

1. Wykorzystując wzory Cramera znajdź rozwiązania układów równań:

a)

$$x_1 - x_2 + 5x_3 = 4$$

$$2x_1 - 2x_2 + 3x_3 = 1$$

$$2x_1 + x_2 + x_3 = 5$$

b)

$$2x_1 - x_3 = -1$$

$$3x_1 + 2x_2 = 4$$

$$4x_2 + 3x_3 = 6$$

c)

$$x_1 + x_2 + x_3 = -2$$

$$x_1 - x_2 + x_3 = 2$$

$$-x_1 + x_2 - x_3 = -2$$

d)

$$x_1 + 2x_2 + 3x_3 = 6$$

$$2x_1 + 4x_2 + 6x_3 = 12$$

$$3x_1 + 6x_2 + 9x_3 = 1$$

Rozwiązanie:

Obliczamy wyznacznik macierzy układu równań W a potem wyznaczniki W_1 , W_2 i W_3 .

a)

$$W = \begin{vmatrix} 1 & -1 & 5 \\ 2 & -2 & 3 \\ 2 & 1 & 1 \end{vmatrix} = 21 \quad W \neq 0, \text{ wobec czego układ jest oznaczony}$$

$$W_1 = \begin{vmatrix} 4 & -1 & 5 \\ 1 & -2 & 3 \\ 5 & 1 & 1 \end{vmatrix} = 21 \quad W_2 = \begin{vmatrix} 1 & 4 & 5 \\ 2 & 1 & 3 \\ 2 & 5 & 1 \end{vmatrix} = 42 \quad W_3 = \begin{vmatrix} 1 & -1 & 4 \\ 2 & -2 & 1 \\ 2 & 1 & 5 \end{vmatrix} = 21$$

i mamy

$$x_1 = \frac{W_1}{W} = 1, \quad x_2 = \frac{W_2}{W} = 2, \quad x_3 = \frac{W_3}{W} = 1.$$

b)

$$W = \begin{vmatrix} 2 & 0 & -1 \\ 3 & 2 & 0 \\ 0 & 4 & 3 \end{vmatrix} = 0 \quad W_1 = \begin{vmatrix} -1 & 0 & -1 \\ 4 & 2 & 0 \\ 6 & 4 & 3 \end{vmatrix} = -10$$

Ponieważ wyznacznik główny jest równy 0 a wyznacznik $W_1 \neq 0$, układ jest sprzeczny i nie ma potrzeby sprawdzania kolejnych wyznaczników.

c) macierz współczynników naszego układu równań to

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix}.$$

Ponieważ jej 1-sza i 3-cia kolumna są identyczne, wyznacznik $W = 0$.

Zauważmy, że wektor wyrazów wolnych jest proporcjonalny do drugiej kolumny. Zatem jeśli w macierzy znajdzie się 1 i 3 kolumna macierzy współczynników lub 2 kolumna oraz wektor wyrazów wolnych, to wyznacznik będzie równy zero.

Zatem $W = W_1 = W_2 = W_3 = 0$.

Rząd macierzy uzupełnionej U musi być wobec tego < 3 .

Zauważmy, że po odrzuceniu 3-go wiersza i 3-ciej kolumny dostajemy macierz o niezerowym wyznaczniku. Zatem rząd macierzy współczynników $\text{rank}(\mathbf{A}) = 2$, wobec czego także $\text{rank}(\mathbf{U}) = 2$ i układ jest nieoznaczony a rozwiązanie da się wyrazić przez $3 - 2 = 1$ parametr.

Z układu odrzucamy równanie, które nie weszło do wyznacznika określającego rząd macierzy. Trzecia kolumna macierzy nie została użyta do obliczenia wyznacznika, więc odpowiadającą jej zmienną (x_3) traktujemy jako parametr. Mamy więc układ:

$$x_1 + x_2 = -2 - x_3$$

$$x_1 - x_2 = 2 - x_3$$

Obliczamy teraz wyznaczniki:

$$W = \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix} = -2, \quad W_1 = \begin{vmatrix} -2 - x_3 & 1 \\ 2 - x_3 & -1 \end{vmatrix} = 2x_3, \quad W_2 = \begin{vmatrix} 1 & -2 - x_3 \\ 1 & 2 - x_3 \end{vmatrix} = 4$$

i mamy

$$x_1 = -x_3, \quad x_2 = -2 \quad \text{a } x_3 \text{ możemy wybrać dowolnie.}$$

d)

Macierz współczynników układu ma postać

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix}$$

Ponieważ wszystkie jej kolumny są proporcjonalne do siebie i to samo zachodzi dla wierszy mamy $W = W_1 = W_2 = W_3 = 0$.

Z tego samego powodu $\text{rank}(\mathbf{A}) = 1$.

Zauważmy jednak, że z macierzy uzupełnionej możemy wybrać niezerowy wyznacznik 2×2 , np.

$$\begin{vmatrix} 4 & 12 \\ 6 & 1 \end{vmatrix},$$

czyli $\text{rank}(\mathbf{U}) = 2 > \text{rank}(\mathbf{A})$. Układ jest wobec tego sprzeczny.

Przykłady c) i d) pokazują, że jeśli wyznacznik główny i wszystkie wyznaczniki niewiadomych są równe 0, to układ może być nieoznaczony lub sprzeczny.

2. Badając macierze i wykorzystując wzory Cramera znajdź rozwiązanie układu jednorodnego

$$3x_1 + 2x_2 - x_3 = 0$$

$$x_1 + 3x_2 - 4x_3 = 0.$$

$$x_1 - 4x_2 + 7x_3 = 0$$

Rozwiązanie.

Wyznacznik macierzy współczynników układu $W = \det(\mathbf{A}) = 0$, wobec czego układ ma nietrywialne rozwiązanie.

Zauważmy, że wyznacznik powstały po skreśleniu ostatniego wiersza i ostatniej kolumny jest różny od zera. Odrzucamy wobec tego ostatnie równanie a zmienną x_3 traktujemy jak parametr i otrzymujemy układ

$$3x_1 + 2x_2 = x_3$$

$$x_1 + 3x_2 = 4x_3$$

$$W = \begin{vmatrix} 3 & 2 \\ 1 & 3 \end{vmatrix} = 7, \quad W_1 = \begin{vmatrix} x_3 & 2 \\ 4x_3 & 3 \end{vmatrix} = -5x_3, \quad W_2 = \begin{vmatrix} 3 & x_3 \\ 1 & 4x_3 \end{vmatrix} = 11x_3$$

Zatem

$$x_1 = -\frac{5}{7}x_3, \quad x_2 = \frac{11}{7}x_3 \quad \text{a } x_3 \text{ jest dowolne.}$$

W szczególności dla $x_3 = 0$ dostajemy rozwiązanie trywialne.

