Obliczenia naukowe

Sprawozdanie nr 3

Barbara Banaszak
236514

22-11-2018

1 Zadanie 1

1.1 Opis problemu

W zadaniu należy zaimplementować algorytm bisekcji, służący do znajdowania przybliżonego miejsca zerowego zadanej funkcji. Metoda bisekcji korzysta z własności Darboux dla funkcji ciągłych, tzn jeśli $f(a)f(b) < 0$ to funkcja zmienia znak w przedziale $[a, b]$, czyli ma miejsce zerowe w przedziale (a, b). Korzystając z tej własności, jeśli $f(a)f(b) < 0$ to obliczamy $c = \frac{1}{2}(a + b)$ i sprawdzamy czy $f(a)f(c) < 0$, jeśli tak, to funkcja ma miejsce zerowe w $[a, c]$, a w przeciwnym wypadku $f(b)f(c) < 0$ i funkcja ma miejsce zerowe w $[c, b]$.

1.2 Rozwiązanie

INPUT: f (zadana funkcja), a, b (końce przedziału początkowego), $delta, epsilon$ (dokładność obliczeń)
OUTPUT: c (przybliżenie pierwiastka), w (wartość f(c)), k (ilość iteracji), err (status błędu 0-brak błędu; 1-funkcja nie zmienia znaku w przedziale)

$u \leftarrow f(a)$
$v \leftarrow f(b)$
$e \leftarrow b - a$
if $sgn(u) = sgn(v)$ **then**
  return $err = 1$
end if
for $k = 1$ **to** M **do**
  $e \leftarrow \frac{e}{2}$
  $e \leftarrow a + e$
  $w \leftarrow f(c)$
  if $|e| < delta$ **or** $|w| < epsilon$ **then**
    return $c, w, k, err = 0$
  end if
  if $sgn(u) \neq sgn(v)$ **then**
    $b \leftarrow e$
    $v \leftarrow w$
  else
    $a \leftarrow c$
    $u \leftarrow w$
  end if
end for
return $c, w, k = M, err = 0$

W zaimplementowanym algorytmie bisekcji znajdzie się także kilka elementów, które warto wyjaśnić. Po pierwsze punkt środkowy c jest obliczany przy pomocy instrukcji $c \leftarrow a + (b - a)/2$, a nie $c \leftarrow (a + b)/2$, ponieważ zawsze lepiej jest obliczać nową wielkość poprzez dodanie do poprzedniej małej poprawki. Po drugie badamy zmianę znaku wartości funkcji za pomocą nierówności $sgn(u) \neq sgn(v)$, dzięki czemu nie wykonujemy zbędnego mnożenia $u * v$, które także mogłoby powodować błędy. W algorytmie znajdziemy także zmienną M, która określa maksymalną dopuszczoną liczbę kroków i jest koniecznym zabezpieczeniem pozwalającym usunąć ryzyko niekończących się obliczeń.

2 Zadanie 2

2.1 Opis problemu

W zadaniu należy zaimplementować algorytm stycznych inaczej Newtona,służący do znajdowania przybliżonego miejsca zerowego zadanej funkcji. Metoda stycznych opiera się na linearyzacji funkcji, czyli zastąpieniu funkcji funkcją liniową, jak sama nazwa mówi, będącą styczną do wykresu funkcji w danym punkcie x_n. Zbieżność metody Newtona jest kwadratowa, zatem gdy przybliżenia tworzone tą metodą są dostatecznie bliskie rzeczywistego pierwiastka, w kilka dodatkowych przybliżeń łatwo jest otrzymać maksymalną dokładność. Należy jednak pamiętać o tym, że metoda Newtona nie zawsze jest zbieżna, z tego powodu często używa się jej w kombinacji z inną wolniejszą metodą, np. bisekcji. Jej wadą jest także to, że za każdym razem musimy obliczać pochodna funkcji, której zera szukamy.

2.2 Rozwiązanie

INPUT: f (zadana funkcja),pf (pochodna funkcji) x_0 (punkt początkowy), $delta, epsilon$ (dokładność obliczeń), M (maksymalna ilość iteracji)
OUTPUT: c (przybliżenie pierwiastka), v (wartość f(c)), k (ilość iteracji), err (status błędu 0-metoda zbieżna; 1-nie osiągnięto wymaganej dokładności; 2-pochodna bliska 0)

$v \leftarrow f(x_0)$
if $|v| < epsilon$ **then**
  return $c, v, 0, 0$
end if
for $k = 1$ **to** M **do**
  if $|pf(x_0)| < epsilon$ **then**
    return $c, v, k, err = 2$
  end if
  $x_1 \leftarrow x_0 - \frac{v}{fp(x0)}$
  $x \leftarrow f(x_1)$
  if $|x_1 - x_0| < delta$ **or** $|v| < epsilon$ **then**
    return $c, w, k, err = 0$
  end if
  $x_0 \leftarrow x_1$
end for
return $c, w, k = M, err = 1$

Zaimplementowana metoda Newtona opiera się o wzór $x_{n+1} := x_n - \frac{f(x_n)}{f'(x_n)}$ dla $n \geqslant 0$ z definicji metody Newtona. W algorytmie konieczne jest także sprawdzanie wartości pochodnej pf, jeśli jest ona dostaczenie bliska zera, program zakończy swoje oblicznia i zwróci błąd, niemożliwe jest wtedy zastosowanie metody Newtona.

