

# Wyznaczanie zer wielomianu metodą Bairstowa

Barbara Seweryn, Michał Wieteki

4 grudnia 2023 r.

## 1 Opis metody

Metoda Bairstowa służy do obliczania pierwiastków rzeczywistych oraz zespolonych wielomianów o współczynnikach rzeczywistych. Pierwiastki zespolone oblicza się parami korzystając z własności, że jeśli liczba zespolona  $z$  jest pierwiastkiem wielomianu rzeczywistego  $w(x)$  to jej sprzężenie, a więc  $\bar{z}$  również jest pierwiastkiem tego wielomianu.

Skoro  $z$  oraz  $\bar{z}$  są różnymi pierwiastkami wielomianu  $w(x)$  to wielomian ten ma czynnik kwadratowy

$$(x - z)(x - \bar{z}) = x^2 - (z + \bar{z})x + z\bar{z}$$

gdzie  $(z + \bar{z})$  oraz  $(z\bar{z})$  są liczbami rzeczywistymi.

Weźmy wielomian

$$w(x) = a_n x^{n-1} + a_{n-1} x^{n-2} + \dots + a_2 x + a_1$$

Podzielenie go przez wielomian kwadratowy  $x^2 - ux - v$  daje:

$$q(x) = b_n x^{n-3} + b_{n-1} x^{n-4} + \dots + b_4 x + b_3$$

$$r(x) = b_2(x - u) + b_1$$

czyli odpowiednio iloraz i resztę z dzielenia. Współczynniki  $b_k$  dla  $k \in \{1, 2, \dots, n\}$  można obliczać za pomocą rekurencji ze wzoru:

$$b_{n+1} = b_{n+2} = 0$$

$$b_k = a_k + u b_{k+1} + v b_{k+2}, \quad k \in \{1, 2, \dots, n\}$$

(zdefiniowanie  $b_{n+1}$  oraz  $b_{n+2}$  jako zera pomoga w pisaniu algorytmu, ponieważ nie trzeba wyróżniać przypadków gdy  $k = n - 1$  lub  $k = n$ ).

W naszej metodzie chcemy, aby wcześniej przytoczony wielomian  $q(x)$  był czynnikiem wielomianu  $w(x)$ , a co za tym idzie, żeby wielomian  $r(x)$  był zerowy. Zatem dążymy do tego, aby współczynniki  $b_2$  oraz  $b_1$  były równe zero. Jako że są one zależne od  $u$  i  $v$ , to możemy je zapisać jako funkcje od tych zmiennych, czyli:

$$b_2 = b_2(u, v), \quad b_1 = b_1(u, v)$$

Obliczamy zatem pochodne cząstkowe  $b_k$  z wcześniej wypisanych związków rekurencyjnych i dostajemy dwa takie same rekurencyjne związki na pochodne cząstkowe  $b_k$  po  $u$  i  $v$ .

Po paru przekształceniach, które dokładniej przedstawione są w książce, w której znaleźliśmy opis tej metody, dostajemy wzory na odpowiednio, błędy początkowych (dla każdego kroku metody Bairstowa) wartości  $u$  i  $v$ :

$$\delta u = \frac{(c_2 b_2 - c_3 b_1)}{J} \quad \delta v = \frac{(c_2 b_1 - c_1 b_2)}{J}$$

gdzie  $J = c_1 c_3 - c_2^2$  jest Jakobianem pary funkcji  $b_1(u, v), b_2(u, v)$ . Gdy funkcja, po wykonaniu odpowiedniej liczby kroków, otrzymuje przybliżone  $u$  i  $v$ , należy je wyliczyć (trzeba sprawdzić czy są rzeczywiste, czy zespolone), a następnie przeprowadzić podobne kroki dla wielomianu  $q(x)$ , który jest wynikiem dzielenia  $w(x)$  przez czynnik kwadratowy z naszymi obliczonymi  $u$  i  $v$  jako pierwiastkami (czyli po prostu wielomianu o współczynnikach  $b_k$  dla  $k \in \{3, 4, \dots, n\}$ ).

