, 15$. Najpierw przypadek $N = 5$:

```
N=5;
H=hilb(N);
R = chol (H);
norm(R'*R-H,1)
[L,U,P]=lu(H)
norm(L*U-P*H,1)
[Q,R]= qr (H);
norm(Q*R-H,1)
norm(Q*Q'-eye(size(H)),1)
norm(Q'*Q-eye(size(H)),1)
```

Spróbujmy wykonać te same obliczenia dla $N = 20$:

```
N=20;
H=hilb(N);
R = chol (H);
```

Czy inne rozkłady można przeprowadzić?

```
[L,U,P]=lu(H);
norm(L*U-P*H,1)
[Q,R]= qr (H);
norm(Q*R-H,1)
norm(Q*Q'-eye(size(H)),1)
norm(Q'*Q-eye(size(H)),1)
```

Spróbujmy rozwiązać $Ax = b$ za pomocą tych rozkładów:

```
s= rand(N,1);
f=H*s;
y=L\(P*f);
x=U\y;
norm(H*x-f,1)
norm(x-s,1)
```

Niestety wynik jest bardzo zły. A co z wynikiem uzyskanym przy pomocy rozkładu QR?

```
xx=R\(Q'*f);
norm(H*xx-f,1)
norm(xx-s,1)
```

Uwarunkowanie macierzy tego układu jest bardzo duże. Żaden algorytm bezpośredni rozwiązywania układów równań liniowych, przy takiej dokładności obliczeń w arytmetyce zmiennopozycyjnej podwójnej precyzji, nie jest w stanie znaleźć rozwiązania.