

MNUM – Projekt 3.10

Wojciech Rokicki gr. AR2

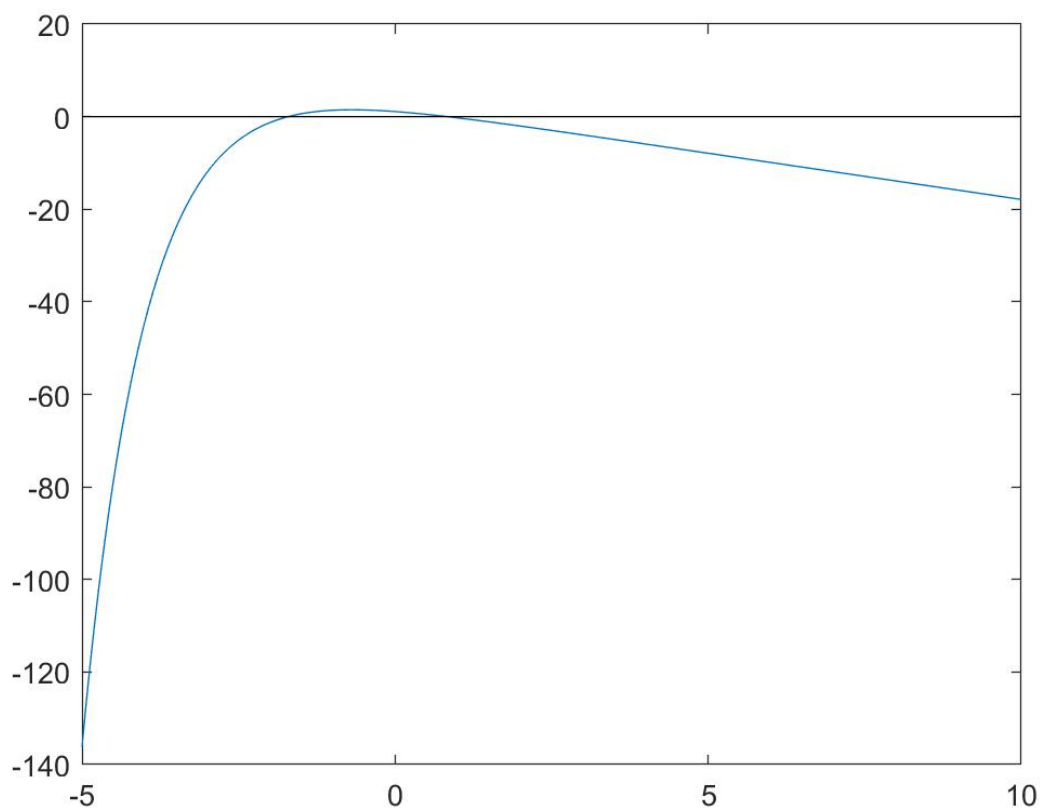
Zadanie 1

Znajdowanie zer funkcji:

$$f(x) = 2.1 - 2x - e^{-x}$$

W przedziale $[-5, 10]$, używając metod:

- a) bisekcji
- b) siecznych.



Widać, że od pewnego miejsca wartości dla ujemnych x są znacznie mniejsze od zera, a wartości dla x dodatnich maleją, więc obszar poszukiwań miejsc zerowych zawężymy do przedziału $[-3, 3]$.

Stąd przedziały startowe to: $[-3, 0]$ oraz $[0, 3]$.

- a) Metoda bisekcji

Startujemy z początkowego przedziału $[a, b] = [a_0, b_0]$ izolacji pierwiastka. W metodzie bisekcji, w każdej iteracji:

- a. Bieżący przedział zawierający zero funkcji, $[a_n, b_n]$, jest dzielony na dwie połowy, punktem środkowym c_n ,

$$c_n = \frac{a_n + b_n}{2},$$

i obliczana jest wartość funkcji w nowym punkcie $f(c_n)$.

- b. Obliczane są iloczyny $f(a_n) \cdot f(c_n)$ i $f(b_n) \cdot f(c_n)$, nowy przedział zawierający pierwiastek jest wybierany jako ten z dwóch podprzedziałów, któremu odpowiada iloczyn ujemny. Końce tego przedziału oznaczane są przez a_{n+1}, b_{n+1} .