3. Rozwiąż układy z zadania 1 wykorzystując metodę eliminacji Gaussa.

Rozwiązanie.

Będziemy przeprowadzać operacje na macierzy uzupełnionej (co odpowiada działaniom na równaniach). Dla przejrzystości kolumnę wyrazów wolnych oddzielimy pionową linią. Równoważność macierzy będziemy zaznaczać znakiem $\sim$. Nasze działania będą miały na celu doprowadzenie macierzy uzupełnionej do postaci schodkowej.

Zwróćmy uwagę, że obliczenia prowadzimy dokładnie, zatem w odróżnieniu od obliczeń numerycznych wybór elementu głównego nie wpływa na dokładność wyniku. Wygodnie natomiast będzie wybierać element równy 1 lub -1 .

a) Macierz uzupełniona układu ma postać

$$\left[\begin{array}{ccc|c} 1 & -1 & 5 & 4 \\ 2 & -2 & 3 & 1 \\ 2 & 1 & 1 & 5 \end{array} \right]$$

Odejmując od drugiego i trzeciego wiersza pierwszy wiersz pomnożony przez 2 dostajemy równoważną macierz

$$\left[\begin{array}{ccc|c} 1 & -1 & 5 & 4 \\ 0 & 0 & -7 & -7 \\ 0 & 3 & -9 & -3 \end{array} \right]$$

W tym momencie współczynnik przy x_2 w drugim równaniu jest równy 0, więc elementu głównego szukamy w innym wierszu. Zamieniamy wiersze 2 i 3 otrzymując

$$\left[\begin{array}{ccc|c} 1 & -1 & 5 & 4 \\ 0 & 3 & -9 & -3 \\ 0 & 0 & -7 & -7 \end{array} \right]$$

i widać, że mamy już macierz w postaci schodkowej. Liczba schodków w macierzy wynosi 3, więc $\text{rank}(\mathbf{A}) = 3$ i układ jest oznaczony.

Trzeci wiersz macierzy oznacza równość

$$-7x_3 = -7$$

skąd mamy $x_3 = 1$. Podstawiając wstecz do wiersza 2 mamy

$$3x_2 - 9 = -3, \text{ skąd } x_2 = 2$$

Wstawiając do wiersza 1 mamy

$$x_1 - 2 + 5 = 4, \text{ czyli } x_1 = 1.$$

Rozwiązanie jest identyczne, jak otrzymane w zadaniu 1 z wzorów Cramera.

Zauważmy, że łatwo teraz obliczyć wyznacznik przekształconej macierzy równy -21 . Ponieważ po drodze dokonaliśmy jednego przestawienia wierszy, wyznacznik macierzy $\mathbf{A}$ ma przeciwny znak i jest równy 21 (co obliczyliśmy wcześniej w zadaniu 1).

Mnożniki użyte do eliminacji x_1 z 2-go i 3-go wiersza wynosiły 2, x_2 z trzeciego już eliminować nie musieliśmy (przypadkiem wyszedł współczynnik 0), więc odpowiedni mnożnik jest równy 0.

Mamy zatem rozkład $\mathbf{LU}$ macierzy współczynników:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} 1 & -1 & 5 \\ 0 & 3 & -9 \\ 0 & 0 & -7 \end{bmatrix}.$$

Obliczając iloczyn możemy sprawdzić, że

$$\mathbf{LU} = \begin{bmatrix} 1 & -1 & 5 \\ 2 & 1 & 1 \\ 2 & -2 & 3 \end{bmatrix},$$

czyli dostaliśmy macierz $\mathbf{A}$ z przestawionymi wierszami 2-gim i 3-cim.

b)

W trakcie eliminacji przekształcamy macierz uzupełnioną układu do postaci schodkowej

$$\left[\begin{array}{ccc|c} 2 & 0 & -1 & -1 \\ 3 & 2 & 0 & 4 \\ 0 & 4 & 3 & 6 \end{array} \right] \sim \left[\begin{array}{ccc|c} 2 & 0 & -1 & -1 \\ 0 & 2 & \frac{3}{2} & \frac{11}{2} \\ 0 & 4 & 3 & 6 \end{array} \right] \sim \left[\begin{array}{ccc|c} 2 & 0 & -1 & -1 \\ 0 & 2 & \frac{3}{2} & \frac{11}{2} \\ 0 & 0 & 0 & -\frac{10}{2} \end{array} \right]$$

W kolejnych etapach wykonaliśmy następujące operacje:

- od wiersza 2 odjęliśmy wiersz 1 pomnożony przez $3/2$
- od wiersza 3 odjęliśmy wiersz 2 pomnożony przez 2

Ostatni wiersz macierzy $\mathbf{A}$ ma same zera, zatem $\text{rank}(\mathbf{A}) = 0$. Ale macierz $\mathbf{U}$ ma o jeden schodek więcej, zatem $\text{rank}(\mathbf{U}) = 3 > \text{rank}(\mathbf{A})$, czyli na mocy twierdzenia Kroneckera-Capellego układ jest sprzeczny.

Widać to z ostatniego wiersza oznaczającego

$$0 = -10/2, \text{ czyli sprzeczność.}$$

c)

Mamy

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & -2 \\ 1 & -1 & 1 & 2 \\ -1 & 1 & -1 & -2 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & 1 & 1 & -2 \\ 0 & -2 & 0 & 4 \\ 0 & 2 & 0 & -4 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & 1 & 1 & -2 \\ 0 & -2 & 0 & 4 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

Kolejne równoważne macierze otrzymaliśmy

– odejmując wiersz 1 od wiersza 1 i dodając wiersz 1 do wiersza 3

– dodając wiersz 2 do wiersza 3

Z ostatniego, zerowego wiersza wynika, że $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{U}) = 2$, zatem układ ma rozwiązanie.

Przyjmując x_3 za parametr i podstawiając wstecz do równania 2 mamy

$$-2x_2 = 4 \text{ skąd } x_2 = -2$$

Wstawiając do 1 dostajemy

$$x_1 - 2 + x_3 = -2, \text{ czyli } x_1 = -x_3.$$

d)

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 6 \\ 2 & 4 & 6 & 12 \\ 3 & 6 & 9 & 1 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 3 & 6 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -17 \end{array} \right]$$

(od 2 wiersza odjęliśmy wiersz 1 mnożony przez 2 a od 3-go wiersz 1 mnożony przez 3).

