Metoda iteracyjna Müllera

Jakub Grobelny

Analiza Numeryczna (M) - Zadanie P2.4
16 grudnia 2018

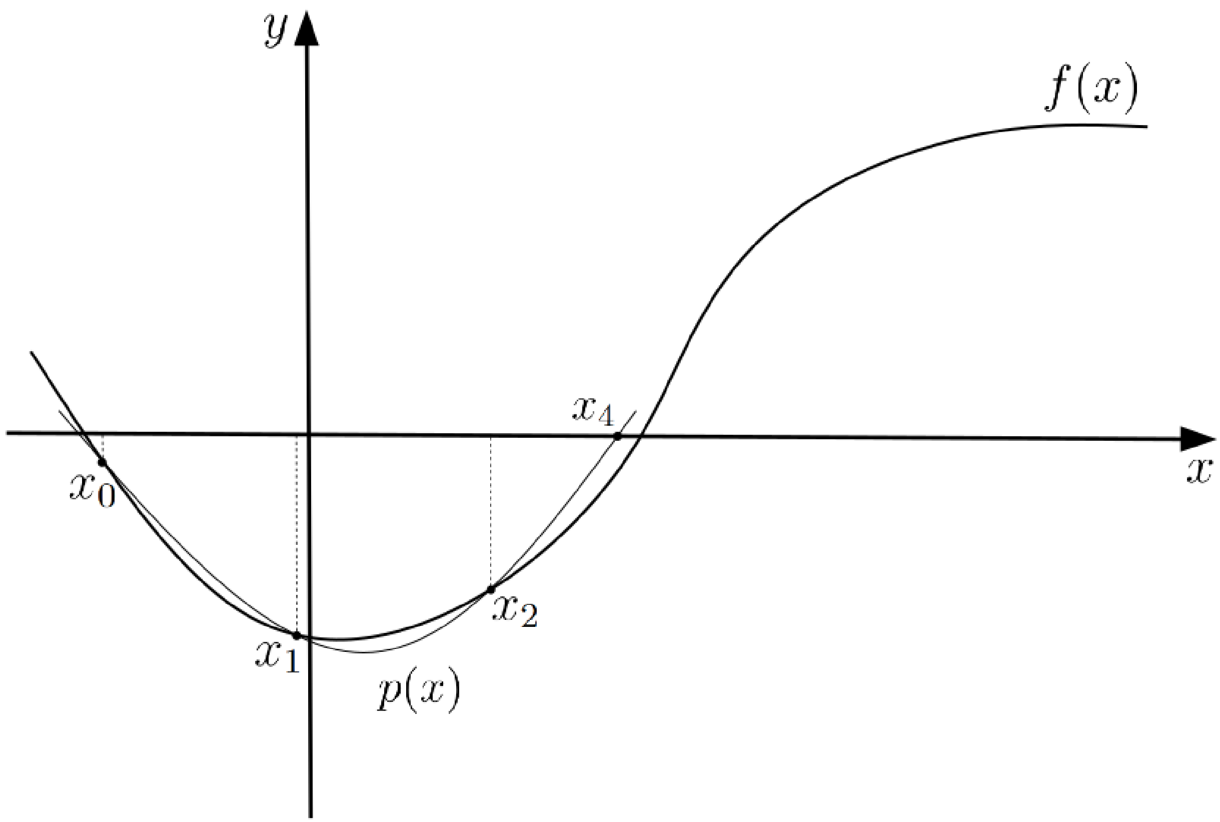
Streszczenie

Rozwiązywanie równań nielinowych jest jednym z podstawowych zagadnień analizy numerycznej. Polega ono na znajdowaniu miejsc zerowych funkcji, które nie są funkcjami liniowymi. Zostało opracowanych wiele metod iteracyjnych, które pozwalają znajdywać te miejsca. Niniejsze sprawozdanie jest podsumowaniem eksperymentu polegającego na porównaniu metody Müllera z innymi metodami iteracyjnymi, a dokładniej z metodą siecznych oraz metodą Newtona.

Spis treści

1 Metoda iteracyjna Müllera

Niech p_n będzie wielomianem drugiego stopnia interpolującym funkcję f w punktach x_{n-2}, x_{n-1}, x_n. Zdefiniujmy x_{n+1} jako miejsce zerowe wielomianu f, które jest najbliższe x_n. Skoro wielomian p_n interpoluje funkcję f, to możemy oczekiwać, że jego miejsce zerowe znajdzie się relatywnie blisko miejsca zerowego funkcji f. Można zauważyć, że w rzeczywistości ciąg miejsc zerowych kolejnych wielomianów interpolujących stanowi ciąg kolejnych przybliżeń takiej wartości α, że $f(\alpha) = 0$.



Rysunek 1: Graficzna interpretacja metody Müllera

Na podstawie powyższej intuicji można wyprowadzić wzór na metodę iteracyjną, działającą w opisany sposób.

Definicja 1: Iloraz różnicowy [1]

Ilorazem różnicowym n-tego rzędu funkcji $f : X \to Y$ w punktach $x_0, x_1, ..., x_n$ nazywamy funkcję

$$f[x_0, x_1, ..., x_n] := \sum_{i=0}^{n} \frac{f(x_i)}{\prod\limits_{\substack{j=0 \\ j \neq i}} (x_i - x_j)}$$

Definicja 2: Postać Newtona wielomianu interpolującego [1]

