

Obliczenia Naukowe

Lista 4

Laboratoria

Piotr Szyma

7 grudnia 2017

1 Zadanie 1

1.1 Opis problemu

Napisać funkcję obliczającą ilorazy różnicowe zgodnie ze specyfikacją podaną w treści zadania bez użycia tablicy dwuwymiarowej (macierzy).

```
function ilorazyRoznicowe (x::VectorFloat64, f::VectorFloat64)
```

1.2 Analiza

1.3 Implementacja

Moja implementacja algorytmu w języku Julia znajduje się w module `MyModule` znajdującym się w pliku załączonym do tego sprawozdania.

2 Zadanie 2

2.1 Opis problemu

Celem zadania było stworzenie funkcji obliczającej wartość wielomianu interpolacyjnego stopnia n w postaci Newtona $N_n(x)$ w punkcie $x = t$ za pomocą uogólnionego algorytmu Hornera w czasie $O(n)$. (implementacja zadania 8 z listy nr 4 z ćwiczeń)

```
function warNewton (x::VectorFloat64, fx::VectorFloat64, t::Float64)
```

2.2 Analiza

2.3 Implementacja

Moja implementacja algorytmu w języku Julia znajduje się w module `MyModule` znajdującym się w pliku załączonym do tego sprawozdania.

3 Zadanie 3

3.1 Opis problemu

W tym zadaniu należało dla zadanych współczynników wielomianu interpolacyjnego w postaci Newtona $c_0 = f[x_0]$, $c_1 = f[x_0, x_1]$, $c_2 = f[x_0, x_1, x_2]$, \dots , $c_n = f[x_0, \dots, x_n]$ (ilorazy różnicowe) oraz węzłów x_0, x_1, \dots, x_n napisać funkcję obliczającą w czasie $O(n^2)$ współczynniki a_0, \dots, a_n jego postaci naturalnej, tzn. $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$

3.2 Analiza

3.3 Implementacja

Moja implementacja algorytmu w języku Julia znajduje się w module `MyModule` znajdującym się w pliku załączonym do tego sprawozdania.

4 Zadanie 4

4.1 Opis problemu

W tym zadaniu należało napisać funkcję, która zinterpoluje zadaną funkcję $f(x)$ w przedziale $[a, b]$ za pomocą wielomianu interpolacyjnego stopnia n w postaci Newtona, a następnie wygeneruje wielomian interpolacyjny i interpolowaną funkcję. (np. przy pomocy pakietu `Plots`, `PyPlot` lub `Gadfly`). Do interpolacji należało użyć węzłów równoodległych, tj. $x_k = a + kh, k = \frac{b-a}{n}, k = 0, 1, \dots, n$.

4.2 Analiza

4.3 Implementacja

Moja implementacja algorytmu w języku Julia znajduje się w module `MyModule` znajdującym się w pliku załączonym do tego sprawozdania.

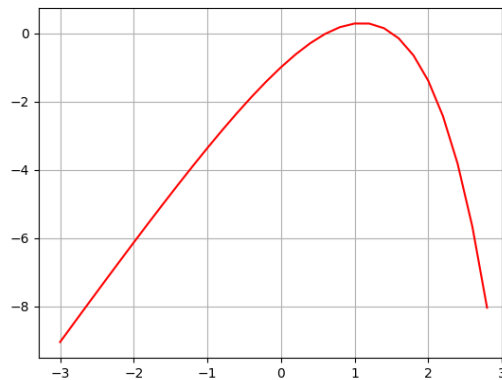
5 Zadanie 5

5.1 Opis problemu

Zadanie polegało na znalezieniu wartości zmiennej x , dla której przecinają się wykresy funkcji $f(x) = 3x$ i $g(x) = e^x$. Wymagana dokładność obliczeń to $\delta = 10^{-4}$, $\epsilon = 10^{-4}$

5.2 Rozwiązanie

W celu wyznaczenia takiego punktu, tj. pary (x, y) , zestawilem ze sobą te funkcje, tj. $f(x) = g(x) \Rightarrow 3x = e^x \Rightarrow 3x - e^x = 0$ i przeanalizowałem nowopowstałą funkcję, tj. $h(x) = 3x - e^x$. Za pomocą biblioteki matplotlib wygenerowałem wykres tej funkcji. Z analizy wykresu doszedłem do tego, że funkcja ta ma dwa miejsca zerowe, jedno wśród argumentów z zakresu $x \in (0.0, 1.0)$ oraz drugie z zakresu $x \in (1.0, 2.0)$. W związku z zaobserwowanymi własnościami funkcji $h(x)$ za pomocą funkcji `mbisekcji` odnalazłem miejsca zerowe z dokładnością wymaganą w treści zadania.



