Obliczenia naukowe
Lista 3

Szymon Janiak

November 19, 2023

Zadanie 1

Opis problemu

Napisać funkcję rozwiązującą równanie $f(x) = 0$ metodą bisekcji.

Dane wejściowe

- `f` — funkcja f w postaci anonimowej funkcji
- `a,b` — liczby typu `Float64` określające końce przedziału początkowego
- `delta, epsilon` — liczby typu `Float64` określające dokładności obliczeń

Czwórka wartości `(r, val, it, err)`.

- `r` — przybliżenie pierwiastka równania $f(x) = 0$
- `val` — wartość funkcji w r
- `it` — liczba wykonanych iteracji
- `err` — sygnalizacja błędu, możliwe wartości:
  - `0` — brak błędu
  - `1` — funkcja nie zmienia znaku w przedziale $[a; b]$

Opis algorytmu

Metoda bisekcji polega na stopniowym zawężaniu przedziału szukania naszego pierwiastka do momentu, gdy nasz wynik będzie wystarczająco bliski, co zależne jest od zdefiniowanego przez nas ϵ. Do zastosowania tej metody potrzebne są spełnione dwa założenia:

- funkcja $f(x)$ jest ciągła w przedziale domkniętym $[a, b]$,
- funkcja przyjmuje różne znaki na końcach przedziału.

Kolejne iteracje algorytmu przesuwają jedną z granic przedziału bliżej pierwiastka o połowę długości przedziału. Wybór końca do przesunięcia polega na sprawdzeniu które przesunięcie zwróci nam przedział, który spełnia warunek zmiany znaku.

Rozwiązanie

Algorithm 1 bisection method

$val \leftarrow 0$
$it \leftarrow 0$
$e \leftarrow b - a$
$u \leftarrow f(a)$
$v \leftarrow f(b)$
$r \leftarrow \frac{1}{2} * (a + b)$
if $sign(u) = sign(v)$ **then**
    $err \leftarrow 1$
    return r, val, it, err
end if
while $|e| > \delta$ and $|f(r)| > \epsilon$ **do**
    $e \leftarrow frace2$
    $r \leftarrow a + e$
    $val \leftarrow f(r)$
    $it \leftarrow it + 1$
    if $sign(val) \neq sign(u)$ **then**
        $g \leftarrow r$
        $v \leftarrow val$
    else
        $a \leftarrow r$
        $u \leftarrow val$
    end if
end while
return r, val, it, err

Zadanie 2

Opis problemu

Napisać funkcję rozwiązującą równanie $f(x) = 0$ metodą Newtona.

Dane wejściowe

- `f` — funkcja f w postaci anonimowej funkcji
- `pf` — pochodna funkcji f w postaci anonimowej funkcji
- `x0` — przybliżenie początkowe
- `delta, epsilon` — liczby typu `Float64` określające dokładności obliczeń
- `maxit` — liczba całkowita określająca dopuszczalną liczbę iteracji

Dane wyjściowe

Czwórka wartości `(r, val, it, err)`.

- `r` — przybliżenie pierwiastka równania $f(x) = 0$
- `val` — wartość funkcji w r
- `it` — liczba wykonanych iteracji
- `err` — sygnalizacja błędu, możliwe wartości:
  - `0` — metoda zbieżna
  - `1` — nie osiągnięto wymaganej dokładności w `maxit` iteracji
  - `2` — pochodna bliska zeru

Opis algorytmu

W tym algorytmie potrzebujemy założyć, że w naszym przedziale $[a, b]$ znajduje sie dokładnie jeden pierwiastek funkcji f, oraz, że funkcja przyjmu wartości różnych znaków na krańcach przedziału. Dodatkowym wymaganiem jest stały znak pierwszej i drugiej pochodnej funkcji w tym przedziale. Na początku przyjmujemy sobie za x_1 granicę a lub b i wyznaczamy równanie stycznej do wykresu funkcji w punkcie $[x_1, f(x_1)]$, następnie wyznaczamy odciętą x_2 punktu przecięcia tej stycznej z osią OX - w ten sposób otrzymujemy kolejne przybliżenie naszego rozwiązania.Całą procedurę powtarzamy do momentu gdy otrzymamy wynik mieszczący się w naszym ϵ tworząc kolejne styczne.

Rozwiązanie

Algorithm 2 Newton method

$val \leftarrow f(x_0)$
$val\_prime \leftarrow 0$
$x_1 \leftarrow 0$
$it \leftarrow 1$
if $abs(v) < \epsilon$ **then**
  $err \leftarrow 0$
  return x_0, val, it, err
end if
for it to $maxit$ **do**
  $val\_prime \leftarrow pf(x_0)$
  $x_1 = x_0 - \frac{val}{val\_prime}$
  $val \leftarrow f(x_1)$
  if $|val\_prime| < \epsilon$ **then**
    $err \leftarrow 2$
    return $x_0, f(x_0), it, err$
  end if
  if $|x_1 - x_0| < \delta$ or $|val| < \epsilon$ **then**
    return return x_1, val, it, err
  end if
  $x_0 \leftarrow x_1$
end for
$err \leftarrow 1$
return x_0, val, it, err

Zadanie 3

Opis problemu

Napisać funkcję rozwiązującą równanie $f(x) = 0$ metodą siecznych.