Mając funkcję f oraz zadane $n+1$ punktów $x_0, x_1, ..., x_n$ można skonstruować wielomian L_n stopnia n interpolujący funkcję f przy użyciu następującego wzoru:

$$L_n(x) := \sum_{i=0}^{n} b_i \, p_i(x),$$

gdzie wielomiany p_i dla $i = 0, 1, ..., n$ mają postać

$$p_i(x) := \prod_{j=0}^{i-1} (x - x_j),$$

zaś współczniki b_i definiujemy następująco:

$$b_i := f[x_0, ..., x_i].$$

Korzystając z definicji 2 możemy zapisać wielomian p_n w następującej postaci:

$$p_n(x) = f(x_n) + f[x_n, x_{n-1}](x - x_n) + f[x_n, x_{n-1}, x_{n-2}](x - x_n)(x - x_{n-1}),$$

którą dalej możemy przekształcić do

$$p_n(x) = f(x_n) + \lambda(x - x_n) + f[x_n, x_{n-1}, x_{n-2}](x - x_n)^2,$$

gdzie

$$\lambda = f[x_n, x_{n-1}] + f[x_n, x_{n-2}] - f[x_{n-1}, x_{n-2}].$$

Mając wzór na wielomian p_n można wyprowadzić wzór na jego miejsca zerowe, korzystając ze wzoru

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

na miejsca zerowe funkcji $y = ax^2 + bx + c$.

Po podstawieniu otrzymujemy ostateczny wzór

$$x_{n+1} = x_n - \frac{2f(x_n)}{\lambda \pm \sqrt{\lambda^2 - 4f(x_n)f[x_n, x_{n-1}, x_{n-2}]}}, \tag{1}$$

gdzie znak w mianowniku wybieramy tak, żeby wartość bezwzględna z mianownika była możliwie największa. Dzięki temu uzyskamy to miejsce zerowe, które jest położone najbliżej x_n.

Metoda Müllera jest metodą o złożoności w przybliżeniu równej 1.84.

Korzystając ze wzoru (1) możemy napisać funkcję w języku Julia, która będzie obliczać kolejne przybliżenie miejsca zerowego dla danych dotychczasowych przybliżeń.

Program 1: Implementacja metody Müllera

```julia
function mullers_method(x0, x1, x2, f)
    f0 = f(x0)
    f1 = f(x1)
    f2 = f(x2)

    numerator = BigFloat(2) * f2

    # Funkcja difference_quotient() oblicza iloraz roznicowy dla
    # zadanych punktow w postaci (x,y)
    lambda = difference_quotient([(x2, f2), (x1, f1)]) +
             difference_quotient([(x2, f2), (x0, f0)]) -
             difference_quotient([(x1, f1), (x0, f0)])

    x0_x1_x2_diff_quot = difference_quotient([
        (x0, f0),
        (x1, f1),
        (x2, f2)
    ])

    denom_r = sqrt(lambda^2.0 - 4.0*f(x2) * x0_x1_x2_diff_quot + 0im)
    denom_0 = lambda - denom_r
    denom_1 = lambda + denom_r

    denominator = abs(denom_0) > abs(denom_1) ? denom_0 : denom_1

    return x2 - numerator/denominator
end
```

2 Inne metody rozwiązywania równań nieliniowych

2.1 Metoda Newtona [1]

Niech $f(x)$ będzie funkcją, której miejsce zerowe chcemy wyznaczyć numerycznie, oraz niech $f(\alpha) = 0$ i niech x będzie przybliżeniem α. Jeżeli istnieje f'', to ze wzoru Taylora mamy

$$0 = f(\alpha) = f(x + h) = f(x) + hf'(x) + \mathcal{O}(h^2),$$

gdzie $h = \alpha - x$. Jeżeli h dąży do zera, to możemy pominąć składnik $\mathcal{O}(h^2)$. Mamy wtedy