3 Zadanie 3

3.1 Opis problemu

W zadaniu należy zaimplementować algorytm siecznych, służący do znajdowania przybliżonego miejsca zerowego zadanej funkcji. Algorytm ten jest zmodyfikowaną metodą Newtona, gdzie we wzorze $x_{n+1} := x_n - \frac{f(x_n)}{f'(x_n)}$ pochodną zastępujemy ilorazem różnicowym $f'(x_n) \approx \frac{f(x_n)-f(n_{n-1})}{x_n - x_{n-1}}$ i otrzymujemy $x_{n+1} := x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n)-f'(x_{n-1})}$. Metoda siecznych podobnie jak metoda Newtona opiera się o linearyzację funkcji, w metodzie siecznych przybliżamy funkcję sieczną. Warto także zauważyć, że x_{n+1} wyraża się przez x_n i x_{n-1}, więc potrzebne są dwa punkty początkowe.

3.2 Rozwiązanie

INPUT: f (zadana funkcja),x_0, x_1 (przybliżenia początkowe), $delta, epsilon$ (dokładność obliczeń), M (maksymalna ilość iteracji)
OUTPUT: x_0 (przybliżenie pierwiastka), f_a (wartość f(c)), k (ilość iteracji), err (status błędu 0-metoda zbieżna; 1-nie osiągnięto wymaganej dokładności)

$f_a \leftarrow f(a)$
$f_b \leftarrow f(b)$
for $k = 2$ **to** M **do**
  if $|f_a| > |f_b|$ **then**
    $a \leftrightarrow b$
    $f_a \leftrightarrow f_b$
  end if
  $x_1 \leftarrow \frac{b-a}{f_b - f_a}$
  $b \leftarrow a$
  $f_b \leftarrow f_a$
  $a \leftarrow a - f_a * s$
  $f_a \leftarrow f(a)$
  if $|f_a| < epsilon$ **or** $|b - a| < delta$ **then**
    return $x_0, f_a, k, err = 0$
  end if
end for
return $x_0, f_a, k, err = 1$

Warto zauważyć, że zaimplementowany algorytm przestawia końce x_0 i x_1 przedziału (symbol $\leftrightarrow$), gdy wymaga tego utrzymanie nierówności $|f_a| \leqslant |f_b|$.

Dzięki temu, od drugiego kroku moduły wartości funkcji w punktach x_n nie rosną.

4 Zadanie 4

4.1 Opis problemu

Zadanie polega na znalezieniu miejsca zerowego równania $sin(x) - (\frac{1}{2}x)^2 = 0$. Z wykorzystaniem metod:

1. bisekcji z przedziałem początkowym $[1.5, 2]$, $\delta = \frac{1}{2}10^{-5}$, $\epsilon = \frac{1}{2}10^{-5}$
2. Newtona z przybliżeniem początkowym $x_0 = 1.5$, $\delta = \frac{1}{2}10^{-5}$, $\epsilon = \frac{1}{2}10^{-5}$
3. siecznych z przybliżeniem początkowym $x_0 = 1$, $x_1 = 2$, $\delta = \frac{1}{2}10^{-5}$, $\epsilon = \frac{1}{2}10^{-5}$

4.2 Rozwiązanie

W celu znalezienia pierwiastków równania podanymi metodami, wykorzystamy wcześniej zaimplementowane funkcje, znajdujące się w module **RootFindMethod**.

4.3 Wyniki

Metoda	Miejsce zerowe	Wartość funkcji	Iteracje	Błąd
Bisekcji	1.9337539672851562	-2.7027680138402843e-7	16	0
Newtona	1.933753779789742	-2.2423316314856834e-8	4	0
Siecznych	1.9337539405015145	-2.3487103129049558e-7	5	0

4.4 Wnioski

Wszystkie 3 metody poradziły sobie ze znalezieniem miejsca zerowego zadanej funkcji, przedziały i przybliżenia początkowe zostały dobrane dobrze. Patrząc na otrzymane wyniki łatwo także zauważyć znaczącą różnicę w liczbie iteracji wykonanych przez metodę bisekcji i metody Newtona oraz siecznych. Metoda bisekcji jest zdecydowanie dużo wolniejszą metodą. Warto jednak pamiętać o tym, że metody Newtona i siecznych będą się sprawdzać w momencie, gdy uda nam się dobrze dobrać punkt początkowy.

