

Analiza numeryczna (M) - Pracownia 1 - Zadanie P1.24

Analiza numeryczna iteracyjnej metody Bairstowa

Bartosz Bednarczyk

Jan Góra

Listopad 15, 2015

Spis treści

Przegląd podstawowych zagadnień związanych z wielomianami	2
Podstawowe definicje	2
Postać iloczynowa wielomianu i dzielenie wielomianu	2
Pochodna wielomianu i jej obliczanie	3
Inne przydatne pojęcia matematyczne	3
Metoda Newtona oraz wielowymiarowa metoda Newtona	3
Opis klasycznej metody Newtona	3
Zastosowanie klasycznej metody Newtona do szukania zer wielomianu	3
Metoda Newtona dla wielu funkcji wielu zmiennych	4
Metoda Newtona dla funkcji zespolonych	4
Wybrane metody wyszukiwania miejsc zerowych wielomianu	6
Metoda Laguerre'a	6
Metoda Mullera	6
Metoda Bairstowa	6
Opis metody Bairstowa	6
Analiza teoretyczna metody Bairstowa	7
Przykład rozbieżności metody Bairstowa	7
Implementacja metody Bairstowa	8
Proponowane usprawnienia	8
Testy numeryczne	9
Porównanie opisanych metod	9
Przykład działania opisanych metod	12

Krótki opis sprawozdania

Najprostsze metody numeryczne bardzo często okazują się być mało wydajne, dlatego matematycy dążą do uzyskania metod o bardzo niskim czasie działania. Celem tego sprawozdania jest pokazanie jednej z nich, jaką jest iteracyjna metoda Bairstowa. Na podstawie odpowiednich testów numerycznych sprawdzone zostaną dokładność, stabilność i efektywność tej metody. Poza tym pobieżnie omówione zostaną różne warianty metody Newtona, metoda Laguerre'a oraz metoda Mullera, których wydajność porównamy z metodą Bairstowa.

Przegląd podstawowych zagadnień związanych z wielomianami

Podstawowe definicje

Definicja 1. Wielomianem stopnia $n \in \mathbb{N}$ nad ciałem \mathbb{K} będziemy nazywać przekształcenie $\mathbb{K} \mapsto \mathbb{K}$ zadane wzorem $W(x) = a_0 + a_1x + \dots + a_nx^n$, gdzie a_i to pewne współczynniki z ciała \mathbb{K} oraz $a_n \neq 0$.

Definicja 2. Niech W będzie pewnym wielomianem (nad ciałem \mathbb{K}). Liczbę a taką, że $W(a) = 0$, będziemy nazywać pierwiastkiem wielomianu.

Uwaga 1. Z faktu, że wielomian W ma współczynniki z ciała \mathbb{K} , nie wynika fakt, że jego pierwiastki również będą należały do \mathbb{K} . Klasycznym przykładem jest wielomian $x^2 + 1$, który ma współczynniki rzeczywiste, a jego pierwiastkami są liczby zespolone.

Uwaga 2. Istnieją takie ciała \mathbb{K} , że dla dowolnego wielomianu stopnia większego od 0 wszystkie jego pierwiastki należą do \mathbb{K} . Ciała takie będziemy nazywać algebraicznie domkniętymi. Przykładem takiego ciała jest \mathbb{C} , czego nie będziemy dowodzić.

Podczas całego tego sprawozdania będziemy zajmować się następującym problemem:

Problem znajdowania miejsc zerowych wielomianu

Niech W będzie wielomianem. Znaleźć zbiór $\ker(W) = \{a \mid W(a) = 0\}$.

Powyższy problem, choć pozornie prosty, jest sformułowany bardzo ogólnie. Na potrzeby tej pracy od tej pory ograniczymy się tylko do ciał \mathbb{R} oraz \mathbb{C} , choć nic nie stało na przeszkodzie by poeksperymentować z innymi ciałami. Aktualnie nie wiemy czy każdy wielomian ma pierwiastki, a jeśli ma, to jaka jest moc ich zbioru. Nie znamy również żadnych metod rozwiązywania $W(x) = 0$. By lepiej zrozumieć podane zagadnienie, przejdźmy przez ciąg różnych definicji, algorytmów, twierdzeń i lematów związanych z wielomianami (warto je zrozumieć, gdyż kolejne rozdziały będą z nich korzystać).

Twierdzenie 1. Każdy wielomian $W(x)$ nad \mathbb{C} stopnia $n \in \mathbb{N}_+$ ma co najmniej jeden pierwiastek.

Dowód. To twierdzenie jest nazywane zasadniczym twierdzeniem algebry. Udowodnione w [1], s. 105. \square

Wniosek 1. $|\ker(W)| \leq n$, gdzie n to stopień wielomianu W .

Postać iloczynowa wielomianu i dzielenie wielomianu

Definicja 3. Wielomian $W(x)$ nazywamy podzielny przez wielomian $P(x)$, różny od wielomianu zerowego, wtedy i tylko wtedy, gdy istnieje taki wielomian $Q(x)$, że $W(x) = Q(x) \cdot P(x)$. Wielomian $Q(x)$ nazywamy ilorazem wielomianu $W(x)$ przez $P(x)$. Mówimy, że wielomian $P(x)$ jest dzielnikiem wielomianu $W(x)$.

Twierdzenie 2. Dowolny wielomian $W(x)$ możemy zapisać jako $W(x) = P(x) \cdot Q(x) + R(x)$ dla pewnych wielomianów P, Q, R . Mówimy, że wielomian $W(x)$ jest podzielny przez $Q(x)$, jeżeli $R(x) = 0$.