$$0 = f(x) + hf'(x)$$

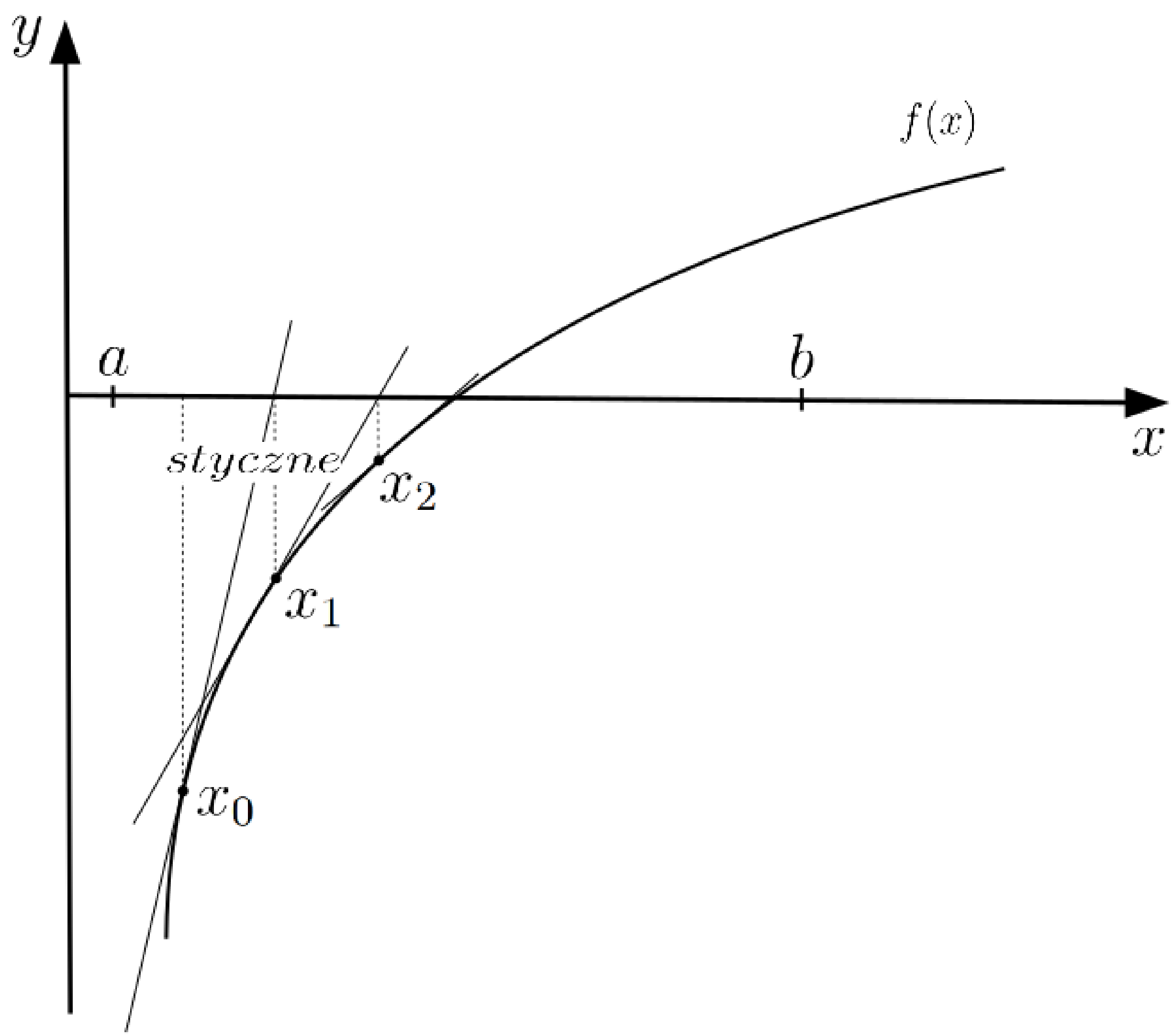
$$-f(x) = hf'(x)$$

$$h = \frac{-f(x)}{f'(x)}.$$

Przyjęliśmy, że x jest przybliżeniem α, więc $x - \frac{f(x)}{f'(x)}$ powinno być lepszym przybliżeniem. Dzięki czemu możemy wyprowadzić wzór rekurencyjny na n-te przybliżenie:

$$x_n := x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}. \tag{2}$$

Metoda Newtona wymaga podania początkowego punktu x_0. Wybór tego punktu jest istotny, gdyż metoda Newtona nie zawsze jest zbieżna. Rząd zbieżności metody Newtona to 2 (zbieżność kwadratowa) z wyjątkiem przypadków, gdy istnieją wielokrotne miejsca zerowe. Wtedy zbieżność jest linowa (rząd zbieżności równy 1).



Rysunek 2: Graficzna interpretacja metody Newtona

Obliczanie kolejnych przybliżeń w rzeczywistości polega na wyznaczaniu stycznej do wykresu $f(x)$ w punkcie x_n, a następnie znajdowaniu punktu, w którym ta styczna przecina oś OX. Otrzymany punkt przyjmujemy jako kolejne przybliżenie x_{n+1}.

Implementacja metody Newtona w jęzku Julia przy użyciu wzoru (2) jest bardzo prosta:

Program 2: Implementacja metody Newtona

```
function newtons_method(x0, f, df)
    return x0 - f(x0) / df(x0)
end
```

2.2 Metoda siecznych [1]

Wadą metody Newtona jest to, że trzeba znać pochodną funkcji $f(x)$, której miejsce zerowe chcemy wyznaczyć.

Korzystając z faktu, że

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} = f[x_{n-1}, x_n],$$

możemy zastąpić pochodną we wzorze z metody Newtona na iloraz różnicowy, co daje nam wzór

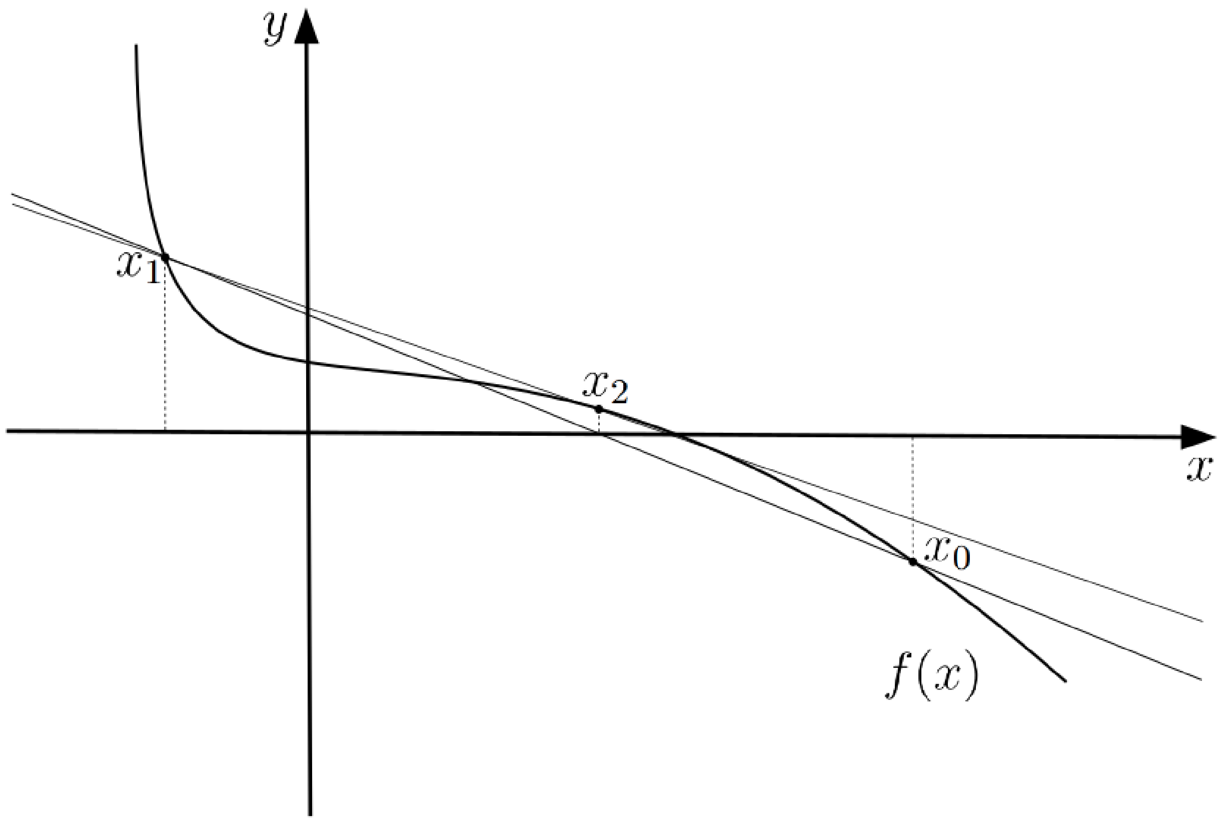
$$x_n := x_{n-1} - f(x_{n-1}) \frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})}.$$