Metoda jest powtarzana tak długo, aż np. $|f(c_n)| \leq \delta$, gdzie δ to założona dokładność rozwiązania. Test ten może być nieprecyzyjny dla funkcji, których pochodna jest bardzo mała w otoczeniu zera funkcji, dlatego też sprawdzana jest długość przedziału, $b_n - a_n$, żądając aby była dostatecznie mała.

Dokładność rozwiązania uzyskanego metodą bisekcji zależy jedynie od ilości wykonanych iteracji, a nie zależy od dokładności obliczania wartości funkcji $f(x)$ na krańcach kolejnych przedziałów izolowanego pierwiastka. Niech $\varepsilon_n = b_n - a_n$ oznacza długość przedziału w n -tym kroku iteracji, $\varepsilon_0 = [a, b], n = 0, 1, 2, \dots$

Wówczas

$$\varepsilon_{n+1} = \frac{1}{2} \varepsilon_n$$

Stąd metoda bisekcji jest zbieżna liniowo ($p = 1$), z ilorazem zbieżności $k = 0.5$.

- b) Metoda siecznych jest bardzo podobna do metody bisekcji – różnica polega na tym, że aktualny przedział $[a_n, b_n]$ izolacji pierwiastka dzielony jest nie na dwa równe, ale na dwa najczęściej nierówne podprzedziały, prostą (sieczną) łączącą na płaszczyźnie (f, x) punkty $(f(x_{n-1}), x_{n-1})$ i $(f(x_n), x_n)$, przecinającą oś rzędnych w punkcie x_{n+1} . W przeciwieństwie jednak do metody bisekcji nie dbamy o zachowanie przedziału izolacji pierwiastka (sieczną prowadzimy zawsze pomiędzy dwoma ostatnio wyznaczonymi punktami i te dwa ostatnie punkty wyznaczają nowy przedział). Z konstrukcji graficznej będziemy mieli:

$$\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} = \frac{f(x_n) - 0}{x_n - x_{n+1}}$$

Skąd

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

Rząd zbieżności metody siecznych $p = \frac{1+\sqrt{5}}{2} = 1.618$, tak więc metoda ta jest szybsza od metody bisekcji. Jednakże, jest ona zbieżna jedynie lokalnie, stąd w praktyce może być niezbieżna - jeśli początkowy przedział izolacji pierwiastka nie jest dostatecznie mały.

Kod źródłowy:

Metoda bisekcji:

```
function [ x, y ] = bisection(a,b,imax)
    %sprawdzamy obszar izolacji
    if (f(a)*f(b)>0)
        error('Brak obszaru izolacji');
    end
    for i = 1:imax
        %obszar jest dzielony na pół
        c = (a+b)/2;
        %sprawdzamy warunki iloczynu i wybieramy kolejny przedział
```

```

        if(f(c) * f(a)<0)
            b = c;
        else
            a = c;
        end
        x(i) = c;
        y(i) = f(c);
    end
end

```

Metoda siecznych:

```

function [ x, y ] = secant( xnp, xn, imax )
    %sprawdzamy obszar izolacji
    if(f(xnp)*f(xn)>0)
        error('Brak obszaru izolacji');
    end
    for i = 1:imax+1
        %obliczamy wartość kolenego przybliżenie miejsca zerowego
        xnn = (xnp*f(xn) - xn*f(xnp)) / (f(xn) - f(xnp));
        x(i) = xnn;
        y(i) = f(xnn);
        %Przechodzimy do kolejnego przedziału
        xnp = xn;
        xn = xnn;
    end
end

```

Wyniki:

Dla pierwszego zaproponowanego przedziału $[-3,0]$ metoda siecznych okazała się być niebezpieczna, obszar poszukiwań wychodził poza przedział izolacji i algorytm znajdował zero z drugiego przedziału. W związku z tym obszar izolacji został zawężony do $[-2,-0.5]$.