Twierdzenie 3. Wielomian $W(x)$ jest podzielny przez wielomian $Q(x) = (x - a)$ wtedy i tylko wtedy, gdy $W(a) = 0$.

Dowód. Udowodnione w [2], s. 198-199. \square

Chcielibyśmy umieć w efektywny sposób realizować procedurę dzielenia wielomianu przez jednomiany postaci $x - a$. Służy do tego następujący algorytm:

1. $P(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$.

2. Niech $\alpha = a_n$.
3. Kolejno dla $k = n - 1, n - 2, \dots, 0$ wykonaj $\alpha := a_k + x\alpha$.
4. Wynik to $p(x) = \alpha$.

Dokładny opis metody oraz jej analizę możemy znaleźć w [3], s. 103.

Pochodna wielomianu i jej obliczanie

Definicja 4. Pochodną wielomianu $p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ będziemy nazywać wielomian $p'(x) = n \cdot a_n x^{n-1} + (n-1) a_{n-1} x^{n-2} + \dots + a_1$.

Wyznaczanie wielomianu w punkcie x_0 możemy zrealizować za pomocą schematu Hornera:

1. $P(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$.
2. Niech $\alpha := a_n, \beta := 0$.
3. Kolejno dla $k = n - 1, n - 2, \dots, 0$ wykonaj $\beta := \alpha + x\beta, \alpha := a_k + x\alpha$.
4. Wynik to $p'(x) = \beta$.

Inne przydatne pojęcia matematyczne

Oprócz wymienionych w rozdziale pojęć związanych z wielomianami, zakładać będziemy u czytelnika znajomość wielowymiarowego rachunku różniczkowego, definicji funkcji holomorficznej oraz podstawowych pojęć związanych z analizą błędów. Pojęcia te można doczytać w [1] i [3].

Metoda Newtona oraz wielowymiarowa metoda Newtona

Opis klasycznej metody Newtona

Klasyczną metodą Newtona zastosowaną dla pewnego punktu startowego p oraz funkcji $f : \mathbb{R} \mapsto \mathbb{R}$ klasy C^1 nazywać będziemy metodę iteracyjną postaci:

$$x_n = \begin{cases} p, & \text{gdy } n = 0 \\ x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}, & \text{w.p.p.} \end{cases}$$

Można pokazać, że x_n zbiega do pewnego pierwiastka funkcji f . Analizę klasycznej metody Newtona można znaleźć w [3], s. 71-81.

Zastosowanie klasycznej metody Newtona do szukania zer wielomianu

Jeśli mamy wielomian o współczynnikach i pierwiastkach rzeczywistych, możemy policzyć jego pierwiastki za pomocą klasycznej metody Newtona. Podstawiamy za f z poprzedniego opisu nasz wielomian, a f' to jego pochodna. Po znalezieniu jednego pierwiastka (nazwijmy go a) dzielimy nasz wielomian przez $x - a$ i uruchamiamy program dla otrzymanego wielomianu. Proces kontynuujemy tak długo, aż dojdziemy do wielomianu o stopniu 0.

Uwaga 3. Poniżej przedstawiamy przykładową implementację metody Newtona w języku Julia. Wartość w punkcie wielomianu i jego pochodnej możemy wyznaczyć z pomocą schematu Hornera, który był omówiony wcześniej (w kodzie przykładowym skorzystaliśmy z funkcji bibliotecznych dla większej czytelności).

```

1 using Polynomials
2 # W - wielomian, n - stopień wielomianu, p - punkt startowy, eps - dokładność
3
4 function klasyczna_metoda_newtona(W, n, p, eps)
5     dW = polyint(W) # oblicza pochodną wielomianu
6     x'n = p
7
8     while bigger(abs(polyval(W, x'n)), eps)
9         x'n = x'n - (polyval(W, x'n)/polyval(dW, x'n))
10    end
11
12    return x'n # zwroc szukany pierwiastek
13 end

```

Uwaga 4. Powyższa metoda nie nadaje się do obliczania miejsc zerowych wielomianu, którego pierwiastki są zespolone (z powodu tego, że operujemy tutaj na tylko rzeczywistych przybliżeniach x_n).

Metoda Newtona dla wielu funkcji wielu zmiennych

Załóżmy, że mamy do rozwiązania układ równań:

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \dots \\ f_n(x_1, x_2, \dots, x_n) = 0, \end{cases}$$

gdzie $f_i \in \mathbb{R}^n \mapsto \mathbb{R}$ jest klasy C^1 .

Każdą z tych funkcji możemy rozisać ze wzoru Taylora jako:

$$0 = f_i(x_1 + h_1, x_2 + h_2, \dots, x_n + h_n) \approx f_i(x_1, x_2, \dots, x_n) + \sum_{j=1}^n h_j \cdot \frac{\partial f_i}{\partial x_j}(x_1, x_2, \dots, x_n)$$

Powyższy układ możemy zapisać w postaci macierzowej:

$$\begin{pmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \dots \\ f_n(x_1, x_2, \dots, x_n) \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x_1, x_2, \dots, x_n) & \frac{\partial f_1}{\partial x_2}(x_1, x_2, \dots, x_n) & \dots & \frac{\partial f_1}{\partial x_n}(x_1, x_2, \dots, x_n) \\ \frac{\partial f_2}{\partial x_1}(x_1, x_2, \dots, x_n) & \frac{\partial f_2}{\partial x_2}(x_1, x_2, \dots, x_n) & \dots & \frac{\partial f_2}{\partial x_n}(x_1, x_2, \dots, x_n) \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1}(x_1, x_2, \dots, x_n) & \frac{\partial f_n}{\partial x_2}(x_1, x_2, \dots, x_n) & \dots & \frac{\partial f_n}{\partial x_n}(x_1, x_2, \dots, x_n) \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ \dots \\ h_n \end{pmatrix}$$