Jak widać, $\text{rank}(\mathbf{A}) = 1 < \text{rank}(\mathbf{U}) = 2$, czyli układ jest sprzeczny, co wynika z ostatniego wiersza oznaczającego

$$0 = -17.$$

4. Rozwiąż zadanie 2 metodą eliminacji Gaussa.

Rozwiązanie.

Mamy

$$\left[\begin{array}{ccc|c} 3 & 2 & -1 & 0 \\ 1 & 3 & -4 & 0 \\ 1 & -4 & 7 & 0 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & 3 & -4 & 0 \\ 3 & 2 & -1 & 0 \\ 1 & -4 & 7 & 0 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & 3 & -4 & 0 \\ 0 & -7 & 11 & 0 \\ 0 & -7 & 11 & 0 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & 3 & -4 & 0 \\ 0 & -7 & 11 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

– zamieniliśmy wiersz 1 z 2 (dla wygody)

– od wiersza 2 odjęliśmy wiersz 1 mnożony przez 3, od wiersza 3 odjęliśmy wiersz 1

– odjęliśmy wiersz 2 od wiersza 3

$\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{U}) = 2 = 3-1$, zatem rozwiązanie będzie miało 1 parametr.

Z wiersza 2 mamy $-7x_2 + 11x_3 = 0$, czyli $x_2 = \frac{11}{7}x_3$.

Wstawiając do równania 1 mamy

$$x_1 + \frac{33}{7}x_3 - 4x_3 = 0, \text{ skąd } x_1 = -\frac{5}{7}x_3.$$

5. Rozwiąż nadokreślone układy równań liniowych

a)

$$5x_1 + 3x_2 - x_3 = 3$$

$$2x_1 + x_2 - x_3 = 1$$

$$3x_1 - 2x_2 + 2x_3 = -4$$

$$x_1 - x_2 + 2x_3 = -2$$

b)

$$4x_1 - x_2 = 7$$

$$3x_1 + x_2 = 14$$

$$2x_1 + 3x_2 = 0$$

Rozwiązanie.

Stosując metodę eliminacji mamy:

a)

$$\left[\begin{array}{ccc|c} 5 & 3 & -1 & 3 \\ 2 & 1 & -1 & 1 \\ 3 & -2 & 2 & -4 \\ 1 & -1 & 2 & -2 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & -1 & 2 & -2 \\ 2 & 1 & -1 & 1 \\ 3 & -2 & 2 & -4 \\ 5 & 3 & -1 & 3 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & -1 & 2 & -2 \\ 0 & 3 & -5 & 5 \\ 0 & 1 & -4 & 2 \\ 0 & 8 & -11 & 13 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & -1 & 2 & -2 \\ 0 & 1 & -4 & 2 \\ 0 & 3 & -5 & 5 \\ 0 & 8 & -11 & 13 \end{array} \right]$$

$$\sim \left[\begin{array}{ccc|c} 1 & -1 & 2 & -2 \\ 0 & 1 & -4 & 2 \\ 0 & 0 & 7 & -1 \\ 0 & 0 & 21 & -3 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & -1 & 2 & -2 \\ 0 & 1 & -4 & 2 \\ 0 & 0 & 7 & -1 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

– przestawiliśmy równanie 1 z 4

– odjęliśmy równanie 1: pomnożone przez 2 od r. 2, pomnożone przez 3 od r. 3 i pomnożone przez 5 od r. 4

– przestawiliśmy równanie 2 z 3

– odjęliśmy równanie 2: pomnożone przez 3 od r. 3 i pomnożone przez 8 od r. 4

– od równania 4 odjęliśmy równanie 3 pomnożone przez 3

Jak widać, $\text{rank}(\mathbf{U}) = \text{rank}(\mathbf{A}) = 3$, zatem układ ma rozwiązanie.

Równanie 4 możemy odrzucić.

Z równania 3 mamy: $7x_3 = -1$, skąd $x_3 = -\frac{1}{7}$.

Wstawiając do r. 2 mamy: $x_2 + \frac{4}{7} = 2$, skąd $x_2 = \frac{10}{7}$.

Podstawiając do r. 1 otrzymujemy: $x_1 - \frac{10}{7} - \frac{2}{7} = -2$, skąd $x_1 = -\frac{2}{7}$.

b)

$$\left[\begin{array}{cc|c} 4 & -1 & 7 \\ 3 & 1 & 14 \\ 2 & 3 & 0 \end{array} \right] \sim \left[\begin{array}{cc|c} 2 & 3 & 0 \\ 3 & 1 & 14 \\ 4 & -1 & 7 \end{array} \right] \sim \left[\begin{array}{cc|c} 2 & 3 & 0 \\ 0 & -\frac{7}{2} & 14 \\ 0 & -7 & 7 \end{array} \right] \sim \left[\begin{array}{cc|c} 2 & 3 & 0 \\ 0 & -\frac{7}{2} & 14 \\ 0 & 0 & 21 \end{array} \right]$$

– przestawiliśmy r. 1 z 3

– odjęliśmy r.1: pomnożone przez 3/2 od r.2 i pomnożone przez 2 od r. 3

– odjęliśmy r. 2 pomnożone przez 2 od r. 3

Widać, że $\text{rank}(\mathbf{U}) = 3 > \text{rank}(\mathbf{A}) = 2$, zatem układ jest sprzeczny.
Ostatni wiersz oznacza sprzeczność $0 = 21$.

Warto tu uczynić pewną dygresję. Otóż w praktycznych zastosowaniach często nie poszukuje się tego typu rozwiązań („klasycznych”) dla układów nadokreślonych. Układy takie pojawiają się, np. gdy modelujemy pewne zjawisko liniowo zależne od n zmiennych (z pewnymi współczynnikami). Jeśli przeprowadzimy m pomiarów badanej wielkości zmieniając zmienne, to otrzymamy układ m równań liniowych z n niewiadomymi, który możemy rozwiązać wyznaczając współczynniki w badanym modelu. Ponieważ jednak zarówno wartości zmiennych jak i wyniki pomiaru obarczone są błędami pomiarowymi, otrzymany układ równań na ogół będzie sprzeczny. Zamiast poszukiwać jego klasycznego rozwiązania, postępuje się inaczej: szuka się wektora, który jak najlepiej przybliży to rozwiązanie, tzn. takiego $\mathbf{x}$, że norma wektora $\mathbf{Ax} - \mathbf{b}$ osiąga minimum.