Dane wejściowe

- `f` — funkcja f w postaci anonimowej funkcji
- `x0,x1` — przybliżenia początkowe
- `delta, epsilon` — liczby typu `Float64` określające dokładności obliczeń
- `maxit` — liczba całkowita określająca dopuszczalną liczbę iteracji

Dane wyjściowe

Czwórka wartości `(r,v,it,err)`.

- `r` — przybliżenie pierwiastka równania $f(x) = 0$
- `v` — wartość funkcji w r
- `it` — liczba wykonanych iteracji
- `err` — sygnalizacja błędu, możliwe wartości:
  - `0` — metoda zbieżna
  - `1` — nie osiągnięto wymaganej dokładności w `maxit` iteracji

Opis algorytmu

Do użycia tej metody potrzebujemy dwa punkty startowe x_0 i x_1, których będziemy używać do wyznaczania następnych przybliżeń. Obliczamy $f(x_0)$ i $f(x_1)$, a następnie poprowadzamy przez te dwa punkty sieczną. Przecięcie siecznej z osią OX wyznaczy nam punkt do następnej iteracji. W ten sposób będziemy wyznaczać coraz to bliższe przybliżenia funkcji używając do tego zawsze dwóch wartości x_n oraz x_{n+1}. Metoda ta nie zawsze jest zbieżna.

Rozwiązanie

Algorithm 3 secant method

$it \leftarrow 0$
$a \leftarrow x_0$
$b \leftarrow x_1$
$val \leftarrow f(x_0)$
$val\_next \leftarrow f(x_1)$
for it to $maxit$ **do**
    if $|val| > |val\_next|$ **then**
        $a, b = b, a$
        $val, val\_next \leftarrow val\_next, val$
    end if
    $d \leftarrow \frac{b-a}{val\_next - val}$
    $b \leftarrow a$
    $val\_next \leftarrow val$
    $a \leftarrow a - d * val$
    $val \leftarrow f(a)$
    if $|val| < \epsilon$ or $|b - a| < delta$ **then**
        return a, val, it, err
    end if
end for
$err \leftarrow 1$
return a, val, it, err

Zadanie 4

Opis problemu

Należy obliczyć pierwiastek równania $\sin x - \left(\frac{1}{2}x\right)^2 = 0$ przy pomocy wcześniej zaprogramowanych metod:

- metoda bisekcji z przedziałem początkowym $[1.5, 2]$
- metoda Newtona z przybliżeniem początkowym $x_0 = 1.5$
- metoda siecznych z przybliżeniami początkowymi $x_0 = 1$, $x_1 = 2.0$

i precyzją $\delta = \frac{1}{2}10^{-5}$, $\epsilon = \frac{1}{2}10^{-5}$.

Wyniki

Metoda bisekcji

- pierwiastek, $r = 1.9337539672851562$
- wartość funkcji, $f(r) = -2.7027680138402843 \cdot 10^{-7}$
- liczba iteracji, $it = 16$
- brak błędu, $\text{err} = 0$

Metoda Newtona

- pierwiastek, $r = 1.933753779789742$
- wartość funkcji, $f(r) = -2.2423316314856834 \cdot 10^{-8}$
- liczba iteracji, $it = 4$
- brak błędu, $\text{err} = 0$

Metoda siecznych

- pierwiastek, $r = 1.933753644474301$
- wartość funkcji, $f(r) = 1.564525129449379 \cdot 10^{-7}$
- liczba iteracji, $it = 3$
- brak błędu, $\text{err} = 0$

Wnioski

Wszystkie metody zwróciły nam bardzo podobne rezultaty jeśli chodzi o wartość naszego przybliżonego pierwiastka, który w rzeczywistości wynosi 2. Możemy natomiast zauważyć różnice w ilości przeprowadzonych iteracji. Dla metody Newtona oraz siecznych jest to kolejno 4 i 3 co jest bardzo dobrym wynikiem. Odstaje tutaj metoda bisekcji, która musiała wykonać aż 16 iteracji, gdyż jest metodą globalną i powoli, liniowo zbliża się do pierwiastka. Zaletą jej jednak jest to, że do jej użycia potrzebujemy mniej informacji początkowych.

Zadanie 5

Opis problemu

Należy wyznaczyć punkt, w którym przecinają się funkcje $y = 3x$ oraz $y = e^x$. Dokładność obliczeń wynosi: $\delta = 10^{-4}$, $\epsilon = 10^{-4}$. Należy użyć metody bisekcji.