Aby nieco skrócić ten układ, będziemy go zapisywać jako $F(X) = -J \cdot H$. Jeśli macierz J jest nieosobliwa, to układ ma rozwiązanie w postaci:

$$-J^{-1} \cdot F(X) = H$$

Ostatecznie wzór Newtona dla układu funkcji wielu zmiennych możemy wyrazić wzorem:

$$X_{k+1} = X_k + H_k = X_k - J^{-1}(X_k)F(X_k)$$

Metoda Newtona dla funkcji zespolonych

Lemat 1. Dowolną funkcję analityczną $f : \mathbb{C} \mapsto \mathbb{C}$ możemy zapisać jako

$$f(z) = f(x + yi) = P(x, y) + iQ(x, y),$$

gdzie $x, y \in \mathbb{R}$, $P(x, y) \in \mathbb{R}$, $Q(x, y) \in \mathbb{R}$

Przykład 1.

$$f(z) = z^3 - 2z = f(x + iy) = (x + iy)^3 - 2(x + iy) = (x^3 - 3xy^2 - 2x) + i(3x^2y - y^3 - 2y) = P(x, y) + iQ(x, y)$$

Niech $f(z) = P(x, y) + iQ(x, y)$. Równanie $f(z) = 0$ możemy sprowadzić do układu równań $Q(x, y) = 0$ i $P(x, y) = 0$. Taki układ równań rozwiązujemy za pomocą metody Newtona dla funkcji wielu zmiennych.

$$v_{n+1} = v_n - \frac{f(v_n)}{f'(v_n)}$$

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} - J^{-1} \begin{pmatrix} P(x_n, y_n) \\ Q(x_n, y_n) \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} - \begin{pmatrix} \frac{\partial P}{\partial x}(x_n, y_n) & \frac{\partial P}{\partial y}(x_n, y_n) \\ \frac{\partial Q}{\partial x}(x_n, y_n) & \frac{\partial Q}{\partial y}(x_n, y_n) \end{pmatrix}^{-1} \begin{pmatrix} P(x_n, y_n) \\ Q(x_n, y_n) \end{pmatrix}$$

Ponieważ wielomian jest funkcją holomorficzną, to zachodzi równanie Cauchy'ego-Riemanna:

$$\frac{\partial P}{\partial x} = \frac{\partial Q}{\partial y}, \quad -\frac{\partial P}{\partial y} = \frac{\partial Q}{\partial x}$$

Oznaczając $P = P(x_n, y_n)$, $Q = Q(x_n, y_n)$, $P_x = \frac{\partial P}{\partial x}(x_n, y_n)$, $Q_x = \frac{\partial Q}{\partial x}(x_n, y_n)$ oraz korzystając ze wzoru na macierz odwrotną możemy uprościć wzór na metodę Newtona do postaci:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n - \frac{PP_x + QQ_x}{P_x^2 + Q_x^2} \\ y_n - \frac{PP_y + QQ_y}{P_x^2 + Q_x^2} \end{pmatrix}$$

Uwaga 5. Implementacja zespolonej metody Newtona jest nieco problematyczna. Musimy potrafić zaimplementować operacje na funkcjach wielu zmiennych oraz ich różniczkowanie. Przykładowy kod w języku Julia możemy otrzymać dzięki zastosowaniu biblioteki MultiPoly. Wymieniona biblioteka dostarcza nam sposobu na tworzenie nowych zmiennych wielomianu (metoda generators), obliczania wartości w danym punkcie (evaluate) oraz efektywnego różniczkowania wielomianu po zadanej zmiennej (diff).

```

1 using MultiPoly # biblioteka dla wielomianow wielu zmiennych
2
3 # a - tablica wspolczynnika, x0+i*y0 -punkt startowy, eps - precyzja
4 function complex'newton(a, n::Int, x0::Float64, y0::Float64, eps::Float64)
5
6     if n == 0
7         return
8     elseif n == 1
9         w = Complex128(-W[1])/Complex128(W[2])
10        @printf("%0.16lf + "t %0.16lf i"n ", real(w), imag(w))
11        return
12    end
13    # x, y - zmienne w wielomianie
14    x, y = generators(MPoly-Float64", :x, :y); p = zero(MPoly-Complex128")
15
16    for i in 1:(n+1) # zamiana zespolonego wielomianu na funkcje dwu zmiennych
17        p = p + W[i] * (x+y*im)^(i-1)
18    end
19
20    xn = x0; yn = y0; P = real(p); Q = imag(p-P)
21    Px = diff(P, :x); Py = diff(P, :y); Qx = diff(Q, :x); Qy = diff(Q, :y)
22
23    while( bigger(abs( evaluate(p, xn, yn) ),eps )
24        eP = evaluate(P, xn, yn); eQ = evaluate(Q, xn, yn); ePx = evaluate(Px, xn, yn);
25        eQy = evaluate(Qy, xn, yn); ePy = evaluate(Py, xn, yn); eQx = evaluate(Qx, xn, yn)
26        xn = xn - (eP * ePx + eQ * eQx)/(ePx^2 + eQx^2); yn = yn - (eP * ePy + eQ * eQy)/(ePx^2 + eQx^2)
27    end
28
29    @printf("%0.16lf + "t %0.16lf i"n ", xn, yn)
30    complex'newton(divide(W, complex(xn, yn)), n-1, x0, y0, eps)
31 end

```

Wybrane metody wyszukiwania miejsc zerowych wielomianu

Metoda Laguerre'a

Jedną z metod iteracyjnych wyszukiwania pierwiastków wielomianu używanych w nowoczesnych systemach informatycznych jest metoda Laguerre'a.