Rząd zbieżności metody siecznych to $\varphi = \frac{1+\sqrt{5}}{2} \approx 1.618$.

Program 3: Implementacja metody siecznych

```
function secant_method(x0, x1, f)
    f0 = f(x0)
    f1 = f(x1)
    return x1 - f1 * (x1 - x0) / (f1 - f0)
end
```

Interpretacja graficzna metody siecznych jest taka, iż prowadzimy prostą przechodzącą przez punkty $(x_{n-1}, f(x_{n-1}))$ oraz $(x_n, f(x_n))$, gdzie x_n i x_{n-1} są poprzednimi przybliżeniami miejsca zerowego funkcji f, a następnie szukamy miejsca, w którym poprowadzona prosta przecina oś OX i przyjmujemy to miejsce jako nowe przybliżenie x_{n+1}.



Rysunek 3: Graficzna interpretacja metody siecznych

3 Opis eksperymentu

Celem niniejszego sprawozdania było sprawdzenie, jak metoda Müllera wypada na tle innych metod rozwiązywania równań nieliniowych. W tym celu wszystkie trzy metody zostały użyte do znalezienia miejsc zerowych różnych funkcji, a następnie obliczona została liczba poprawnych cyfr w kolejnych iteracjach przy użyciu wzoru

$$\lfloor -log_{10}(\delta) \rfloor,$$

gdzie δ oznacza błąd względny otrzymanego wyniku.

Obliczenia zostały wykonane w języku Julia (wersja 1.0.1) przy użyciu 256-bitowej precyzji (`setprecision(256)`).

Metody były testowane w następujący sposób:

1. Dla każdej funkcji wybrane zostało pierwsze przybliżenie x_0.
2. Jednokrotnie wykonywano iterację metody Newtona aby uzyskać x_1.
3. Jednokrotnie wykonywano iterację metody siecznych dla x_0 i x_1, uzyskując jako wynik x_2.
4. Uruchamiano metodę Newtona dla x_0, metodę siecznych dla x_0 i x_1 oraz metodę Müllera dla x_0, x_1 i x_2.

Wybrany sposób daje delikatną przewagę metodzie siecznych oraz metodzie Müllera, lecz pozwala zautomatyzować wybór wielu punktów początkowych. Biorąc pod uwagę kwadratową zbieżność metody Newtona, nie powinno to znacząco wpłynąć na wiarygodność pomiarów.

4 Wyniki eksperymentu

Uwaga: w tablicach z wynikami symbol x^* oznacza, że obliczona została dokładna wartość miejsca zerowego.