Przedział $[-3,0]$:

Iteracja	Metoda bisekcji		Metoda siecznych	
	x	y	x	y
1	-1.50000	0.61831	-0.25218	1.31753
2	-2.25000	-2.88773	1.27520	-0.72979
3	-1.87500	-0.67081	0.73075	0.15694
4	-1.68750	0.06905	0.82711	0.00845
5	-1.78125	-0.27477	0.83260	-0.00012
6	-1.73437	-0.09663	0.83252	9.48916e-08
7	-1.71093	-0.01227	0.83252	1.06159e-12
8	-1.69921	0.02876	0.83252	-5.55111e-17
9	-1.70507	0.00834	0.83252	-5.55111e-17
10	-1.70800	-0.00194	NaN	NaN

Przedział $[0,3]$:

Iteracja	Metoda bisekcji		Metoda siecznych	
	x	y	x	y
1	1.50000	-1.12313	0.6535	0.27278
2	0.75000	0.12763	0.8051	0.04278
3	1.12500	-0.47465	0.8333	-0.00118
4	0.93750	-0.16660	0.8325	4.56300e-06
5	0.84375	-0.01759	0.8325	4.78518e-10
6	0.79687	0.05551	0.8325	-5.55111e-17
7	0.82031	0.01908	0.8325	1.11022e-16
8	0.83203	0.00077	0.8325	-5.55111e-17
9	0.83789	-0.00840	0.8325	-5.55111e-17
10	0.83496	-0.00381	NaN	NaN

Wnioski:

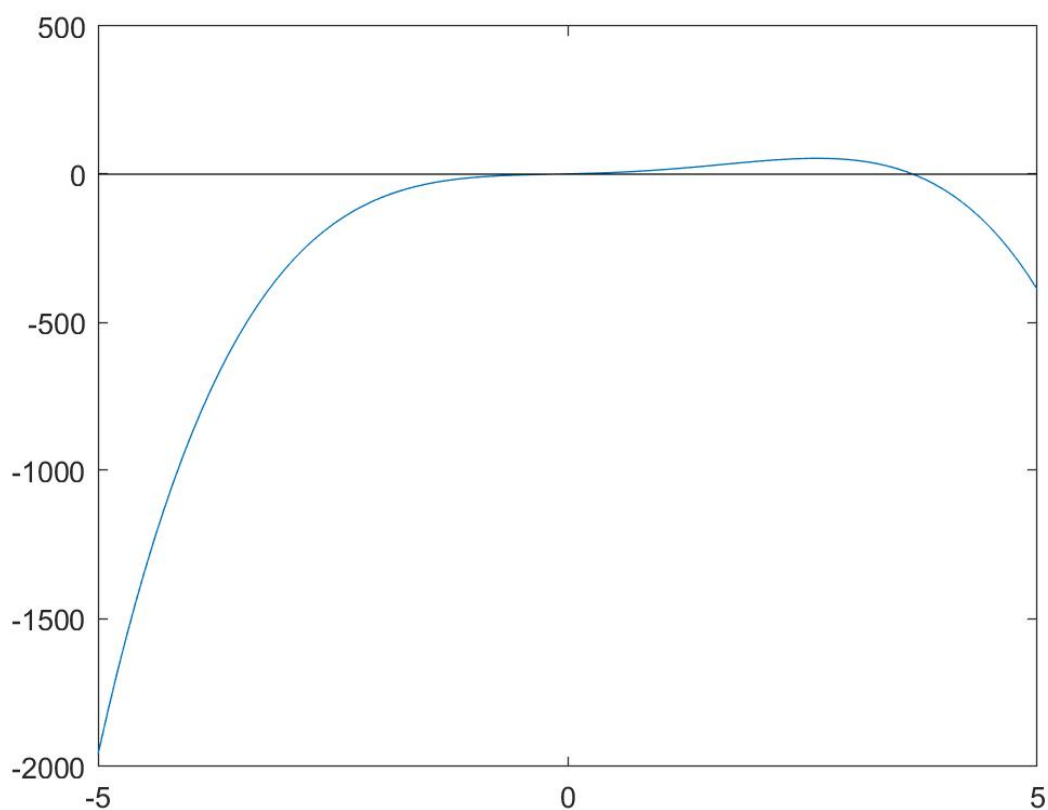
W przypadku pierwszego inicjalnego przedziału $[-3,0]$ metoda siecznych okazała się niebezpieczna, wynika to z tego, że w tym przedziale pochodna funkcji przyjmuje wartości bliskie zero (funkcja jest „płaska”). Przez to sieczna może wyjść poza obszar izolacji, co doprowadzało do znajdowania drugiego zera funkcji. Jednakże po ograniczeniu przedziału do $[-2,-0.5]$, zarówno w nim jak i w drugim przedziale $[0,3]$, metoda siecznych była znacznie szybciej zbieżna, co potwierdza teoretyczne rozważania. Warto jednak zauważyć, że metoda bisekcji zwracała dobry wynik dla obszaru izolacji o dowolnym rozmiarze.