Niech $p(z)$ będzie wielomianem stopnia n , którego pierwiastki mamy znaleźć. Kolejne kroki w metodzie wykonujemy za pomocą następujących wzorów:

$$A = \frac{-p'(z)}{p(z)}, \quad B = A^2 - \frac{p''(z)}{p(z)}, \quad C = \frac{A \pm \sqrt{(n-1)(nB - A^2)}}{n}, \quad z_{\text{nowe}} = z + \frac{1}{C}$$

Metoda Laguerre'a jest bardzo efektywnym algorytmem, ponieważ w okolicach pojedynczego pierwiastka wielomianu p jest zbieżna sześciennie. Dokładną analizę tej metody pozostawiamy czytelnikowi do przeczytania w [3], s. 112-116.

Metoda Mullera

Metoda Mullera jest modyfikacją metody stycznych. Zamiast przybliżać nasz wielomian f funkcją liniową, będziemy go aproksymować funkcją kwadratową.

Rozważmy trzy punkty x_0, x_1, x_2 wraz z wartościami funkcji f w tych punktach. Przyjmujemy, że x_2 jest aktualnym przybliżeniem rozwiązania. Oznaczmy $z = x - x_2$, $h_0 = x_0 - x_2$, $h_1 = x_1 - x_2$.

Oznaczmy szukaną parabolę przez $g(z) = az^2 + bz + c$. Z definicji paraboli w punkcie $z = x_k$ dostajemy, że

$$2a = f''(x_k), \quad b = f'(x_k), \quad c = f(x_k),$$

co prowadzi do wzoru

$$x_{k+1} = x_k - \frac{2f(x_k)}{f'(x_k) + \operatorname{sgn}(f'(x_k)) \cdot \sqrt{(f'(x_k))^2 - 2f(x_k)f''(x_k)}}$$

Więcej na temat tej metody oraz jej modyfikacji można poczytać w skrypcie [8].

Metoda Bairstowa

Opis metody Bairstowa

Ostatnią i zarazem najważniejszą metodą, którą omówimy w sprawozdaniu, będzie metoda Bairstowa. Wiemy, że nawet jeśli wielomian ma współczynniki rzeczywiste, to może mieć pierwiastki zespolone (np. $x^2 + 1$). Metoda Bairstowa pozwala na obliczenie wszystkich pierwiastków bez użycia arytmetyki zespolonej.

Lemat 2. *Jeżeli w jest pierwiastkiem nierzeczywistym wielomianu $p(z)$, a $p(z)$ jest wielomianem o współczynnikach rzeczywistych, to pierwiastkiem $p(z)$ jest również \bar{w} . Iloczyn $(x - w)(x - \bar{w})$ jest czynnikiem kwadratowym o współczynnikach rzeczywistych.*

Dowód. Udowodnione w [3], s. 108. □

Zauważmy, że pierwiastki zespolone możemy wyszukiwać parami. Zamiast wyszukiwać pierwiastki pojedynczo, będziemy wyszukiwać dwumianu postaci $z^2 - uz - v$.

Lemat 3. *Dowolny wielomian $p(z) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$ możemy zapisać w postaci:*

$$p(z) = (b_n x^{n-2} + b_{n-1} x^{n-3} + \dots + b_3 z + b_2) (z^2 - uz - v) + b_1 (z - u) + b_0$$

Wielomian $(b_n x^{n-2} + b_{n-1} z^{n-3} + \dots + b_3 z + b_2)$ będziemy dalej oznaczać jako $Q(z, u, v)$. Powyższe współczynniki możemy obliczać rekurencyjnie według wzorów:

$$b_{n+1} = b_{n+2} = 0, \quad b_k = ub_{k+1} + vb_{k+2} \quad (n \geq k \geq 0).$$

Dowód. Dowód w [3], s. 109. □

Chcemy, by nasz wyjściowy wielomian był podzielny przez $z^2 - uz - v$. Zatem musi zachodzić $b_0 = b_1 = 0$. Potraktujmy podane współczynniki jako funkcje zmiennych u, v . Wtedy dostajemy do rozwiązania układ równań:

$$\begin{cases} b_0(u + h_1, v + h_1) = 0 \\ b_1(u + h_2, v + h_2) = 0 \end{cases}$$

Podany układ możemy rozwiązać przedstawioną wcześniej metodą Newtona dla wielu funkcji wielu zmiennych. Po znalezieniu współczynników u, v dzielimy wyjściowy wielomian przez otrzymany dwumian i kontynuujemy proces wyszukiwania pierwiastków dla mniejszego wielomianu (z uwzględnieniem tego, że przypadki dla wielomianu stopnia 0 i 1 traktujemy osobno).

Analiza teoretyczna metody Bairstowa

Lemat 4. *Metoda Bairstowa jest zbieżna lokalnie.*

Dowód. Wynika to bezpośrednio z tego, że metoda Newtona jest zbieżna lokalnie. □

Głównym założeniem w lokalnej zbieżności metody Bairstowa jest to, że jacobian wyliczany przy metodzie Newtona się nie zeruje dla podanych wcześniej punktów startowych i kolejnych przybliżeń. Zastanówmy się w jaki sposób zerowanie się jacobianu zależy od punktów startowych oraz pierwiastków wielomianu.