4.1 $f(x) = x^2 - 3x + 2$

$$f(x) = x^2 - 3x + 2$$

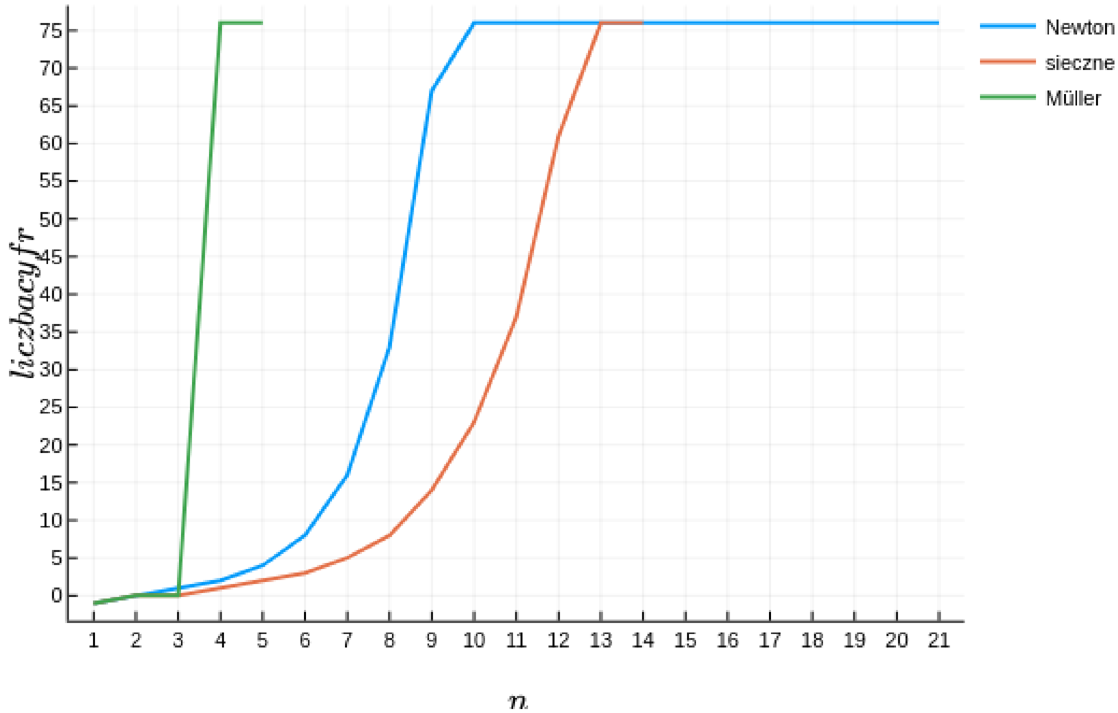
$$f'(x) = 2x - 3$$

$$x_0 = -0.2$$

Dokładna wartość miejsca zerowego: $x = 1$

n	Newton	Sieczne	Müller
0	0.0	0.0	0.0
1	0.0	0.0	0.0
2	1.0	0.0	0.0
3	2.0	1.0	7.6e+01
4	4.0	2.0	7.6e+01
5	8.0	3.0	x^*
6	1.6e+01	5.0	x^*
7	3.3e+01	8.0	x^*
8	6.7e+01	1.4e+01	x^*
9	7.6e+01	2.3e+01	x^*
10	7.6e+01	3.7e+01	x^*
11	7.6e+01	6.1e+01	x^*
12	7.6e+01	7.6e+01	x^*
13	7.6e+01	7.6e+01	x^*
14	7.6e+01	x^*	x^*
15	7.6e+01	x^*	x^*
16	7.6e+01	x^*	x^*
17	7.6e+01	x^*	x^*
18	7.6e+01	x^*	x^*
19	7.6e+01	x^*	x^*
20	7.6e+01	x^*	x^*

Tablica 1: Porównanie liczby poprawnych cyfr w kolejnych iteracjach metod



Rysunek 4: Szybkość zbieżności wszystkich metod do miejsca zerowego $f(x) = x^2 - 3x + 2$

W przypadku funkcji $f(x) = x^2 - 3x + 2$ można zaobserwować wyjątkową przewagę metody Müllera nad metodą siecznych i metodą Newtona. Jest ona najprawdopodobniej spowodowana tym, iż funkcja f jest wielomianem stopnia drugiego, tak samo jak wielomian interpolujący p_n, który jest używany do obliczania kolejnych przybliżeń. Istnieje tylko jeden wielomian stopnia drugiego przechodzący przez dane trzy punkty, a więc $p_n \equiv f$.

4.2 $f(x) = \sqrt{x} - 4$

$$f(x) = \sqrt{x} - 4$$

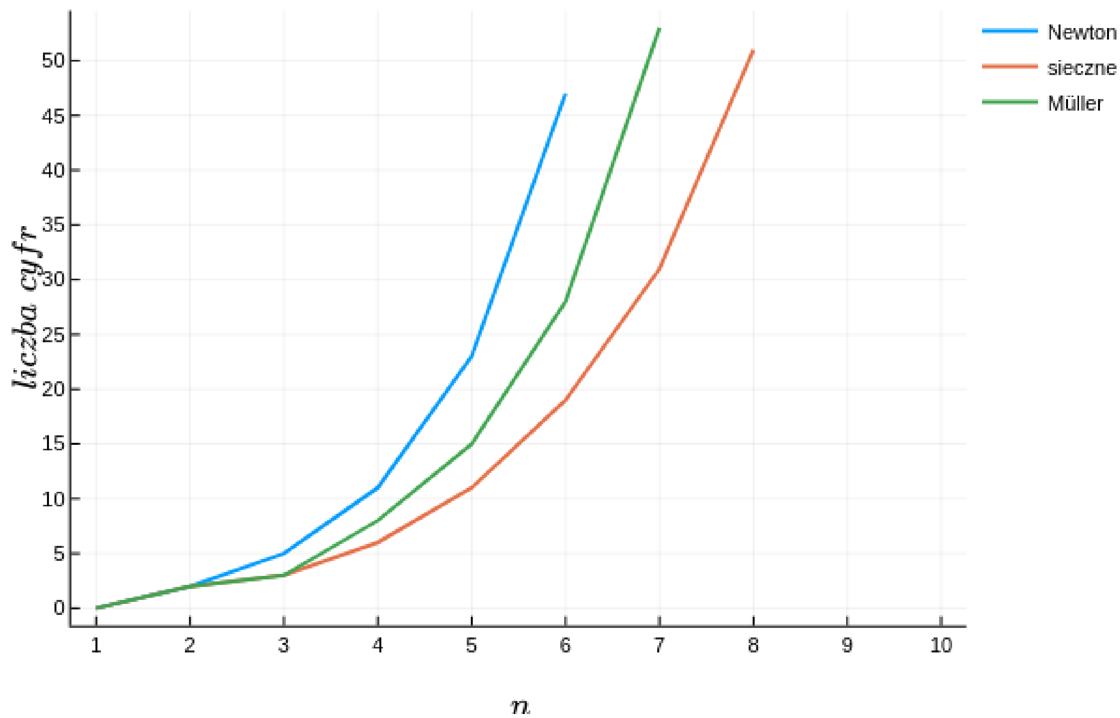
$$f'(x) = \frac{1}{2}x$$

$$x_0 = 18$$

Dokładna wartość miejsca zerowego: $x = 16$

n	Newton	Sieczne	Müller
0	0.0	0.0	0.0
1	2.0	2.0	2.0
2	5.0	3.0	3.0
3	1.1e+01	6.0	8.0
4	2.3e+01	1.1e+01	1.5e+01
5	4.7e+01	1.9e+01	2.8e+01
6	x^*	3.1e+01	5.3e+01
7	x^*	5.1e+01	x^*
8	x^*	x^*	x^*

Tablica 2: Porównanie liczby poprawnych cyfr w kolejnych iteracjach metod



Rysunek 5: Szybkość zbieżności wszystkich metod do miejsca zerowego $f(x) = \sqrt{x} - 4$