Pobierz ze strony:

<http://www.chemia.uj.edu.pl/~eilmes/mnumch/>
plik prog2.zip

Rozpakuj plik poleceniem
unzip prog2

W wyniku otrzymasz katalog mnumch2 z przykładowymi programami.

Skompiluj programy poleceniami:

```
gfortran -o ulin.x ulin.f -llapack
```

```
gfortran -o ul-chol.x ul-chol.f -llapack
```

Programy te służą do rozwiązywania układów równań liniowych n równań z n niewiadomymi. Można równocześnie rozwiązać wiele układów różniących się prawymi stronami.

Program ulin.x wykorzystuje rozkład LU. Dane należy umieszczać w pliku ulin.inp w następującej kolejności:

liczba_równań, liczba_prawych_stron

macierz współczynników układu zapisana wierszami

macierz kolumnowych wektorów prawych stron

Wyniki zapisane zostają do pliku ulin.out. W pliku tym pojawiają się rozwiązania układu, rozkład $\mathbf{A} = \mathbf{PLU}$, wyznacznik macierzy $\mathbf{A}$ oraz informacje o przestawieniach wierszy.

Program ul-chol.x służy do rozwiązywania układów z symetryczną macierzą dodatnio określoną przez rozkład Cholesky’ego. Dane należy umieszczać w pliku ul-chol.inp, jak dla programu ulin.x, z tą różnicą, że wystarczy zapisać dolny trójkąt macierzy. Wyniki zapisane zostają do pliku ul-chol.out

6. Wykorzystując program `ulin.x` rozwiąż

a) przypadek a) z zadania 1

b) układ

$$5x_1 + 3x_2 + 4x_3 = -18$$

$$3x_1 + x_3 = -7$$

$$6x_1 + 3x_2 + 6x_3 = -27$$

Rozwiązanie:

a) dane w pliku `ulin.inp` powinny wyglądać następująco:

3 1

1 -1 5

2 -2 3

2 1 1

4

1

5

W pliku wynikowym dostajemy znane już nam rozwiązanie oraz informację o rozkładzie macierzy (czynniki nie będą takie same, jak w zadaniu 3a), ze względu na inne przestawienia wierszy. Przestawienia wierszy macierzy **A** opisane są kolejnymi wartościami zmiennej `ipiv`: jeśli *i*-ta wartość jest > od *i*, to *i*-ty wiersz został zamieniony z wierszem o podanym numerze. Np. wartości 2,3,3 oznaczają

2 pierwszy wiersz zamieniono z drugim

3 drugi wiersz zamieniono z trzecim

3 trzeci nie był przestawiany

b) Rozwiązaniem jest $x_1 = -1$, $x_2 = 1$, $x_3 = -4$.

7. Wykorzystując program `ulin.x` oblicz:

a) wyznacznik macierzy

$$\begin{bmatrix} 2 & 1 & 2 & 3 & 2 \\ 3 & -2 & 7 & 5 & -1 \\ 3 & -1 & -5 & -3 & -2 \\ 5 & -6 & 4 & 2 & -4 \\ 2 & -3 & 3 & 1 & -2 \end{bmatrix}$$

b) macierz odwrotną do

$$\begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 4 & 1 & 2 \end{bmatrix}$$

Rozwiązanie:

a) Do obliczenia wyznacznika wystarczy nam rozkład LU macierzy. Można zatem programowi podać naszą macierz i dowolny wektor prawych stron (w praktyce napisalibyśmy

do tego osobny program, który zatrzymałby się na etapie rozkładu macierzy). Wyznacznik obliczymy jako iloczyn elementów diagonalnych U ze znakiem wynikającym z informacji o przestawieniach wierszy zapisanej w zmiennej `ipiv`. Wynikiem jest 98.

b) Macierz odwrotną otrzymamy ustawiając kolumnami rozwiązania 3 układów równań z prawymi stronami tworzącymi macierz jednostkową stopnia 3. Dane będą wobec tego wyglądać następująco:

3 3

2 1 2
1 2 3
4 1 2

1 0 0
0 1 0
0 0 1

W wyniku otrzymamy

$$A^{-1} = \begin{bmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \\ -5 & 2 & 2 \\ \frac{7}{2} & -1 & -\frac{3}{2} \end{bmatrix}.$$

8. Wykorzystując program `ul-chol.x` rozwiąż układ

$$4x_1 + 2x_2 + 4x_3 + 3x_4 = 2$$

$$2x_1 + 2x_2 + x_3 + 2x_4 = 2$$

$$4x_1 + x_2 + 6x_3 + 2x_4 = 1$$

$$3x_1 + 2x_2 + 2x_3 + 5x_4 = -1$$

Sprawdź wynik programem `ulin.x`.

Rozwiązanie:

Dane do programu wyglądają następująco:

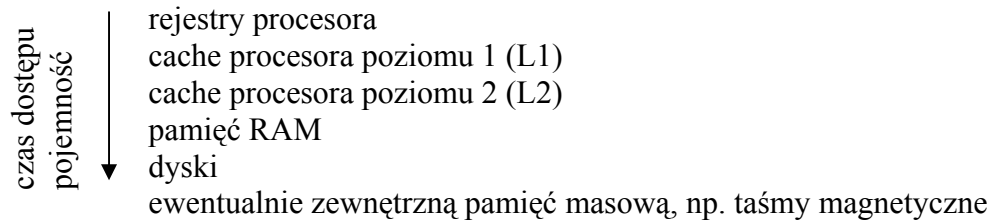
4 1

4
2 2
4 1 6
3 2 2 5

2
2
1
-1

W wyniku dostajemy rozwiązanie $x_1 = 5/3$, $x_2 = 1$, $x_3 = -2/3$, $x_4 = -4/3$. Przy okazji możemy na podstawie czynnika L stwierdzić, że wyznacznik macierzy układu wynosi 9.

9. Dla efektywnego działania programu istotna jest nie tylko redukcja liczby potrzebnych operacji. Istotna bywa też optymalizacja użycia różnych rodzajów pamięci. Wynika to z hierarchicznej struktury pamięci komputera: mamy w nim różne typy pamięci różniące się pojemnością i szybkością działania (czasem dostępu). W szczególności w komputerze typu PC mamy (zaczynając od najszybszych, ale najmniej pojemnych):



W przypadku dużych danych niezbędna bywa optymalizacja użycia dysku, dla mniejszych - poprawę wydajności umożliwia efektywne wykorzystanie cache'u.