Twierdzenie 4. *Niech u, v będą dowolnie wybranymi liczbami rzeczywistymi. Jacobian dla algorytmu Bairstowa jest macierzą odwracalną wtedy i tylko wtedy, gdy $z^2 - uz - v$ oraz wielomian $Q(z, u, v)$ nie mają wspólnych pierwiastków. Rząd jacobianu jest jeden wtedy i tylko wtedy, kiedy liczba wspólnych pierwiastków (z krotnościami) jest równa jeden. Jacobian się zeruje wtedy i tylko wtedy, gdy $z^2 - uz - z$ dzieli $Q(z, u, v)$.*

Twierdzenie 5. *Załóżmy, że $P(z) = Q(z, u, v)(x^2 - u^*z - v^*)$ i załóżmy, że wyrażenia po prawej stronie nie mają wspólnego pierwiastka. Wtedy istnieje dodatnia liczba d taka, że ciąg (u_k, v_k) generowany przez metodę Bairstowa jest zbieżny kwadratowo do (u^*, v^*) , gdzie $|u_0 - u^*| < d \wedge |v_0 - v^*| < d$.*

Dowód. Twierdzenia te zostały udowodnione przez autorów Tibora Fiałę oraz Annę Krebsz w 1987 roku. Kompletne dowody można przeczytać w [5]. □

Analiza zbieżności oraz rozbieżności metody Bairstowa stanowiła podstawę do napisaniu kilku (choć niestety niewielu) prac naukowych. Zainteresowanego czytelnika odsyłamy do [5], [6] oraz [7].

Przykład rozbieżności metody Bairstowa

Rozważmy wielomian postaci $P(x) = (x^2 + ux + v)(x^2 + ux + w) + (w - v)^2$. Jacobian dla metody Bairstowa w punkcie (u, v) będzie wyglądał następująco:

$$J(u, v) = \begin{pmatrix} v - w & 0 \\ 0 & v - w \end{pmatrix}$$

Jeśli uruchomimy metodę Bairstowa dla np. $u = 3, v = 1, w = 2$ to dostaniemy wielomian $x^4 + 6x^3 + 12x^2 + 9x + 3$, dla którego ciąg przybliżeń pierwiastka będzie cykliczny.

Tabela 1: Metoda Bairstowa dla powyższej funkcji

Iteracja	x_0	x_1	x_2	x_3
1	-2.6180339884	-3.8196601113e-01	-2.6180339884	-3.8196601113e-01
2	-2.0	-1.0	-2.0	-1.0
3	-2.6180339884	-3.8196601113e-01	-2.6180339884	-3.8196601113e-01
4	-2.0	-1.0	-2.0	-1.0

Implementacja metody Bairstowa

Jeżeli oznaczymy sobie $c_k := \frac{\partial b_k}{\partial u}$, $d_k := \frac{\partial b_{k-1}}{\partial v}$ to dostajemy związki:

$c_k = d_{k+1} + uc_{k+1} + vc_{j+2}$ ($c_{n+1} = c_n = 0$) oraz $d_k = b_{k+1} + ud_{k+1} + vd_{k+2}$ ($d_{n+1} = d_n = 0$).

Rozwiązujemy układ równań

$$\begin{cases} b_0(u, v) + \frac{\partial b_0}{\partial u} \delta u + \frac{\partial b_0}{\partial v} \delta v = 0 \\ b_1(u, v) + \frac{\partial b_1}{\partial u} \delta u + \frac{\partial b_1}{\partial v} \delta v = 0 \end{cases}$$

Rozwiązaniem powyższego układu jest

$$\delta u = (c_1 b_1 - c_2 b_0)/J, \quad \delta v = (c_1 b_0 - c_0 b_1)/J, \quad J = c_0 c_2 - c_1^2$$

Metodę Bairstowa możemy zapisać w postaci listy kroków:

1. $b_n := a_n$
2. $c_n := 0$
3. $c_{n-1} := a_n$
4. for $j = 1$ to M do
 - (a) $b_{n-1} := a_{n-1} + ub_n$
 - (b) for $k = n - 2$ to 0 step -1 do
 - i. $b_k := a_k + ub_{k+1} + vb_{k+2}$
 - ii. $c_k := b_{k+1} + uc_{k+1} + vc_{k+2}$
 - (c) end do
 - (d) $J := c_0 c_2 - c_1^2$
 - (e) $u := u + (c_1 b_1 - c_2 b_0)/J$
 - (f) $v := v + (c_1 b_0 - c_0 b_1)/J$
 - (g) output j, u, v, b_0, b_1
5. end do

Pseudokod zapożyczony z opisu metody Bairstowa z [3].