4.3 $f(x) = 2(x-1)(x-2)(x-1.5)(x-4)$

$$f(x) = 2(x-1)(x-2)(x-1.5)(x-4)$$

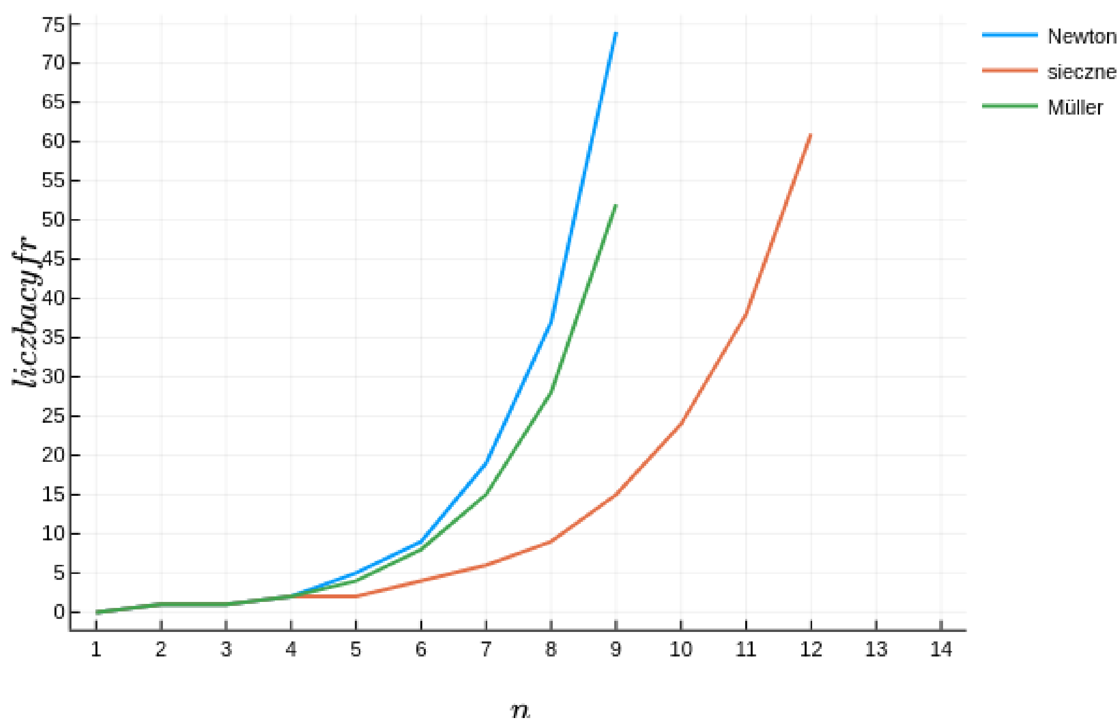
$$f'(x) = 8(x^3 - 6.375x^2 + 12.25x - 7.25)$$

$$x_0 = 3$$

Dokładna wartość miejsca zerowego: $x = 2$

n	Newton	Sieczne	Müller
0	0.0	0.0	0.0
1	1.0	1.0	1.0
2	1.0	1.0	1.0
3	2.0	2.0	2.0
4	5.0	2.0	4.0
5	9.0	4.0	8.0
6	1.9e+01	6.0	1.5e+01
7	3.7e+01	9.0	2.8e+01
8	7.4e+01	1.5e+01	5.2e+01
9	x^*	2.4e+01	x^*
10	x^*	3.8e+01	x^*
11	x^*	6.1e+01	x^*
12	x^*	x^*	x^*
13	x^*	x^*	x^*

Tablica 3: Porównanie liczby poprawnych cyfr w kolejnych iteracjach metod



Rysunek 6: Szybkość zbieżności wszystkich metod do miejsca zerowego $f(x) = 2(x-1)(x-2)(x-1.5)(x-4)$