## 2 Opis programów obliczeniowych

Nasz program obliczeniowy składa się z 8 skryptów:

### 1. funkcja *bairstow.m*

Jest to funkcja, która odpowiada za główny algorytm obliczania jednej pary pierwiastków. Przyjmuje ona  $a$  czyli wektor rzeczywistych współczynników  $a_k$  wielomianu, którego pierwiastki chcemy wyznaczyć, oraz wartości początkowego przybliżenia  $u$  i  $v$ . Funkcja ta zwraca krotkę wartości  $u\_r, v\_r, b1$  oraz  $b2$  (wektory wartości  $u, v, b_1$  oraz  $b_2$  obliczonych w każdej iteracji pętli *while*) oraz wektor  $b\_r$ , czyli wektor współczynników wielomianu  $q(x)$  i liczbę iteracji. Algorytm ten jest przeniesiem metody Bairstowa do środowiska MatLab. Każda kolejna iteracja pętli *while* jest jednym krokiem metody Bairstowa, a sama pętla wykonuje się do momentu, gdy wartości  $b_1$  oraz  $b_2$  nie osiągną naraz zadowalającego przybliżenia do 0, które w naszej funkcji ustawiliśmy na  $10^{-20}$ .

## 2. funkcja *zerowe.m*

Funkcja odpowiadająca za obliczenie wszystkich pierwiastków wielomianu. Korzysta ona z funkcji *bairstow.m* dla coraz to krótszych wielomianów, które są ilorazem z poprzedniego wywołania pętli, sprawdza z pomocą funkcji *uv2w.m* czy pierwiastki są rzeczywiste, a następnie dodaje obliczone pierwiastki do wektora *pierwiastki*. Pętla działa aż do momentu gdy przyjmowany wielomian jest przynajmniej kwadratowy. Gdy liczba współczynników będzie mniejsza niż 3 oznacza to, że nasz wielomian jest liniowy, więc nie ma sensu już korzystać z funkcji *bairstow.m*, ponieważ pierwiastek jest już bardzo prosty do obliczenia. Funkcja *zerowe.m* przyjmuje wektor *a* ze współczynnikami wielomianu, którego pierwiastki chcemy obliczyć, a zwraca wektor *pierwiastki* - przybliżone pierwiastki tego właśnie wielomianu.

## 3. *uv2w.m*

Funkcja ta przyjmuje wartości *u* i *v*, czyli współczynniki w iloczynie kwadratowym  $x^2 - ux - v$ , oblicza jego deltę i w zależności, czy jest ona dodatnia czy nie, oblicza wartości pierwiastków tego dwumianu. Przyjmuje wartości *u* i *v*, a zwraca wektor *w*, który ma dwie składowe jeśli pierwiastki są rzeczywiste, lub jedną składową, jeśli pierwiastki są zespolone (wystarczy jedna liczba zespolona, bo dalej i tak dodamy jej sprzężenie jako drugi pierwiastek). Funkcja zwraca dodatkową zmienną *real*, która jest wyznacznikiem dla funkcji *zerowe.m*, czy pierwiastki są zespolone czy rzeczywiste.

## 4. *timetest.m*

Skrypt służący do testowania szybkości metody Bairstowa w zależności od stopnia wielomianu.

## 5. *wielokrotnypierwiastek.m*

Funkcja przyjmuje parametry wstępnego przybliżenia *u* i *v*, znajduje miejsca zerowe dla wielomianów postaci  $(x - 1)^n$  dla  $n = \{1, 2, 3, 4, 5, 6, 7, 8\}$ . Zwraca tabelę wyników (0 w tabeli oznaczają brak wartości - w kolumnie dla  $n = 1$  na górze jedyny pierwiastek a reszta pozycji została oznaczona zerami) oraz tabelę z wartościami błędów bezwzględnych.