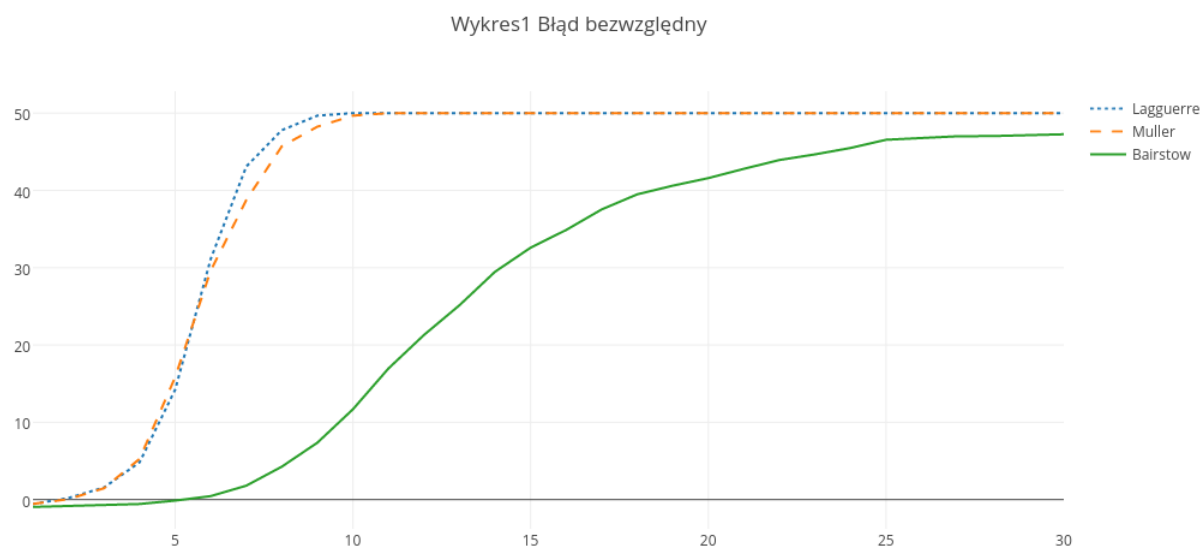
Proponowane usprawnienia

- W przypadku, kiedy po podanej liczbie iteracji nasz (u, v) nie "wyzerował" dostatecznie współczynników b_0 oraz b_1 to możemy wylosować nowy punkt startowy (u^*, v^*) i zacząć metodę od początku. Ominiemy wtedy przypadek, że dla podanych początkowych punktów metoda się zapętla (patrz: s. 7).
- Jeżeli w danej iteracji jacobian się zeruje, a b_1, b_0 nie są "wyzerowane", to możemy również wylosować nowy punkt i zacząć metodę od początku.

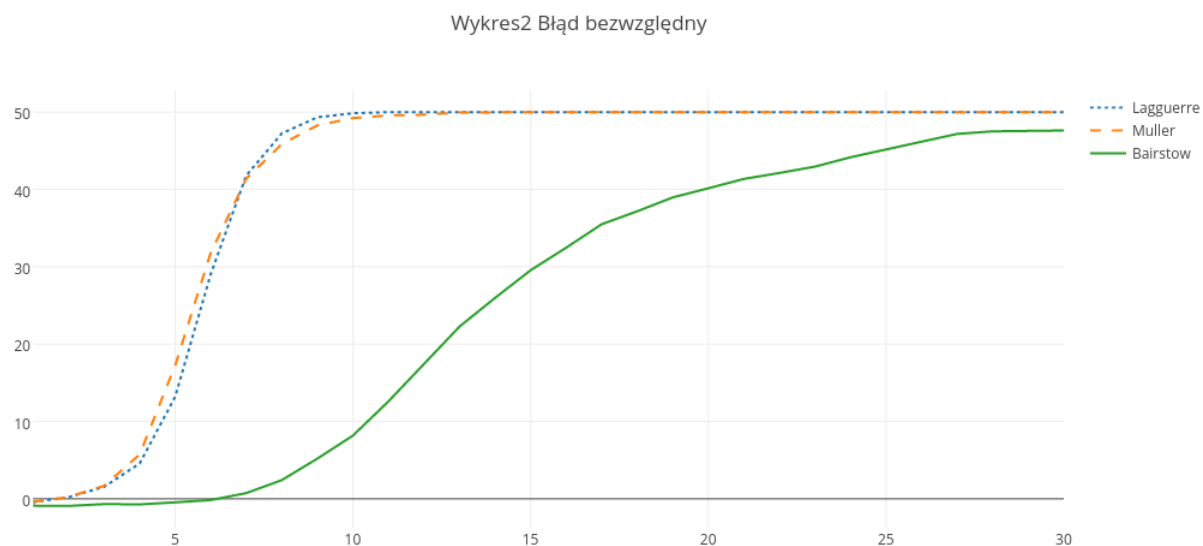
Testy numeryczne

Porównanie opisanych metod

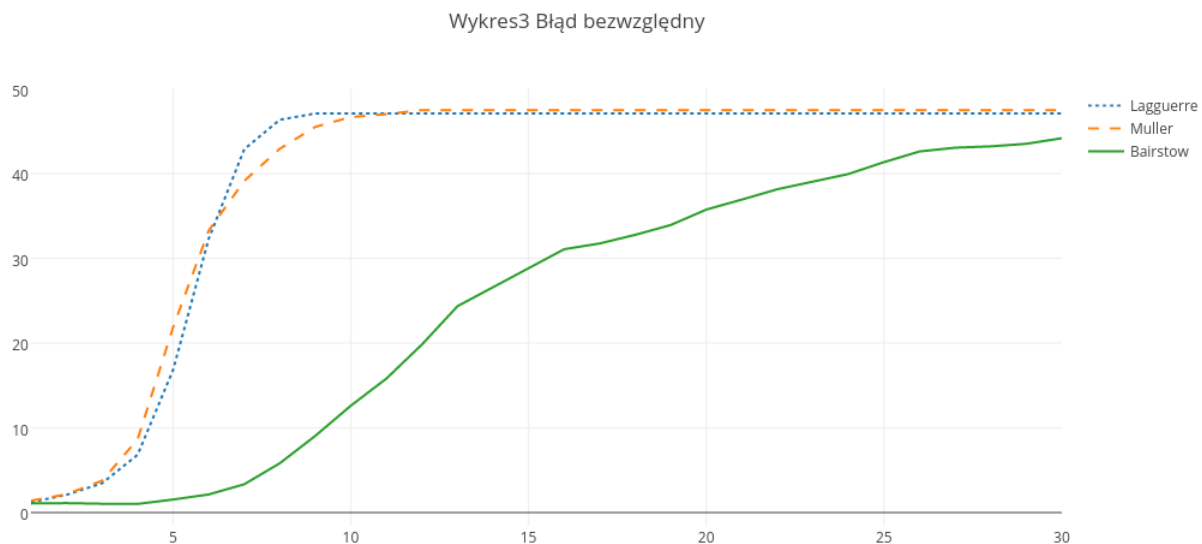
Poniższe wykresy prezentują obserwację zmiany błędu bezwzględnego pierwiastka wielomianu i przybliżenia zadanego przez opisaną metodą w kolejnych iteracjach na podstawie różnych danych wejściowych. Na wykresie oś Y przedstawia błąd bezwzględny w postaci 10^{-y} .