4.4 $f(x) = x2^x - 2$

$$f(x) = x2^x - 2$$

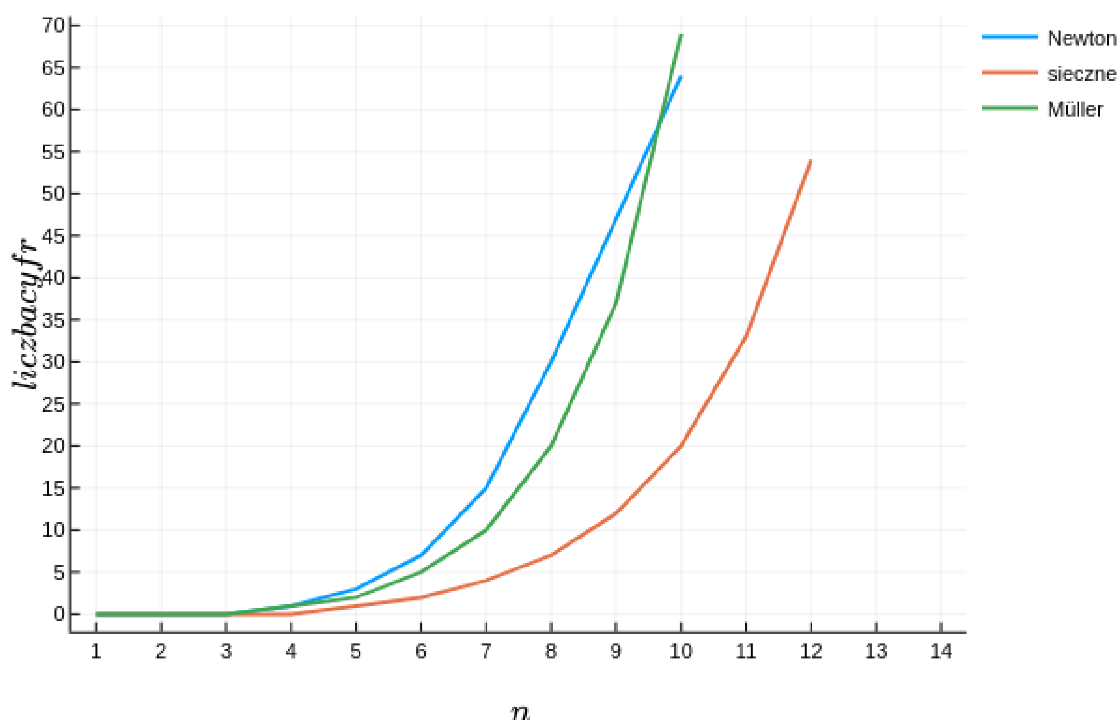
$$f'(x) = 2^x(x\,ln(2) + 1)$$

$$x_0 = 0.123$$

Dokładna wartość miejsca zerowego: $x = 1$

n	Newton	Sieczne	Müller
0	0.0	0.0	0.0
1	0.0	0.0	0.0
2	0.0	0.0	0.0
3	1.0	0.0	1.0
4	3.0	1.0	2.0
5	7.0	2.0	5.0
6	1.5e+01	4.0	1.0e+01
7	3.0e+01	7.0	2.0e+01
8	4.7e+01	1.2e+01	3.7e+01
9	6.4e+01	2.0e+01	6.9e+01
10	x^*	3.3e+01	x^*
11	x^*	5.4e+01	x^*
12	x^*	x^*	x^*

Tablica 4: Porównanie liczby poprawnych cyfr w kolejnych iteracjach metod



Rysunek 7: Szybkość zbieżności wszystkich metod do miejsca zerowego $f(x) = x2^x - 2$

4.5 $f(x) = ln|x|$

$$f(x) = ln|x|$$

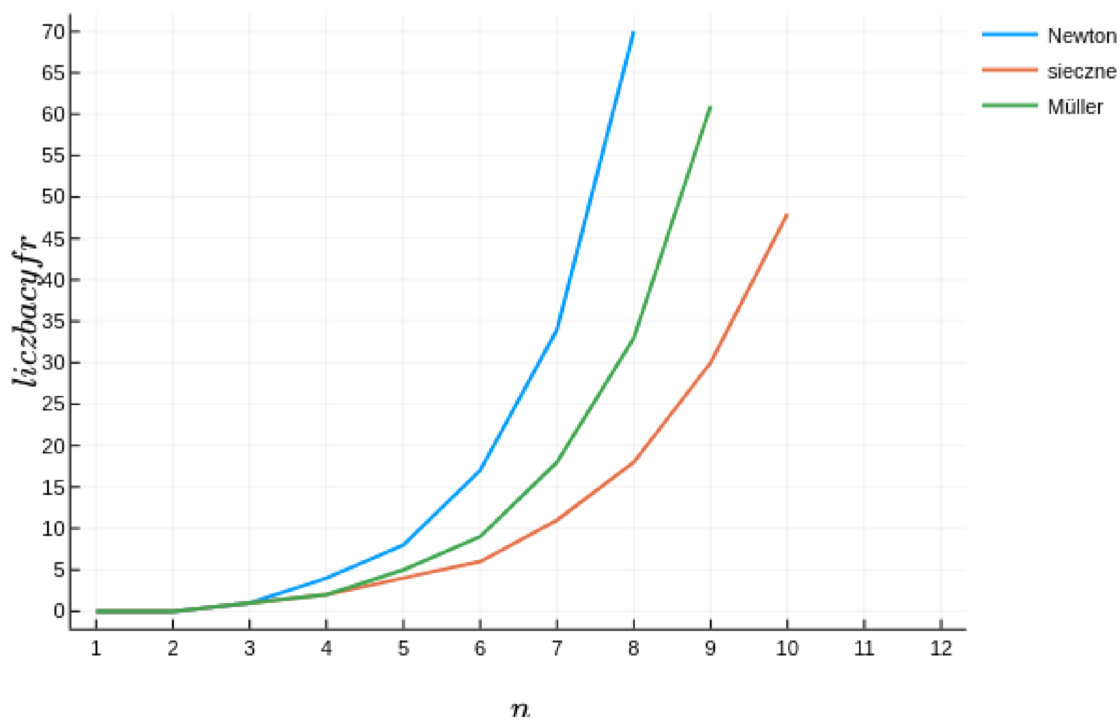
$$f'(x) = \frac{1}{x}$$

$$x_0 = \frac{1}{2}$$

Dokładna wartość miejsca zerowego: $x = 1$

n	Newton	Sieczne	Müller
0	0.0	0.0	0.0
1	0.0	0.0	0.0
2	1.0	1.0	1.0
3	4.0	2.0	2.0
4	8.0	4.0	5.0
5	1.7e+01	6.0	9.0
6	3.4e+01	1.1e+01	1.8e+01
7	7.0e+01	1.8e+01	3.3e+01
8	x^*	3.0e+01	6.1e+01
9	x^*	4.8e+01	x^*
10	x^*	x^*	x^*

Tablica 5: Porównanie liczby poprawnych cyfr w kolejnych iteracjach metod



Rysunek 8: Szybkość zbieżności wszystkich metod do miejsca zerowego $f(x) = ln|x|$