Zadanie 2

Znajdowanie pierwiastków wielomianu

$$f(x) = -2x^4 + 6x^3 + 3x^2 + 7x + 1$$

- Pierwiastków rzeczywistych metodą Newtona
- Pierwiastków rzeczywistych i zespolonych metodą MM2



- Metoda Newtona, zwana też metodą stycznych, zakłada aproksymację funkcji jej liniowym przybliżeniem wynikającym z uciętego rozwinięcia w szereg Taylora w aktualnym punkcie x_n (aktualnym przybliżeniu pierwiastka)

$$f(x) \approx f(x_n) + f'(x_n)(x - x_n)$$

Następny punkt, x_{n+1} , wynika z przyrównania do zera sformułowanej lokalnej liniowej aproksymacji funkcji $f(x)$, tzn. z równania

$$f(x_n) + f'(x_n)(x_{n+1} - x_n) = 0$$

co prowadzi do zależności iteracyjnej

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

Metoda Newtona jest zbieżna lokalnie - jeśli zaczynamy ją stosować w punkcie zbytnio oddalonym od rozwiązania (poza obszarem atrakcji pierwiastka), to może być ona rozbieżna. Natomiast metoda Newtona jest (lokalnie, asymptotycznie) bardzo szybka, jej zbieżność jest kwadratowa (tzn. z rzędem $p = 2$).

Metoda Newtona jest szczególnie efektywna w przypadku, gdy krzywa $f(x)$ jest bardzo stroma w otoczeniu danego pierwiastka, natomiast nie zaleca się jej stosowania w przypadku, gdy krzywa $f(x)$ jest w pobliżu pierwiastka (tzn. punktu przecięcia z osią Ox) prawie pozioma, tzn. $f'(x)$ ma bardzo małą wartość.

b. Metoda Müllera

Metoda polega na aproksymacji wielomianu w otoczeniu rozwiązania funkcją kwadratową. Może być traktowana jako uogólnienie metody siecznych - zamiast interpolacji w dwóch punktach funkcją liniową (tzn. sieczną) wykonujemy interpolację w trzech punktach funkcją kwadratową. Istnieje również efektywna realizacja oparta na wykorzystaniu informacji o wielomianie jedynie w jednym punkcie, tzn. wykorzystująca do wyznaczenia funkcji kwadratowej wartości wielomianu i jego pierwszej i drugiej pochodnej w aktualnym punkcie.

MM2. Wersja metody wykorzystująca informację nie o wartości wielomianu w kolejnych trzech punktach, ale o wartości wielomianu i jego pochodnych, pierwszego i drugiego rzędu w aktualnym punkcie (przybliżeniu zera) x_k - wersja nieco efektywniejsza obliczeniowo z tego powodu, że obliczenie wartości wielomianu w $k + 1$ punktach jest nieco kosztowniejsze niż obliczenie wartości wielomianu i jego k kolejnych pochodnych w jednym punkcie.

Oznaczając szukaną parabolę jak w MM1:

$$y(z) = az^2 + bz + c$$

Dla $z \stackrel{\text{def}}{=} x - x_k$, w punkcie $z = 0$ bezpośrednio wynika:

$$y(0) = c = f(x_k)$$

$$y'(0) = b = f'(x_k)$$

$$y''(0) = 2a = f''(x_k)$$

Co prowadzi do wzoru na pierwiastki:

$$z_{+,-} = \frac{-2f(x_k)}{f'(x_k) \pm \sqrt{(f'(x_k))^2 - 2f(x_k)f''(x_k)}}$$

Do przybliżenia zera α bierzemy pierwiastek paraboli o mniejszym module:

$$x_{k+1} = x_k + z_{\min}$$

Gdzie z_{\min} jest wybierany spośród $\{z_+, z_-\}$.