## 6. *Wielokrotnywykres.m*

Skrypt mający na celu zwizualizowanie błędów metody bairstowa dla wielomianów postaci  $(x - 1)^n$  dla  $n = \{1, 2, 3, 4, 5, 6, 7, 8\}$  dla różnych wartości *u* i *v*.

## 7. *Horner.m*

Funkcja wykorzystuje schemat hornera do obliczania wartości wielomianu w danym punkcie.

Działa również dla danych w postaci zwektoryzowanej. Funkcja horner jest używana do sprawdzania faktycznej wartości wielomianu w miejscach zerowych wyznaczonych metodą Bairstowa.

#### 8. *sprawdzenie.m*

Funkcja korzystająca z funkcji *Horner*, w celu obliczenia dokładnych wartości wielomianów w przybliżonych ich pierwiastkach. Korzystamy z niej w celu obliczania błędów przybliżeń. Funkcja przyjmuje wektory *wspolczynniki* - współczynniki wielomianu oraz *pierwiastki*, a zwraca wektor *y* - dokładne wartości wielomianu o podanych współczynnikach w dla podanych argumentów (pierwiastków).

#### 9. *czas.m*

Skrypt służy do analizy czasu wykonania metody Bairstowa dla wielomianu

$$x^4 - 4x^3 - 91x^2 + 34x + 1320$$

dla różnych wartości *u* i *v*. Wylicza średni czas wykonania metody dla 1000 prób.

### 3 Analiza błędów

Niestety wyznaczenie ogólniejszych błędów i innych zależności dla tej metody nie jest łatwe, ponieważ metoda bardzo różnie zachowuje się w różnych przypadkach. Dlatego zdecydowaliśmy się na analizę metody na wielu przykładach: wielomianów różnych stopni, różnej krotności pierwiastków, różnych rodzajach pierwiastków etc.

Chcieliśmy przeprowadzić analizę błędów w zależności od stopnia wielomianu. dlatego zbadaliśmy szczególny przypadek kolejnych potęg  $(x - 1)^n$  dla  $n = \{1, 2, 3, 4, 5, 6, 7, 8\}$ , aby wielomian był prosty i miał wielokrotne rzeczywiste pierwiastki.

Uwaga: błąd bezwzględny jest w naszym przypadku tym samym co błąd względny, ponieważ  $x = 1$ .