Wyniki

Pierwszy punkt

- pierwiastek, $r = 0.619140625$
- wartość funkcji, $f(r) = 9.066320343276146 \cdot 10^{-5}$
- liczba iteracji, $it = 9$
- brak błędu, $\mathrm{err} = 0$

Drugi punkt

- pierwiastek, $r = 1.5120849609375$
- wartość funkcji, $f(r) = 7.618578602741621 \cdot 10^{-5}$
- liczba iteracji, $it = 13$
- brak błędu, $\mathrm{err} = 0$

Wnioski

Wyniki pokrywają się z rzeczywistością.

Zadanie 6

Opis problemu

Należy znaleźć miejsca zerowe funkcji:

- $f_1(x) = e^{1-x} - 1$
- $f_2(x) = xe^{-x}$

za pomocą metod bisekcji, Newtona i siecznych. Wymagane dokładności obliczeń: $\delta = 10^{-5}$, $\epsilon = 10^{-5}$. Sprawdzić co się stanie, gdy w metodzie Newtona dla f_1 wybierzemy $x_0 \in (1; \infty]$, a dla f_2 wybierzemy $x_0 > 1$, czy możemy wybrać $x_0 = 1$ dla f_2?

Wyniki dla f_1

Metoda bisekcji

Dla $a = 0.4$ oraz $b = 1.4$

- pierwiastek, $r = 1.000006103515625$
- wartość funkcji, $f(r) = -6.103496998477453 \cdot 10^{-6}$
- liczba iteracji, $it = 15$
- brak błędu, err $= 0$

Metoda Newtona

Dla $x_0 = 0.8$

- pierwiastek, $r = 0.9999999848053367$
- wartość funkcji, $f(r) = 1.5194663527395846 \cdot 10^{-8}$
- liczba iteracji, $it = 3$
- brak błędu, err $= 0$

Metoda siecznych

Dla $x_0 = 0.2$ oraz $x_1 = 0.6$

- pierwiastek, $r = 0.9999999855368947$
- wartość funkcji, $f(r) = 1.4463105380002617 \cdot 10^{-6}$
- liczba iteracji, $it = 4$
- brak błędu, err $= 0$

Wyniki dla f_2

Metoda bisekcji

Dla $a = -1.5$ oraz $b = 1.0$

- pierwiastek, $r = -3.814697265625 \cdot 10^{-6}$
- wartość funkcji, $f(r) = -3.814711817567984 \cdot 10^{-6}$
- liczba iteracji, $it = 17$
- brak błędu, err $= 0$

Metoda Newtona

Dla $x_0 = -0.4$

- pierwiastek, $r = -1.8440313309425922 \cdot 10^{-8}$
- wartość funkcji, $f(r) = -1.844031364947108 \cdot 10^{-8}$
- liczba iteracji, $it = 4$
- brak bledu, err $= 0$

Metoda siecznych

Dla $x_0 = -0.4$ oraz $x_1 = -0.2$

- pierwiastek, $r = -6.922503966219477 \cdot 10^{-6}$
- wartość funkcji, $f(r) = -6.922551887446507 \cdot 10^{-6}$
- liczba iteracji, $it = 3$
- brak błędu, err = 0

Wnioski dla powyższych wyników

Na pewno możemy zauważyć podobna sytuację co w zadaniu 4. Metoda bisekcji musi wykonać znacznie więcej iteracji algorytmu aby znaleźć zadowalający wynik. Wszystkie algorytmy zwróciły zadowalające wyniki.

Wyniki metody Newtona dla f_1 oraz f_2 przy dodatkowych założeniach

f_1 **przy** $x_0 \in (1, \infty)$

$x_0 = 15.0$

- pierwiastek, $r = 15.0 \cdot 10^{6}$
- wartość funkcji, $f(r) = -0.9999991684712809$
- liczba iteracji, $it = 1$
- pochodna bliska zeru, err = 2

Widzimy, że dostajemy błąd - pochodna bliska zeru.

f_2 **przy** $x_0 > 1$

$x_0 = 7.0$

- pierwiastek, $r = 14.792276940955892$
- wartość funkcji, $f(r) = 5.569686859646652 \cdot 10^{-6}$
- liczba iteracji, $it = 7$
- brak błedu, err = 0

Można zauważyć, że pomimo braku błędu zwróconego przez program nasz wynik jest niepoprawny. Zarówno dla f_1 i f_2 przy $x_0 > 1$ występują problemy przy obliczaniu bardzo małych wartości e^{-x} ze względu na ograniczenia arytmetyki.

f_2 **przy** $x_0 = 1.0$

- pierwiastek, $r = 1.0$
- wartość funkcji, $f(r) = 0.36787944117144233$
- liczba iteracji, $it = 1$
- pochodna bliska zeru, err = 2

Przy $x_0 = 1.0$ dostajemy błędny wynik, gdyż $f_2(1) = 0$ przez co otrzymana styczna będzie równoległa do osi OX, co przeszkodzi algorytmowi w wyznaczeniu następnego punktu.