Zastanówmy się nad następującym przykładem. Znamy wzór na obliczanie elementów iloczynu macierzy (dla uproszczenia $n \times n$):

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

Musimy oczywiście wykonać obliczenia dla wszystkich potrzebnych i i j .

Możemy próbować zaprogramować go w sposób narzucający się z postaci wzoru:

```
dla wszystkich i od 1 do n powtarzaj:
  dla wszystkich j od 1 do n powtarzaj:
    dla wszystkich k od 1 do n powtarzaj:
      dodaj do  $c_{ij}$  składnik  $a_{ik}b_{kj}$ 
    koniec powtórzeń po k
  koniec powtórzeń po j
koniec powtórzeń po i
```

Ze względu na kolejność zmian indeksów nazwijmy go schematem ijk .

Możliwa jest jednak inna kolejność zmiany indeksów, np. jki

```
dla wszystkich j od 1 do n powtarzaj:
  dla wszystkich k od 1 do n powtarzaj:
    dla wszystkich i od 1 do n powtarzaj:
      dodaj do  $c_{ij}$  składnik  $a_{ik}b_{kj}$ 
    koniec powtórzeń po i
  koniec powtórzeń po k
koniec powtórzeń po j
```

albo schemat ikj :

```
dla wszystkich i od 1 do n powtarzaj:
  dla wszystkich k od 1 do n powtarzaj:
    dla wszystkich j od 1 do n powtarzaj:
      dodaj do  $c_{ij}$  składnik  $a_{ik}b_{kj}$ 
    koniec powtórzeń po j
  koniec powtórzeń po k
koniec powtórzeń po i
```

W kartotece mnumch2 znajdują się programy wykonujące w różny sposób mnożenie macierzy rzeczywistych w pojedynczej dokładności. Skompiluj je poleceniami

```
gfortran -o mm-blas.x mm-blas.f90 -lblas
gfortran -o mm-fort.x mm-fort.f90
gfortran -o mm-ijk.x mm-ijk.f90
gfortran -o mm-jki.x mm-jki.f90
gfortran -o mm-ikj.x mm-ikj.f90
```

Programy `mm-ijk.x`, `mm-jki.x` i `mm-ikj.x` realizują mnożenie odpowiednim schematem opisanym wyżej, `mm-fort.x` wykorzystuje dostępną w Fortranie 90 funkcję `matmul` a `mm-blas.x` używa do mnożenia procedury `sgemm` z biblioteki BLAS.

W pliku `dat` podaj dane do programu:

powinny to być dwie liczby całkowite: pierwsza podaje stopień macierzy, druga liczbę powtórzeń (dla małych macierzy mnożenia trzeba powtórzyć, aby dostać mierzalne czasy). Wartości parametrów trzeba dobrać empirycznie, aby dało się rozsądnie zmierzyć czasy.

Programy uruchamiaj poleceniem

```
time ./program.x < dat
```

W wyniku działania polecenia `time` po zakończeniu działania programu wypisane zostaną trzy czasy. Nas interesuje ten oznaczony jako czas użytkownika (`user`).

Przeprowadź obliczenia dla różnych stopni macierzy zapisując czasy uzyskane w różnych programach.

Oto przykładowe wyniki. uzyskane na maszynie z procesorem Q9550 2.83GHz z 8 GB RAM i pojemnościami pamięci cache L1 4*32 KB, L2 2*6144 KB z systemem operacyjnym Ubuntu 8.10. Wykorzystano standardową bibliotekę BLAS.

W pierwszej kolumnie tabeli podano informacje o stopniu macierzy i liczbie powtórzeń; dla każdego programu podano czas wykonania w sekundach i stosunek tego czasu do najlepszego wyniku.

rozmiar zadania	BLAS		Fortran		<i>jki</i>		<i>ijk</i>		<i>ikj</i>	
	czas	<i>f</i>	czas	<i>f</i>	czas	<i>f</i>	czas	<i>f</i>	czas	<i>f</i>
100,1000	1.2	1.0	1.5	1.25	12.3	10.3	12.1	10.1	12.2	10.2
200,200	1.2	1.0	1.8	1.5	19.1	15.9	18.6	15.5	19.2	16.0
300,75	1.2	1.0	1.9	1.6	23.9	19.9	23.1	19.3	28.2	23.5
500,20	1.2	1.0	2.1	1.75	28.9	24.1	30.9	25.6	40.7	33.9
1000,5	1.7	1.0	3.9	2.3	57.4	33.8	73.9	43.5	92.6	54.5
2000,1	2.2	1.0	6.8	3.1	91.9	41.8	122.8	55.8	165.2	75.1
5000,1	28.1	1.0	100.6	3.6	1445	51.4	1991	70.9	2774	98.7

Jak widać, procedury BLASa i Fortranu biją na głowę nasze ręczne implementacje. Należy zatem przy rozwiązywaniu zadania numerycznego próbować znaleźć zoptymalizowaną bibliotekę, zamiast wywierać drzwi otwarte, z gorszym na ogół skutkiem.

Schematy *jki*, *ijk*, *ikj* dla małych danych potrzebują mniej więcej tyle samo czasu, jednak ze wzrostem stopnia macierzy schemat *jki* jest najszybszy natomiast *ikj* najwolniejszy. Ponieważ

liczba mnożeń i dodawań jest we wszystkich trzech schematach taka sama, różnica musi wynikać z kolejności wykonywania operacji, czyli odwołań do pamięci.

Aby zrozumieć, co się dzieje powinniśmy wiedzieć dwie rzeczy:

1. W trakcie wykonywania obliczeń pewien obszar pamięci można skopiować do cache'u procesora, przy czym przenoszenie do cache'u odbywa się całymi liniami (czyli oprócz aktualnie potrzebnej komórki pamięci ładujemy do cache'u także sąsiednie). Traci się na to trochę czasu. Jeśli jednak będziemy mogli teraz sięgać po dane do cache'u zamiast pamięci RAM, to strata ta zwróci się z nawiązką, bo cache jest szybszy. W momencie, gdy stwierdzimy, że potrzebnej komórki pamięci nie ma aktualnie w cache'u (*cache miss*), możemy skopiować do cache'u kolejny obszar pamięci. Oznacza to, że korzystanie z pamięci cache opłaca się gdy korzystamy często z sąsiadujących komórek pamięci (bo po skopiowaniu bloku pamięci do cache'u przez pewien czas potrzebne komórki będą w nim znajdowane). Jeśli natomiast w kolejnych krokach sięgamy do bardzo różnych rejonów pamięci, to po skopiowaniu pewnego obszaru RAM do cache'u w kolejnym kroku stwierdzimy, że potrzebnej komórki pamięci tam nie ma i musimy znów sięgać do RAM. W takiej sytuacji używanie cache'u nie przynosi nam zysku, tylko stratę.