$$z_+ = \frac{-2c}{b + \sqrt{b^2 - 4ac}}$$

$$z_- = \frac{-2c}{b - \sqrt{b^2 - 4ac}}$$

Do kolejnego przybliżenia rozwiązania bierzemy pierwiastek położony jak najbliżej x_2 , tj. o mniejszym module,

$$x_3 = x_2 + z_{\min}$$

Gdzie:

$$z_{\min} = z_+, \text{ gdy } |b + \sqrt{b^2 - 4ac}| \geq |b - \sqrt{b^2 - 4ac}|$$

$$z_{\min} = z_-, \text{ w przeciwnym przypadku}$$

Podobnie jak MM1, MM2 znajduje zespolone pierwiastki wielomianu.

Metoda Müllera jest zbieżna lokalnie, z rzędem zbieżności 1.84. Jest więc (lokalnie) bardziej efektywna niż metoda siecznych, jest niewiele wolniejsza od metody Newtona. Z konstrukcji

metody wynika, że może ona być stosowana do poszukiwania zer rzeczywistych i zespolonych nie tylko wielomianów, ale i innych funkcji nieliniowych (analitycznych).

Kod źródłowy:

Metoda Newtona:

```
function [ x, y ] = newton( xn, imax )
    for i = 1:imax
        %obliczamy kolejne przybliżenie miejsca zerowego
        xn = xn - (f2(xn))/(f2p(xn));
        x(i) = xn;
        y(i) = f2(xn);
    end
end
```

MM1:

```
function [ x, y ] = mm1( x0, x1, x2, imax )
    for i = 1:imax
        %tworzymy zmienne przyrostowe i obliczamy c
        z0 = x0 - x2;
        z1 = x1 - x2;
        c = f2(x2);
        %tworzymy układ równań do obliczenia a,b i rozwiązujemy go
        A = [z0^2, z0; z1^2, z1];
        B = [f2(x0) - c; f2(x1) - c];
        [w] = linsolve(A,B);
        a = w(1,1);
        b = w(2,1);

        %wybieramy zmin jako ten o najmniejszym module
        if(abs(b + sqrt(b^2 - 4*a*c)) >= abs(b - sqrt(b^2 - 4*a*c)))
            zmin = (-2*c)/(b + sqrt(b^2 - 4*a*c));
        else
            zmin = (-2*c)/(b - sqrt(b^2 - 4*a*c));
        end

        %obliczamy kolejne przybliżenie miejsca zerowego
        x3 = x2 + zmin;
        x(i) = x3;
        y(i) = f2(x3);
        d0 = abs(x3-x0);
        d1 = abs(x3-x1);
        d2 = abs(x3-x2);

        %Odrzucamy spośród x0,x1,x3 to przybliżenie które jest
        % najbardziej oddalone od x3
        if(d1 < d0)
            u = x1;
            x1 = x0;
            x0 = u;
        end
        if(d2 < d1)
            u = x2;
            x2 = x1;
            x1 = u;
        end
        %Przygotowujemy się do kolejnej iteracji
        x2 = x3;
    end
end
```

MM2:

```
function [ x, y ] = mm2( xn, imax )
    for i = 1:imax
        pier = sqrt((f2p(xn))^2 - 2* f2(xn)*f2b(xn));
        %zgodnie z teorią wybieramy pierwiastek o mniejszym module
        if(abs(f2p(xn) + pier) > abs(f2p(xn) - pier))
            z = (-2*f2(xn))/(f2p(xn) + pier);
        else
            z = (-2*f2(xn))/(f2p(xn) - pier);
        end
        %obliczamy kolejne przybliżenie miejsca zerowego
        xn = xn + z;
        x(i) = xn;
        y(i) = f2(xn);
    end
end
```

Wyniki:

Metoda Newtona:

Punkt	-8		-5		0		3	
Iteracja	x	y (* 1.0e+03)	x	y	x	y	x	y
1	-5.8631	-3.5096	-3.6272	-617.4350	-0.1429	0.0429	4.6897	-248.7324
2	-4.2700	-1.1062	-2.6046	-194.9528	-0.1494	0.0000	4.0585	-62.6999
3	-3.0836	-0.3488	-1.8397	-61.9905	-0.1494	0.0000	3.7556	-10.4420
4	-2.1988	-0.1104	-1.2587	-20.0411	-0.1494	-0.0000	3.6812	-0.5363
5	-1.5331	-0.0353	-0.8023	-6.6122	-0.1494	-0.0000	3.6769	-0.0017
6	-1.0203	-0.0116	-0.4330	-2.0257	-0.1494	-0.0000	3.6769	-0.0000
7	-0.6091	-0.0038	-0.1926	-0.2823	-0.1494	-0.0000	3.6769	-0.0000
8	-0.2895	-0.0009	-0.1496	-0.0011	-0.1494	-0.0000	3.6769	0
9	-0.1553	-0.0000	-0.1494	0.0000	-0.1494	-0.0000	3.6769	0
10	-0.1494	-0.0000	-0.1494	-0.0000	-0.1494	-0.0000	3.6769	0

Metoda MM2:

Punkt	-8		-5		0		3	
Iteracja	x	y(*1.0e+03)	x	y(*1.0e+02)	x	y	x	y
1	-5.1359 - 2.0094i	-0.5536 - 2.7018i	-3.1563 - 1.2895i	-0.9562 - 4.7323i	-0.1529	-0.0225	3.7363	-7.7725
2	-3.7252 - 0.0003i	-0.6788 - 0.0002i	-1.6405 - 1.6866i	1.0776 - 0.4473i	-0.1494	0.0000	3.6769	0.0050
3	-2.3200 - 0.9847i	-0.0336 - 0.1635i	-0.5863 - 1.5509i	0.1300 + 0.2238i	-0.1494	-0.0000	3.6769	-0.0000
4	-1.1797 - 1.2900i	0.0358 - 0.0160i	-0.0421 - 1.1089i	-0.0504 + 0.0112i	-0.1494	-0.0000	3.6769	0
5	-0.4334 - 1.1976i	0.0040 + 0.0062i	-0.2587 - 0.9328i	-0.0010 + 0.0022i	-0.1494	-0.0000	3.6769	0
6	-0.2191 - 0.9035i	-0.0005 - 0.0003i	-0.2637 - 0.9168i	0.0000 + 0.0000i	-0.1494	-0.0000	3.6769	0

7	-0.2637 - 0.9167i	0.0000 - 0.0000i	-0.2637 - 0.9168i	0.0000 - 0.0000i	-0.1494	-0.0000	3.6769	0
8	-0.2637 - 0.9168i	-0.0000 + 0.0000i	-0.2637 - 0.9168i	0.0000 + 0.0000i	-0.1494	-0.0000	3.6769	0
9	-0.2637 - 0.9168i	0.0000 - 0.0000i	-0.2637 - 0.9168i	0.0000 - 0.0000i	-0.1494	-0.0000	3.6769	0
10	-0.2637 - 0.9168i	0.0000 - 0.0000i	-0.2637 - 0.9168i	-0.0000 + 0.0000i	-0.1494	-0.0000	3.6769	0

Znalezione pierwiastki:

$$\begin{aligned}x_0 &= -0.1494, & x_1 &= 3.6769 \\x_3 &= -0.2637 - 0.9168i,\end{aligned}$$

Wnioski:

W ogólnym przypadku metody Newtona, MM2, wykazywały podobne szybkości zbieżności, co znajduje odzwierciedlenie w teorii (rzędy zbieżności: dla metody Newtona $p=2$, dla metody Mullera $p = 1.84$). Należy zauważyć, że metoda Newtona nie we wszystkich przypadkach była zbieżna. Punkty te są zbyt oddalone od obszaru atrakcji pierwiastka, bądź też pochodna funkcji przyjmuje w nich małą wartość. W związku z faktem że w niektórych punktach wartości pochodnej funkcji przyjmują małe wartości, Metody Mullera są w takich przypadkach szybciej zbieżne. Zgodnie z oczekiwaniami metody Mullera znajdują pierwiastki zarówno rzeczywiste i urojone, natomiast metoda Newtona tylko pierwiastki rzeczywiste. Dla zaproponowanych przedziałów metoda Mullera znajdowała jeden z pierwiastków zespolonych, drugi jest liczbą do niego sprzężoną.