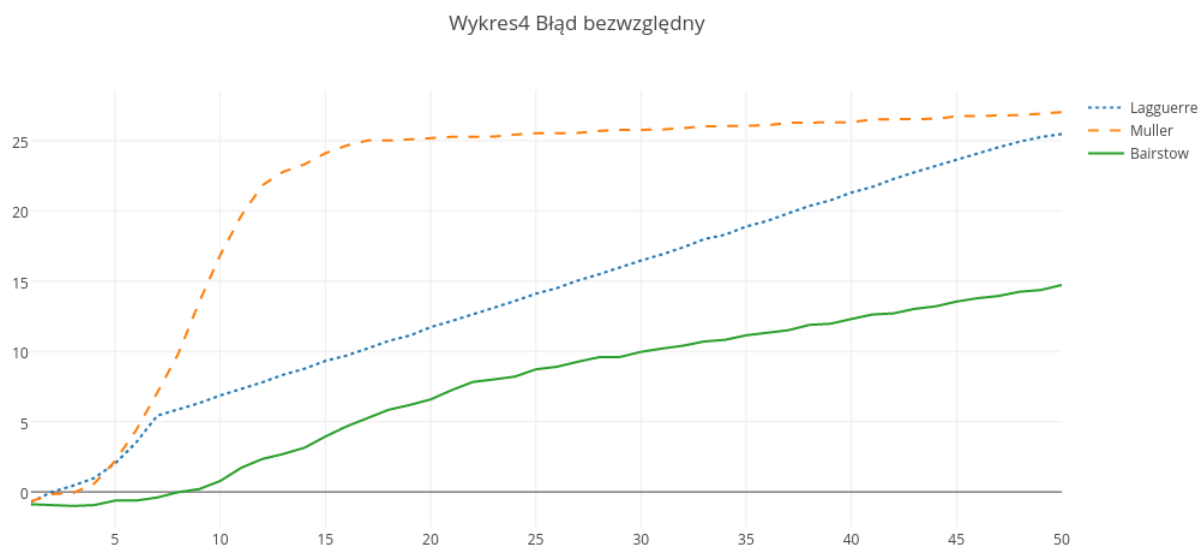
Wykres 1: Próba 100 wielomianów 10-tego stopnia o losowych pierwiastkach typu *BigFloat* (o precyzji 10^{-30}) w zakresie $[-7; 7]$ w zerowych punktach początkowych.



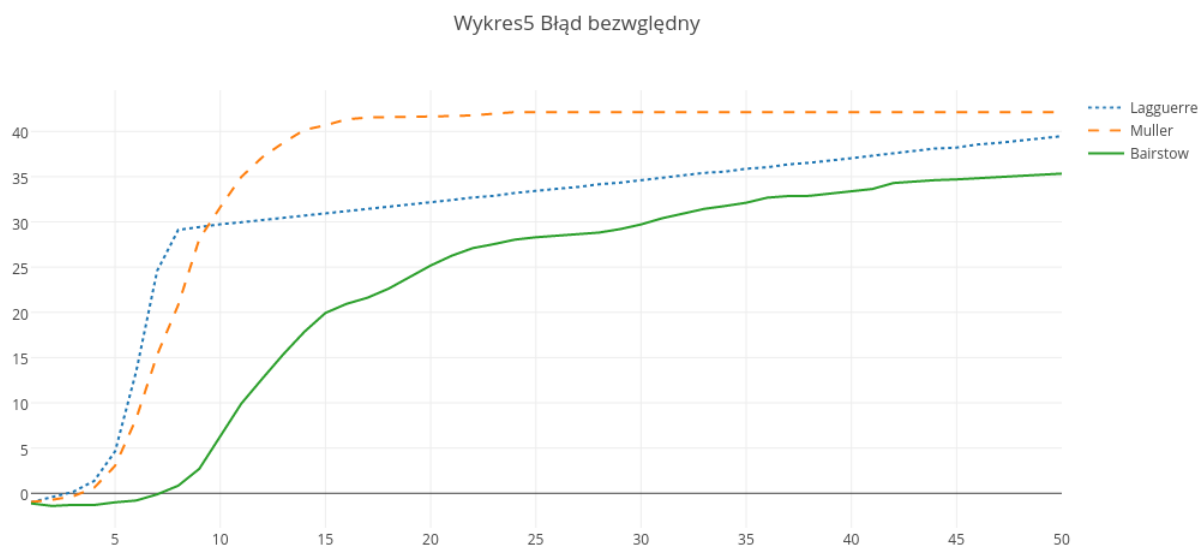
Wykres 2: Próba 100 wielomianów 10-tego stopnia o losowych pierwiastkach typu *BigFloat* (o precyzji 10^{-30}) w zakresie $[-7; 7]$ w zerowych punktach początkowych.