2. W Fortanie macierze zapamiętywane są kolumnami, tzn. elementy macierzy umieszczane są w kolejnych komórkach pamięci w ten sposób, że najszybciej zmienia się pierwszy indeks. Oznacza to, że jeśli w kolejnym kroku zmienimy o 1 pierwszy indeks, to sięgamy do następnej liczby w pamięci. Jeśli zmienimy o 1 drugi indeks, to sięgamy do liczby odległej w pamięci o n liczb. W pierwszym przypadku korzystanie z cache'u może być efektywne, w drugim należy się spodziewać częstych sytuacji *cache miss*.

Popatrzmy teraz na naszą instrukcję dodawania: potrzebujemy tam odwołać się do trzech elementów tablic c_{ij} , a_{ik} oraz b_{kj} .

W schemacie *jki* (najszybszym) najszybciej zmienia się i , które występuje jako pierwszy indeks w 2 macierzach (macierze te można zatem opłacalnie wrzucać do cache'u), dodatkowo pierwszy indeks pozostałej tablicy to zmienna k , która zmienia się w drugiej kolejności.

W schemacie *ijk* najszybciej zmienia się k , które jako pierwszy indeks występuje w jednej macierzy (którą wobec tego można efektywnie cache'ować).

W schemacie *ikj* najszybciej zmienia się zmienna j , która zawsze jest drugim indeksem w macierzy, zatem którąkolwiek macierz próbowalibyśmy umieścić w cache'u, to dla dużych n zaraz okaże się, że kolejny potrzebny element w cache'u się nie znalazł.

Tłumaczy to różnice w szybkości działania dla większych n . Natomiast dla najmniejszych n całe lub prawie całe macierze mieszczą się w cache'u procesora i kolejność zmian indeksów nie odgrywa tak znaczącej roli – stąd podobne czasy działania programów.

Optymalizacja biblioteki BLAS przy użyciu ATLASA polega między innymi na podziale danych, na których wykonywane są operacje na bloki i dobraniu rozmiaru bloku, tak by najlepiej wykorzystać dostępną na danej maszynie pamięć cache.

Układy równań liniowych, wyznaczniki i macierze odwrotne w pakiecie R.

Niech **A** jest macierzą kwadratową, **B** macierzą prawych stron układów równań liniowych.

Interesujące są dla nas funkcje:

`det(A)` – wyznacznik macierzy **A**

`solve(A)` – zwraca macierz odwrotną do **A**,

ale wywołana z dwoma argumentami:

`solve(A, B)` – zwraca macierz rozwiązań $\mathbf{AX} = \mathbf{B}$

10. Rozwiąż w pakiecie R przykłady 6-8.

Rozwiązanie:

Odpowiednie polecenia tworzenia macierzy i wywołań funkcji wyglądają następująco:

Przykład 6a

Utworzenie macierzy **A**

```
(A=matrix(c(1, -1, 5, 2, -2, 3, 2, 1, 1), c(3, 3), byrow=T))  
      [,1] [,2] [,3]  
[1,]    1  -1    5  
[2,]    2  -2    3  
[3,]    2    1    1
```

Utworzenie wektora prawych stron **b**

```
(b=c(4, 1, 5))  
[1] 4 1 5
```

i znalezienie rozwiązania układu

```
solve(A, b)  
[1] 1 2 1
```

Analogicznie postępujemy w przykładzie 6b

```
(A=matrix(c(5, 3, 4, 3, 0, 1, 6, 3, 6), c(3, 3), byrow=T))  
      [,1] [,2] [,3]  
[1,]    5    3    4  
[2,]    3    0    1  
[3,]    6    3    6  
(b=c(-18, -7, -27))  
[1] -18 -7 -27  
solve(A, b)  
[1] -1  1 -4
```

Przykład 7a

Tworzymy macierz

```
(A=matrix(c(2, 1, 2, 3, 2, 3, -2, 7, 5, -1, 3, -1, -5, -3, -2, 5, -6, 4, 2, -4, 2, -3, 3, 1, -  
2), c(5, 5), byrow=T))  
      [,1] [,2] [,3] [,4] [,5]  
[1,]    2    1    2    3    2  
[2,]    3   -2    7    5   -1  
[3,]    3   -1   -5   -3   -2  
[4,]    5   -6    4    2   -4  
[5,]    2   -3    3    1   -2
```

i obliczamy wyznacznik

```
det(A)  
[1] 98
```

Przykład 7b

```
(A=matrix(c(2,1,2,1,2,3,4,1,2),c(3,3),byrow=T))
```

```
      [,1] [,2] [,3]
[1,]    2    1    2
[2,]    1    2    3
[3,]    4    1    2
```

Funkcja `solve` wywołana z macierzą kwadratową jako jedynym argumentem zwróci macierz odwrotną

```
solve(A)
      [,1] [,2] [,3]
[1,] -0.5   0   0.5
[2,] -5.0   2   2.0
[3,]  3.5  -1  -1.5
```

Przykład 8 rozwiązujemy analogicznie do przykładów 6a,b

```
(A=matrix(c(4,2,4,3,2,2,1,2,4,1,6,2,3,2,2,5),c(4,4),byrow=T))
```

```
      [,1] [,2] [,3] [,4]
[1,]    4    2    4    3
[2,]    2    2    1    2
[3,]    4    1    6    2
[4,]    3    2    2    5
```

```
(b=c(2,2,1,-1))
```

```
[1]  2  2  1 -1
```

```
solve(A,b)
```

```
[1]  1.6666667  1.0000000 -0.6666667 -1.3333333
```

RÓWNANIA NIELINIOWE.

11. Bazując na metodzie Newtona znajdowania zer funkcji wyprowadź wzór rekurencyjny na obliczanie pierwiastka kwadratowego.

Rozwiązanie.

Obliczanie pierwiastka kwadratowego $\sqrt{A}$ oznacza znajdowanie rozwiązania równania $x^2 - A = 0$.