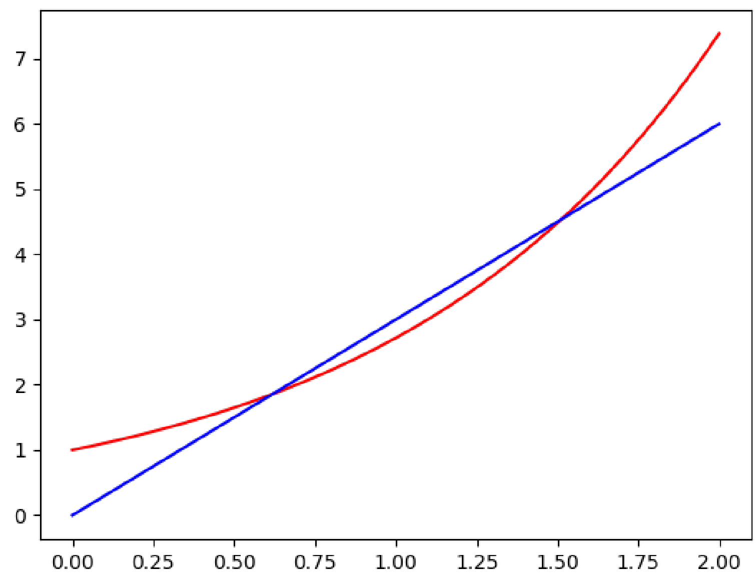
5 Zadanie 5

5.1 Opis problemu

Zadanie polega na znalezieniu punktów przecięcia funkcji $y = 3x$ oraz $y = e^x$ metodą bisekcji, z dokładnością $\delta = 10^{-4}$, $\epsilon = 10^{-4}$.

5.2 Rozwiązanie

W celu znalezienia rozwiązanie przyrównujemy dane funkcje do siebie $3x = e^x$ i otrzymujemy równanie $e^x - 3x = 0$. Szukamy miejsc zerowych otrzymanego równania używając metody bisekcji zaimplementowanej w module **RootFindMethod**. Przedziały dobieramy sugerując się poniższym wykresem przedstawiającym obie funkcje. Będą to [0,1] oraz [1,2].



5.3 Wyniki

Przedział	Miejsce zerowe	Wartość funkcji	Iteracje	Błąd
$[0, 1]$	0.619140625	-9.066320343276146e-5	9	0
$[1, 2]$	1.5120849609375	-7.618578602741621e-5	13	0

5.4 Wnioski

Główną trudnością zadania było znalezienie odpowiednich, poprawnych przedziałów początkowych. Bez narysowania wykresów obu funkcji mogłoby być to nieco kłopotliwe, gdyż punkty przecięcia funkcji leżą dość blisko siebie i już wzięcie przedziału $[0, 2]$ sprawi, że metoda bisekcji nie zadziała, ponieważ znak funkcji na krańcach przedziału się nie zmienia.

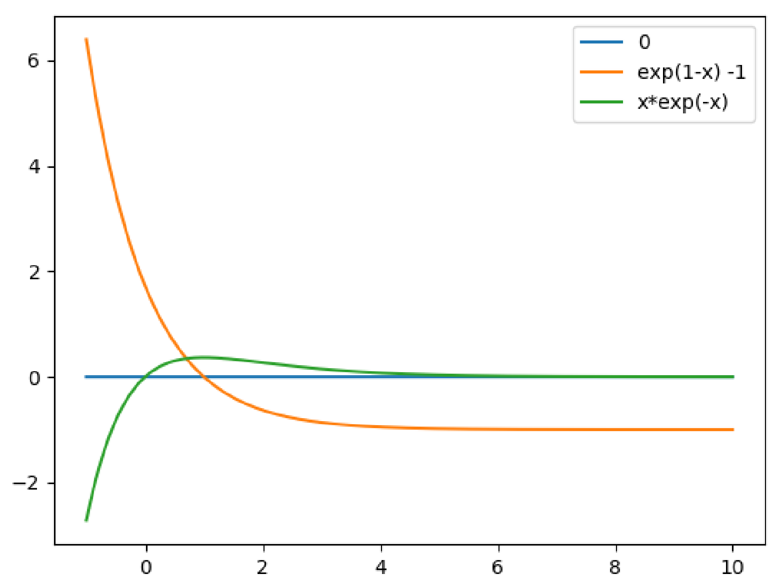
6 Zadanie 6

6.1 Opis problemu

Zadanie polega na znalezieniu miejsc zerowych funkcji $f_1 = e^{1-x} - 1$ oraz $f_2 = xe^{-x}$ za pomocą metody bisekcji, Newtona i siecznych, z dokładnością $\delta = 10^{-5}$, $\epsilon = 10^{-5}$. Należy także sprawdzić co się stanie, gdy w metodzie Newtona dla f_1 wybierzemy $x_0 \in (1, \infty]$, dla f_2 wybierzemy $x_0 > 0$ oraz czy można wybrać $x_0 = 1$ dla f_2.