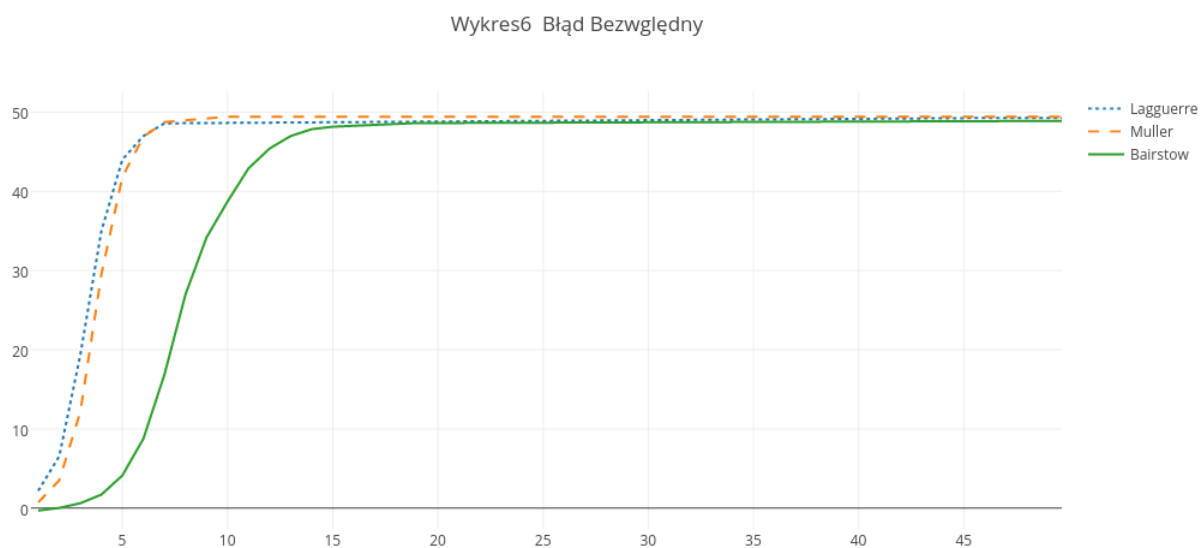
Wykres 3: Próba 50 wielomianów 10-tego stopnia o losowych pierwiastkach typu *BigFloat* (o precyzji 10^{-30}) w zakresie $[0; 1)$ w zerowych punktach początkowych.



Wykres 4: Próba 50 wielomianów 10-tego stopnia o losowych pierwiastkach typu *Int64* w zakresie $[-10; 10]$ w zerowych punktach początkowych.



Wykres 5: Próba 50 wielomianów 10-tego stopnia o losowych pierwiastkach typu $Int64$ w zakresie $[-30, 30]$ w zerowych punktach początkowych.



Wykres 6: Próba 50 wielomianów 3-tego stopnia o losowych pierwiastkach typu $Int64$ w zakresie $[-30, 30]$ w zerowych punktach początkowych.

Wniosek 2. Wykresy 1 i 2 oparte były na podobnych danych. Zauważamy, że funkcje na różnych wielomianach o tym samym typie zbiegają mniej-więcej w ten sam sposób. Widzimy, że metoda Bairstowa jest zdecydowanie wolniej zbieżna od dwóch pozostałych metod.

Na wykresie 3 widać, że pomimo pozornie bliskich pierwiastków wielomianu $([0, 1])$ metody są równie szybko zbieżne.

Na wykresie 4 widzimy, że bliskie pierwiastki są dużym utrudnieniem dla metod. Błąd bezwzględny przy 50 iteracjach nie przekroczył 10^{-30} , podczas gdy w poprzednich przykładach już po 30 iteracjach metody osiągnęły maksymalną dokładność.

Po zwiększeniu przedziału do $[-30; 30]$ metody znów szybko osiągnęły dobrą dokładność.

Ostatni wykres prezentuje zachowanie metod na bardzo małych danych wejściowych. Widać, że wszystkie metody są bardzo szybko zbieżne.

Przykład działania opisanych metod

Przykład 2. Metoda Bairstowa zastosowana do wielomianu

$$168 + 10x - 37x^2 + 2x^3 + x^4 = (x + 2)(x - 3)(x + 7)(x - 4)$$

1. Iteracja nr 1:

-1.917879829307739159294843574606, 2.441230459404106839057630824421,
-7.164414002968976450250192381454, 5.164414002968976450250192381454

2. Iteracja nr 2:

-2.009530033377617967339383794074, 2.900360110039499302979214698574
-6.970374011038727308629332841661, 4.447023380942359628866545591846

3. Iteracja nr 3:

-2.001038440292382015085002297150, 2.994714506062211330442943300673
-6.984875797781202526521865011540, 4.094045721119321190882034107040

4. Iteracja nr 4:

-2.000004229190859710286686827364, 2.999983492821811977876729816709
-6.998950445144011486784616660300, 4.005274379374182171426675656777

5. Iteracja nr 5:

-2.000000000049071271087627181384, 2.999999999847921708048370785679
-6.999996192489175048447504955397, 4.000016928858222780857461966052

- [4] Lily Yau, Adi Ben-Israel, *The Newton and Halley Methods for Complex Roots*, The American Mathematical Monthly 105, 1998, s. 806–818.
- [5] Tibor Fiala, Anna Krebsz, *On the Convergence and Divergence of Bairstow's Method*, Journal Numerische Mathematik, Volume 50 Issue 4, 1987, s. 477-482.
- [6] Wolfgang Gabler *Invariances and convergence properties of Bairstow's method*, International Journal of Pure and Applied Mathematics Volume 27 No. 4, 2006, s. 523-576.
- [7] Sofo, Anthony and Glasson, Alan, *Singularities in Bairstow's method*, Gazette of the Australian Mathematical Society, 37 (2), s. 93-100.
- [8] Piotr Tatjewski, *Równania nieliniowe i zera wielomianów*, Skrypt do wykładu Metody Numeryczne.