W metodzie Newtona mając pewne przybliżenie x_n rozwiązania znajdujemy lepsze z wzoru iteracyjnego

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

W naszym przypadku $f(x) = x^2 - A$, zatem $f'(x) = 2x$. Podstawiając do powyższego wzoru mamy

$$x_{n+1} = x_n - \frac{x_n^2 - A}{2x_n}.$$

Po przekształceniu dostajemy wzór

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{A}{x_n} \right),$$

wykorzystywany w komputerowych implementacjach funkcji obliczającej pierwiastek kwadratowy.

Kolejne zadania przewidziane są do rozwiązania w pakiecie R.

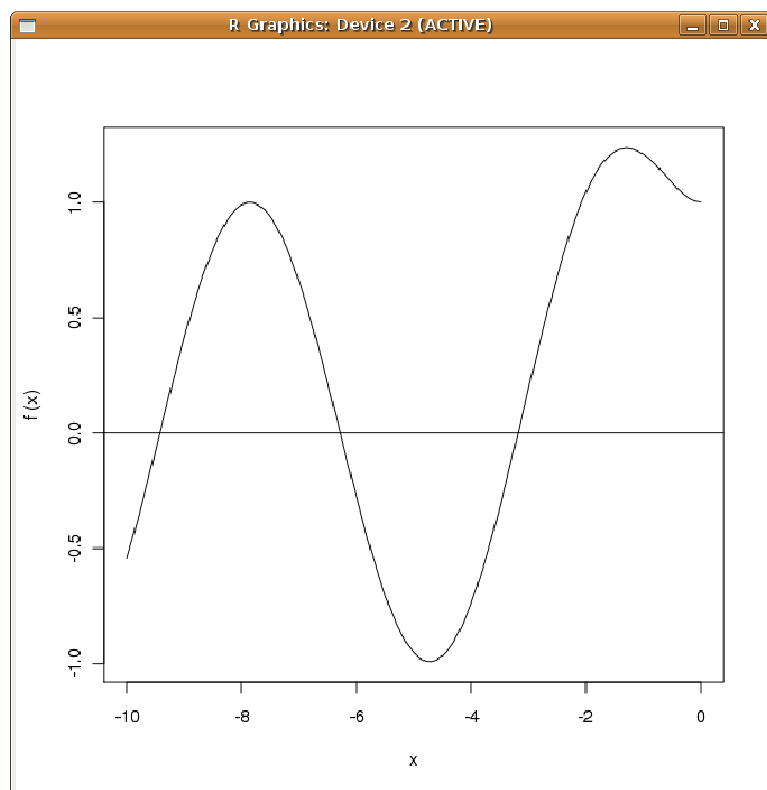
12. Znajdź najbliższe 0 rozwiązanie równania

$$\exp(x) - \sin(x) = 0$$

Rozwiązanie.

Na pewno wartość zero powyższa funkcja może przyjmować tylko dla $x < 0$. Obejrzyjmy wykres funkcji w okolicy zera:

```
f=function(x) exp(x)-sin(x)
curve(f, -10, 0)
abline(h=0)
```



Zero będzie w przedziale $[-5, 0]$, więc wydajemy polecenie:

```
uniroot(f, c(-5, 0), tol=1e-7)
```

dostając po 5 iteracjach przybliżenie zera funkcji równe -3.183063

Poleceniem

```
uniroot(f, c(-7, -5), tol=1e-7)
```

otrzymamy następne miejsce zerowe -6.281314

12. Korzystając z pakietu `animation` obejrzyj działanie metody bisekcji i Newtona. Rozwiąż równanie z poprzedniego zadania.

Rozwiązanie.

Instalacja pakietu odbywa się po wydaniu polecenia

```
install.packages()
```

należy następnie wybrać serwer i wskazać pakiet `animation`

załadowanie pakietu odbywa się przez polecenie

```
library(animation)
```

poleceniami

```
bisection.method()
```

```
newton.method()
```

można obejrzeć w działaniu przykłady dla obu metod

W razie potrzeby poleceniem

```
ani.options(interval=xxx)
```

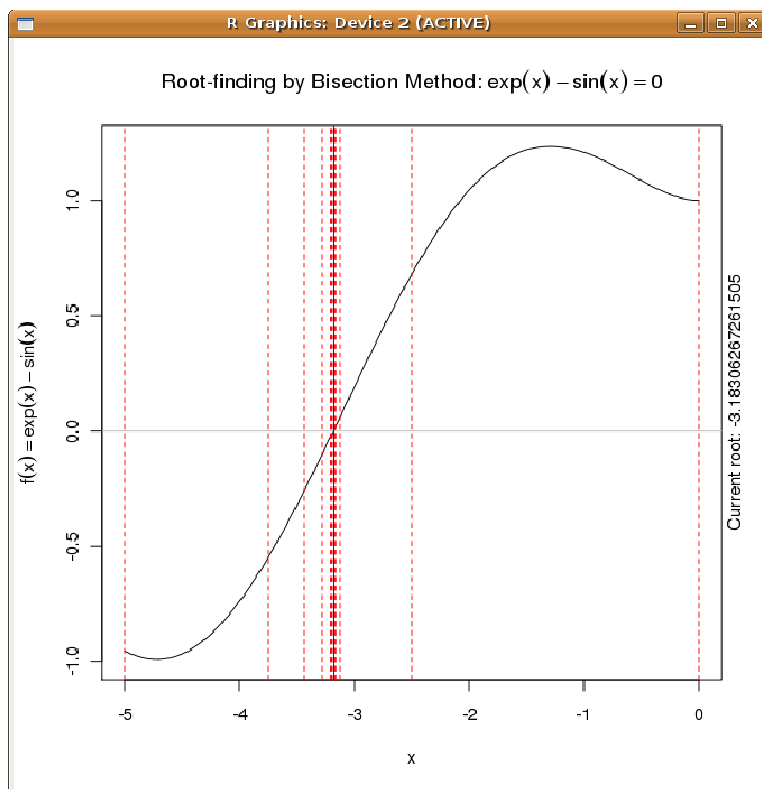
zmieniamy odstęp czasowy pomiędzy ramkami (w miejsce `xxx` należy wpisać wartość).