6.2 Rozwiązanie

W celu rozwiązania zadania zaimplementowane podane funkcje oraz ich pochodne, aby łatwiej było dobierać przedziały i przybliżenia początkowe danych funkcji, przebieg funkcji został zilustrowany wykresem. Do wyznaczenia zer funkcji wymaganymi metodami zostaną użyte funkcje zaimplementowane w module **RootFindMethod**.



6.3 Wyniki

Metoda bisekcji f_1			
Przedział	Miejsce zerowe	Wartość funkcji	Iteracje
$[0, 3]$	1.0000076293945312	-7.6293654275305656e-6	17
$[-10, 10]$	1.0000038146972656	-3.814689989667386e-6	20
$[1, 2]$	1.0	0.0	1
Metoda bisekcji f_2			
Przedział	Miejsce zerowe	Wartość funkcji	Iteracje
$[-1, 1]$	0.0	1.0	1
$[0, 100]$	50.0	9.643749239819589e-21	1
$[-2, 3]$	7.62939453125e-6	7.62933632381113e-6	17

Metoda Newtona f_1			
Początek	Miejsce zerowe	Wartość funkcji	Iteracje
-1	0.9999922654776594	7.734552252003368e-6	5
0	0.9999984358892101	1.5641120130194253e-6	4
1	1.0	0.0	0
2	0.9999999810061002	1.8993900008368314e-8	5
8	Błąd 1	Błąd 1	Błąd 1
20	Bład 2	Bład 2	Bład 2
Metoda Newtona f_2			
Początek	Miejsce zerowe	Wartość funkcji	Iteracje
-2	-1.425500682806244e-9	-1.425500684838296e-9	7
0	0.0	0.0	0
0.5	-3.0642493416461764e-7	-3.0642502806087233e-7	5
1	Bład 2	Bład 2	Bład 2
4	14.398662765680003	8.036415344217211e-6	9
100	100.0	3.7200759760208363e-42	0

Metoda siecznych f_1			
Początek	Miejsce zerowe	Wartość funkcji	Iteracje
$-1, 2$	1.0000009310146594	-9.310142259355558e-7	7
$0, 1.5$	1.000002881921338	-2.8819171853378123e-6	5
$10, 100$	Błąd 1	Błąd 1	Błąd 1
Metoda siecznych f_2			
Początek	Miejsce zerowe	Wartość funkcji	Iteracje
$-1, 0.5$	3.201418966654486e-7	3.2014179417463104e-7	7
$-1, 2$	14.448586456655065	7.671565396829614e-6	16
$-5, 1000$	1000.0000000000001	0.0	1

6.4 Wnioski

Dla metody bisekcji możemy zauważyć, że nie ma znaczenia jak duży przedział wybierzemy. Metoda ta jest globalnie zbieżna, zatem zawsze dostaniemy mniej lub bardziej dokładny, poprawny wynik. Trochę inna sytuacja jest w przypadku funkcji f_2, która zbiega w nieskończoności do zera, co powoduje, że gdy weźmiemy wystarczająco duży przedział, obliczenia zakończą się po jednej iteracji, gdyż wartość funkcji będzie dostatecznie bliska zera.

W przypadku metody Newtona im punkt początkowy jest dalszy od rzeczywistego pierwiastka tym więcej iteracji jest potrzebnych, żeby go osiągnąć. Dla wartości $x_0 > 8$ zaczynamy otrzymywać błędy metody. Najpierw dana wartość iteracji okazuje się niewystarczająca($blad1$ - metoda rozbieżna). Potem natomiast funkcja f_1 zaczyna maleć bardzo powoli, a co za tym idzie jej pochodna jest bliska 0, co uniemożliwia korzystanie z metody Newtona ($blad2$). W przypadku funkcji f_2 dla $x_0 = 1$ otrzymujemy $blad2$, co jest spowodowane tym, że w w tym punkcie pochodna f_2 jest równa 0. Natomiast dla każdej wartości $x_0 > 1$, podobnie jak w przypadku metody bisekcji, metoda Newtona nie zbiega do faktycznego zera, lecz znajduje wystarczająco małą wartość daleką od miejsca zerowego.

Metoda siecznych zachowuje się podobnie do metody Newtona, dla f_1 dla dużych wartości początkowych dana wartość iteracji okazuje się niewystarczająca. Dla funkcji f_2, podobnie jak w przypadku dwóch poprzednich metod, metoda siecznych także nie zbiega do rzeczywistego miejsca zerowego funkcji.