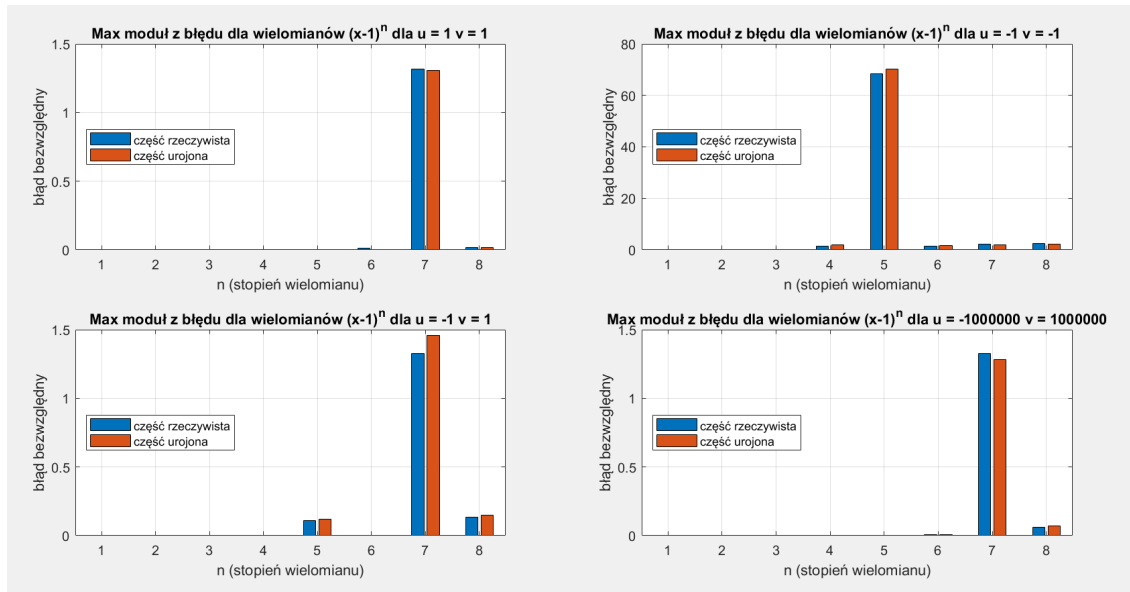


Figure 1: Analiza błędów dla różnych wartości  $u$  i  $v$

Możemy zauważyć, że metoda Bairstowa zachowuje się czasami bardzo dziwnie nawet dla najprostszych wielomianów. W tym przypadku dla większości kombinacji  $u$  i  $v$  odnotowujemy nienaturalne błędy dla wielomianu stopnia 7. Różnice pomiędzy maksymalnym błędem dla tego stopnia, a innych stopni jest od razu zauważalna szczególnie, że dla innych stopni błąd jest równy lub bardzo bliski zeru. Jednakże w prawym górnym wykresie większy błąd występuje dla wielomianu stopnia 5. Proszę przyjrzeć się osi Y przy tym wykresie - maksymalny błąd zarówno dla części rzeczywistej jak i zespolonej sięga 70, gdzie przy innych wykresach sięga on zaledwie 1.5.

## 4 Ciekawe przykłady obliczeniowe

### 1. $u, v = 0$

Rozpoczynając ciekawe przypadki powiemy o przypadkach, w których metoda Bairstowa nie działa do końca poprawnie. Pierwszym z nich jest ustawienie początkowych przybliżeń  $u$  i  $v$  na zera. Gdy wartości te są od początku ustawione na zera, może to prowadzić do wykonywania podstawowych operacji arytmetycznych (np. odejmowania) na liczbach bliskich zeru, co jak wiemy z wykładów, może prowadzić do znacznej utraty liczb znaczących, co może prowadzić z kolei do niedokładności obliczanych pierwiastków.

### 2. $Jakobian = 0$

Kolejny przypadek jest w jakimś stopniu powiązany z poprzednim. Chodzi tutaj o możliwość

napotkania się algorytmu na dzielenie przez zero. Może się to wydarzyć w momencie obliczania kolejnych wartości  $u$  i  $v$  w którejś iteracji pętli *while*. Dzielimy tam przez wartość Jakobianu, który jest obliczany ze wzoru:  $J = c_1c_3 - c_2^2$ . Jak nietrudno zauważyć wystarczy, że  $c_1c_3 = c_2^2$  i wartość Jakobianu wynosi 0. Realne szanse na to nie są zbyt duże, ale niestety nie mamy na to żadnego wpływu i może akurat wystąpić taka sytuacja. Przede wszystkim dzieje się tak, gdy  $c_1 = c_2 = c_3$ . Jako że  $b_k$  oraz  $c_k$  dla  $k \in \{1, n-2\}$  jest obliczane ze wzoru:

$$b_k = a_k + ub_{k+1} + vb_{k+2}$$

$$c_k = b_{k+1} + uc_{k+1} + vc_{k+2}$$

to widać, że gdy  $u$  i  $v$  są zerowe to  $c_k = b_{k+1} = a_{k+1}$ , więc dla  $a_3^2 = a_2a_4$  Jakobian się zeruje i dostajemy dzielenie przez 0.