Graficzny obraz rozwiązania naszego równania dostaniemy wydając polecenia:

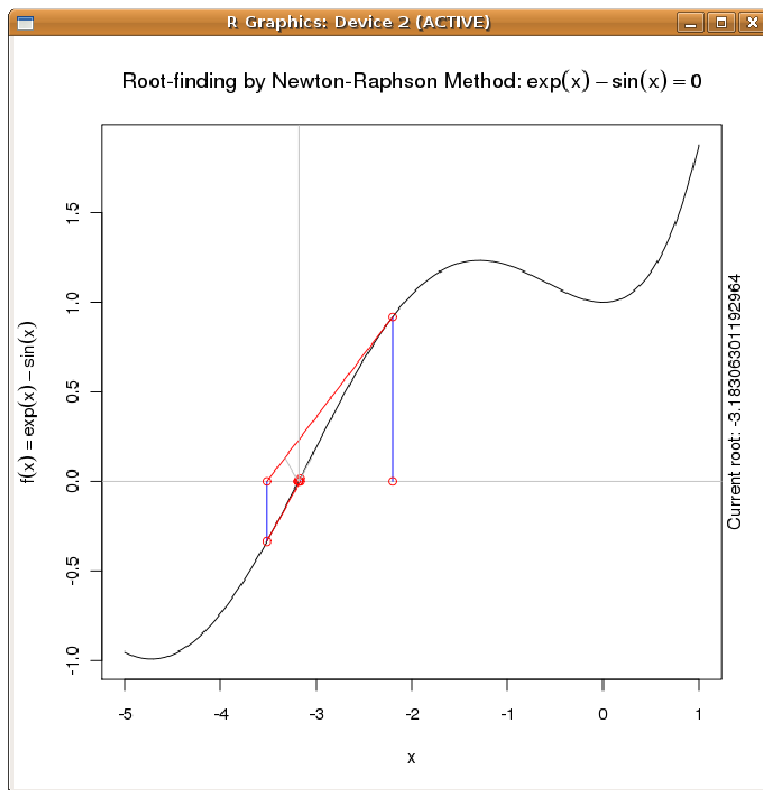
```
f=function(x) exp(x)-sin(x)
```

```
ani.options(nmax=50)
```

```
bisection.method(f,c(-5,0),tol=1e-7)
```

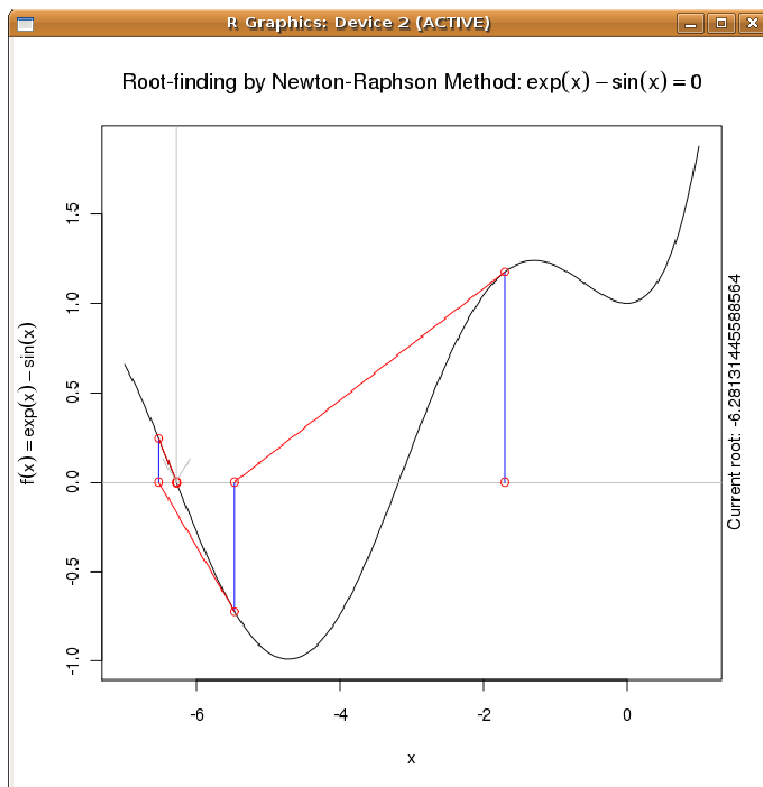


```
ani.options(nmax=50)
newton.method(f, -2.2, c(-5, 1), tol=1e-7)
```

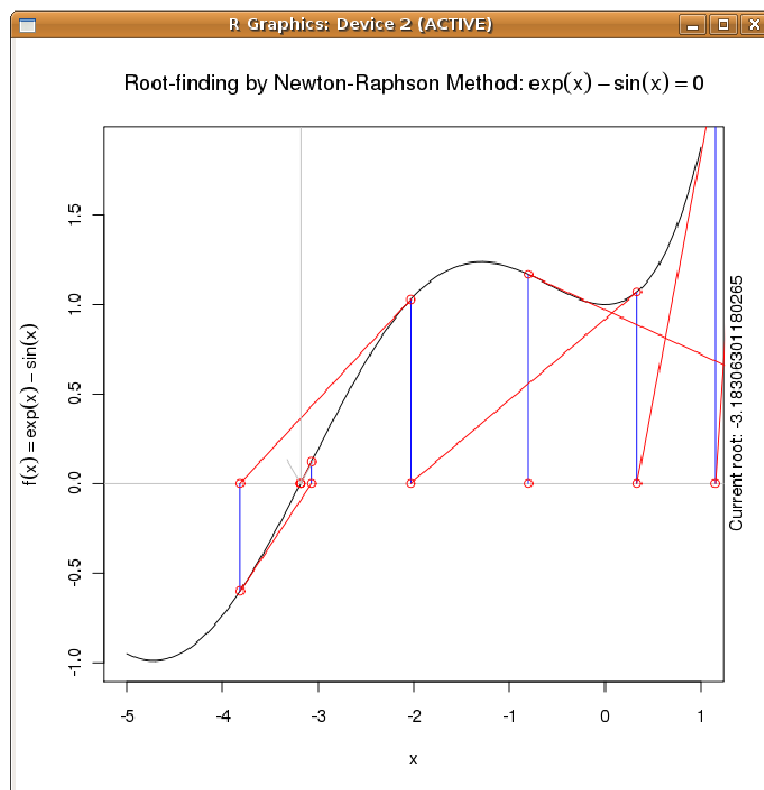


jeśli jednak wystartujemy ze złego punktu, będziemy mieli kłopoty:

```
ani.options(nmax=50)
newton.method(f, -1.7, c(-7, 1), tol=1e-7)
```



choć czasem i tak się uda:
`ani.options(nmax=50)`
`newton.method(f, -0.8, c(-5, 1), tol=1e-7)`



Widać stąd, jak ważne jest, aby być dostatecznie blisko rozwiązania przy wyborze punktu startowego.