4.6 $f(x) = cos(xsinx)$

$$f(x) = cos(xsinx)$$

$$f'(x) = sin(x\,sin(x))(-(sin(x) + x\,cos(x)))$$

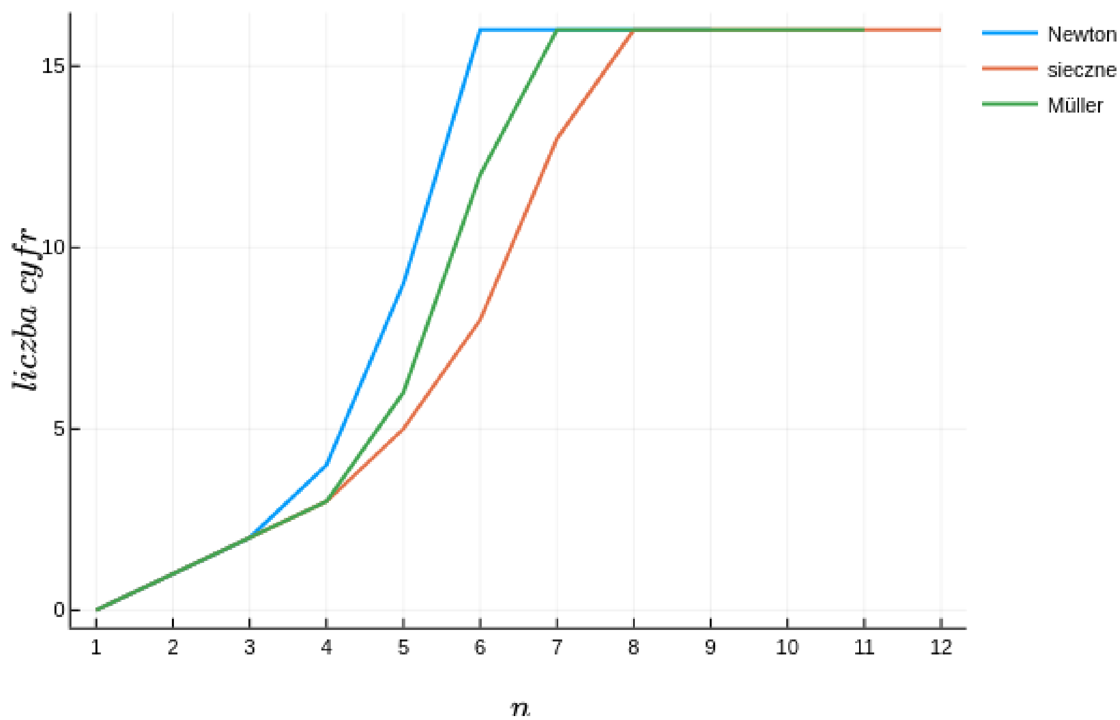
$$x_0 = -1$$

Dokładna wartość miejsca zerowego: $x = -\frac{\pi}{2}$

n	Newton	Sieczne	Müller
0	0.0	0.0	0.0
1	1.0	1.0	1.0
2	2.0	2.0	2.0
3	4.0	3.0	3.0
4	9.0	5.0	6.0
5	1.6e+01	8.0	1.2e+01
6	1.6e+01	1.3e+01	1.6e+01
7	1.6e+01	1.6e+01	1.6e+01
8	1.6e+01	1.6e+01	1.6e+01
9	1.6e+01	1.6e+01	1.6e+01
10	1.6e+01	1.6e+01	1.6e+01
11	1.6e+01	1.6e+01	1.6e+01
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Tablica 6: Porównanie liczby poprawnych cyfr w kolejnych iteracjach metod

Można zauważyć, że w tym przypadku żadna z metod nie obliczyła dokładnej wartości miejsca zerowego $-\frac{\pi}{2}$ i od pownego momentu przybliżenia przestają się poprawiać (funkcje zwracały dokładnie ten sam wynik w kolejnych iteracjach).



Rysunek 9: Szybkość zbieżności wszystkich metod do miejsca zerowego $f(x) = cos(xsinx)$

5 Podsumowanie

Zgodnie z oczekiwaniami, w niemal każdym przykładzie wyniki odzwierciedlają złożoność każdej z metod. Metoda Müllera jest zbieżna szybciej niż metoda siecznych, lecz jest deklasowana przez metodę Newtona, która ma zbieżność kwadratową. Zdecydowaną zaletą metody Müllera jest fakt, iż nie musimy znać wzoru na pochodną funkcji, której miejsce zerowe aproksymujemy. Metoda Müllera okazała się skuteczniejsza jedynie w przypadku znajdowania pierwiastków wielomianów drugiego stopnia (co jest oczywistą konsekwencją tego, iż jest ona oparta o interpolację funkcji wielomianami stopnia drugiego).

Po przeprowadzeniu powyższych obliczeń, można również eksperymentalnie wyznaczyć rząd zbieżności metody Müllera – gdy podzielimy otrzymaną dokładność przez tę, którą otrzymaliśmy jedną iterację wcześniej, otrzymany wynik będzie oscylował w okolicy 1.84. Na przykład dla $f(x) = ln|x|$ mamy

$$\frac{9}{5} = 1.8$$
$$\frac{33}{18} \approx 1.833333333$$
$$\frac{611}{33} \approx 1.848484848.$$

Powyższe obliczenia można również zastosować dla metody Newtona:

$$\frac{8}{4} = 2$$
$$\frac{17}{8} = 2.125$$
$$\frac{34}{17} = 2,$$

oraz metody siecznych:

$$\frac{18}{11} \approx 1.636363636$$
$$\frac{30}{18} \approx 1.666666667$$
$$\frac{48}{30} = 1.6.$$

Podsumowując, metoda Müllera jest bez wątpienia przydatną metodą rozwiązywania równań nielinowych, ze względu na to, iż jest stosunkowo szybka, oraz nie wymaga znajomości pochodnej. Jest ona niestety nieco bardziej skomplikowana niż pozostałe dwie metody, oraz wymaga o wiele więcej obliczeń.

Literatura

[1] Ward Cheney David Kincaid. *Analiza numeryczna.* Wydawnictwa Naukowo-Techniczne Warszawa, 2006.