### 3. Wielomian $(x-1)^5$ przy $u=1, v=-1$

Przy naszej analizie błędów nie znalazł się przypadek dla  $u=1, v=-1$ . Stało się tak dlatego, że metoda Bairstowa przy takich wartościach  $u$  i  $v$  nie radzi sobie z wielomianem  $(x-1)^5$ . Funkcja nawet po dłuższym działaniu nie zwraca żadnego wyniku i trzeba zatrzymać ją ręcznie, co jest ciekawe z uwagi na fakt, że dla większości testowanych przez nas wielomianów metoda Bairstowa działa wyjątkowo szybko, nawet przy  $u$  i  $v$  bardzo odległych od faktycznego miejsca zerowego.

```
In zerowe (line 11)
    [u_r,v_r,b1, b2, b_r] = bairstow(wspolczynniki,up,vp);

In wielokrotnypierwiastek (line 8)
p5 = zerowe([-1,5,-10,10,-5,1],u,v);
□
```

Figure 2: Komunikat w konsoli dla wielomianu  $(x-1)^5$  dla  $u=1, v=-1$

### 4. Metoda działająca dla pierwiastków rzeczywistych

Pomimo że z definicji metoda Bairstowa jest stworzona do szukania par pierwiastków zespolonych, radzi sobie ona również z przypadkami gdy wszystkie pierwiastki wielomianu są rzeczywiste. Przykład:  $(x-2)(x+2)(x+3)$

```
>> zerowe([-12,-4,3,1],1,1)

i =

      8

ans =

      2      -2      -3
```

Figure 3: Poprawny wynik dla wielomianu  $w(x) = (x - 2)(x + 2)(x + 3)$

#### 5. Dwa pierwiastki „niezależne” od $u$ i $v$ , pozostałe bardzo zależne

W naszych obserwacjach często pojawiały się przypadki, w których pewna para pierwiastków była wyznaczana bardzo dokładnie bez względu na wybór punktów początkowych, a pozostałe wyznaczone były o wiele mniej dokładnie a błąd bardzo zmieniał się przy doborze  $u$  i  $v$ . Przykład (wielomian o pierwiastkach rzeczywistych dla łatwiejszej obserwacji):

$$(x + 6)(x - 4)(x + 5)(x - 11) \quad -> \quad \text{pierwiastki} = -6, 4, -5, 11$$

uv	wyniki	iteracje	błędy
"u = v = 1"	" 4.0000e+00 -5.0000e+00 1.1894e+01 -6.8941e+00"	11	"0 0 0.894 -0.8941"
"u = v = -1"	"4.0000e+00 -5.0000e+00 1.2904e+01 -7.9043e+00"	11	" 0 0 1.2904 -1.9043"
"-u = v = 1"	"4.0000e+00 -5.0000e+00 1.2904e+01 -7.9043e+00"	10	" 0 0 1.2904 -1.9043"
"u = -v = 1"	"4.0000e+00 -5.0000e+00 1.1894e+01 -6.8941e+00"	13	" 0 0 0.894 -0.8941"
"u = v = 100"	" 1.1000e+01 -6.0000e+00 4.0000e+00 -5.0000e+00"	23	" 0 0 0 0"
"u = v = 1000000"	"1.1000e+01 -6.0000e+00 4.0000e+00 -5.0000e+00"	57	" 0 0 0 0"

Figure 4: Tabela wyników, liczby iteracji i błędów bezwzględnych dla powyższego wielomianu

#### 6. Szybkość metody

Ciekawy dla nas był sam przebieg metody i jej wydajność. Na histogramie przedstawiona została częstość liczby iteracji dla wielomianu

$$4x^3 + 3x^2 + 2x + 1$$

dla  $u = v = [1:1000]$ . Należy zauważyć że nawet dla bardzo dużych a tym samym odległych od faktycznych pierwiastków  $u$  i  $v$  metoda nie wykonuje więcej niż 100 iteracji, a najczęściej wykonuje

ich od 20 do 40. Można by pomyśleć że im bliższe pierwiastkom początkowe przybliżenia, tym metoda jest bardziej wydajna, ale na kilku sprawdzonych przez nas przykładach okazuje się że często to bardziej odległe  $u$  i  $v$  (a zazwyczaj co za tym idzie większa liczba iteracji) daje dokładniejszy wynik. Pozostaje sprawdzenie czy zabiera to więcej czasu.

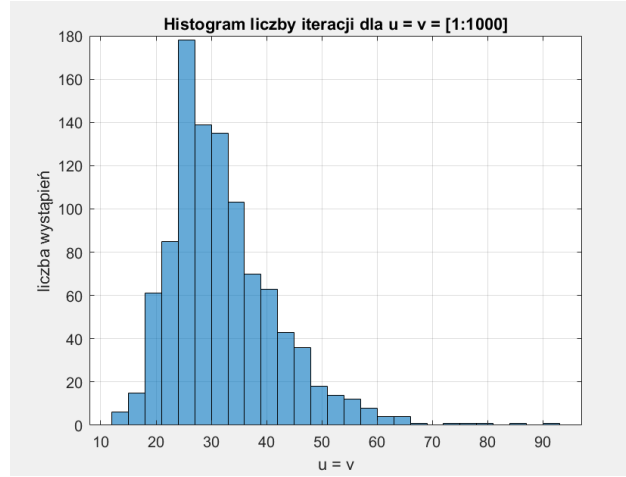


Figure 5: Analiza liczby iteracji dla wielomianu o współczynnikach [1,2,3,4]

<b>uv</b>	<b>czasy</b>
"u = v = 1"	"2.7015e-04"
"u = v = -1"	"3.0507e-04"
"-u = v = 1"	"1.5582e-04"
"u = -v = 1"	"1.5582e-04"
"u = v = 100"	"2.3484e-04"
"u = v = 1000000"	"2.0513e-04"

Figure 6: Analiza sredniego czasu dla 1000 prob dla wielomianu o współczynnikach [1320, 34, -91, -4,1]

W celu analizy zmiany czasu działania programu w zależności od doboru początkowych  $u$  i  $v$  przeprowadziliśmy eksperyment dla wielomianu:

$$x^4 - 4x^3 - 91x^2 + 34x + 1320$$

Dla wartości  $u$  i  $v$  takich jak w tabeli przeprowadziliśmy 1000 prób wyznaczenia wszystkich miejsc zerowych wielomianu oraz policzyliśmy ich średni czas. Okazuje się, że nie dość że metoda Bairstowa



ogólnie jest bardzo szybka to czas działania w stosunku do początkowych  $u$  i  $v$  jest prawie niezmienny. Dodatkowo nawet po wykonaniu tak wielu prób wynik średniego czasu działania dla tych samych  $u$  i  $v$  różnił się w kilku próbach.

## 5 Końcowa analiza uzyskanych wyników

Z naszej analizy wynikło kilka kluczowych wniosków na temat metody Bairstowa.

Zalety metody:

- bardzo dobra szybkość bez względu na dobór punktów początkowych
- możliwość wyznaczania zarówno pierwiastków rzeczywistych jak i zespolonych
- bardzo dokładne wyznaczanie pierwiastków

Wady metody:

- dla niektórych przypadków nie zwraca wartości, ponieważ program zawiesza się
- w niektórych momentach jacobian może się zerować i metoda jako pierwiastek zwraca NaN
- istnieją takie doборы punktów początkowych, które nie zwracają wyniku bądź generują duże błędy
- problemy z lokalizacją powyżej opisanych przypadków - oprócz zerowania się jacobianu bardzo ciężko przewidzieć w jakim eksperymencie metoda może nie zadziałać
- metoda nie do końca radzi sobie z przypadkami wielomianów o nieparzystym stopniu - takich gdzie liczba pierwiastków jest nieparzysta. Gdy po kilkutrotnej iteracji i "usuwaniu" kolejnych czynników kwadratowych pozostaje równanie liniowe, ostatni pierwiastek trzeba policzyć ręcznie, poza metodą.

źródła: "Analiza numeryczna", David Kincaid, Ward Cheney, 2006