Rysunek 1: Wykres $g(x) = 3x - e^x$

5.3 Wynik

Wyniki zestawilem w tabeli poniżej:

Przedział	x	$f(x)$	i	err
(0.0, 1.0)	0.619140625	9.066320343276146e-5	9	0
(0.1, 2.0)	1.5120849609375	7.618578602741621e-5	13	0

Porównując (w pakiecie matematycznym) wyniki uzyskane za pomocą metody bisekcji z rzeczywistymi wartościami, możemy stwierdzić, że wartości te są poprawne (z uwzględnieniem błędu). W metodzie bisekcji bardzo ważnym jest, aby odpowiednio dobrać przedziały (warunki początkowe). Próby eksperymentalnego znalezienia miejsc zerowych na przedziale $\langle 0, 2 \rangle$ nie doprowadziły do poprawnego wyniku. (nie spełniał on założenia o różnych znakach wartości na krańcach przedziału)

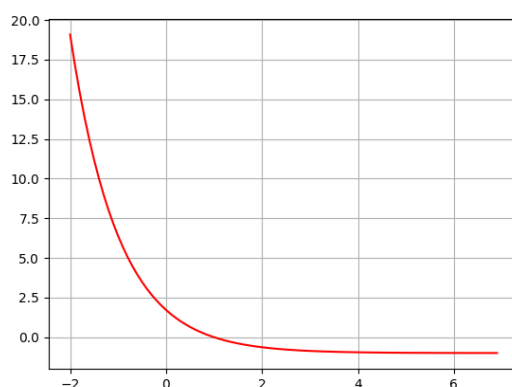
6 Zadanie 6

6.1 Opis problemu

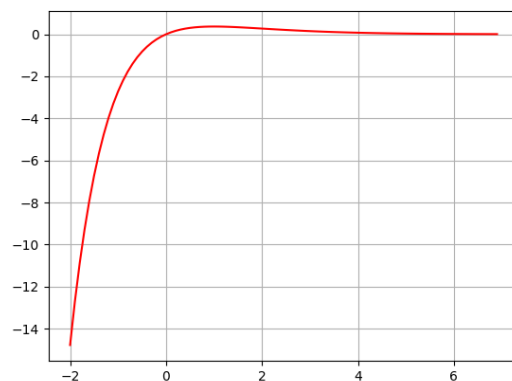
Celem tego zadania było znalezienie miejsc zerowych funkcji $f_1(x) = e^{1-x} - 1$ oraz $f_2(x) = xe^{-x}$ za pomocą wcześniej zaimplementowanych metod przy dokładności $\delta = 10^{-5}$, $\epsilon = 10^{-5}$. Dobrać odpowiednio przedział i przybliżenie początkowe.

6.2 Rozwiązanie

Na samym początku, dla ułatwienia zadania, za pomocą biblioteki `matplotlib` wygenerowałem wykresy funkcji $f_1(x)$ oraz $f_2(x)$.



Rysunek 2: Wykres $f_1(x) = e^{1-x} - 1$



Rysunek 3: Wykres $f_2(x) = xe^{-x}$

Z analizy wykresów (Rysunek 2, Rysunek 3) zaobserwowałem, że poszukiwane miejsca zerowe znajduje się w przypadku $f_1(x)$ przedziale $x \in [0.0, 2.0]$, a w przypadku $f_2(x)$ przedziale $x \in [-1.0, 1.0]$. Krańce tych przedziałów przyjąłem za punkty początkowe w przypadku obliczeń metodą siecznych, tych przedziałów użyłem również do wyliczenia wyniku metodą bisekcji. Za punkt początkowy, w przypadku metody stycznych, przyjąłem -0.5 .

6.3 Wynik

Wyniki zestawilem w tabelach poniżej:

Dla $f_1(x) = e^{1-x} - 1$:

Metoda	x	$f(x)$	i	err
Bisekcja	1.0	0.0	1	0
Stycznych	0.9999922654776594	7.734552252003368e-6	5	0
Siecznych	1.0000017597132702	-1.7597117218937086e-6	6	0

Dla $f_2(x) = xe^{-x}$:

Metoda	x	$f(x)$	i	err
Bisekcja	0.0	0.0	1	0
Stycznych	-3.0642493416461764e-7	-3.0642502806087233e-7	4	0
Siecznych	1.744165849924562e-8	1.7441658195034172e-8	18	0

Podczas wyliczania miejsc zerowych funkcji przedstawionych w treści zadania zauważyłem, że wybór punktów w metodzie bisekcji zawsze doprowadzi nas do rozwiązania. Oczywiście wybranie punktów startowych odległych od miejsca zerowego o $x_{00} = x_0 - \delta$ oraz $x_{01} = x_0 + \delta$ w łatwy sposób pozwoli trafić w miejsce zerowe. Przykładowe wyniki dla bisekcji dla lekko przesuniętych miejsc zerowych to odpowiednio

dla $f_1 \rightarrow (0.999993896484375, 6.1035342515669555e-6, 16, 0)$

dla $f_2 \rightarrow (-3.051757812455591e-6, -3.051767125695548e-6, 17, 0)$

W kwestii zbieżności - globalna zbieżność odróżnia metodę bisekcji od reszty. W przypadku pozostałych dwóch - metody stycznych oraz metody siecznych, dobór punktów startowych jest istotnym elementem. Metody te są jedynie lokalnie zbieżne, a więc złe punkty początkowe nie doprowadzają do dobrego wyniku. Przykładem źle dobranych punktów startowych są zestawy parametrów startowych przedstawione w kolejnej sekcji.

6.4 Dobór parametrów

Dodatkowym punktem zadania było sprawdzenie zachowania metody Newtona dla pewnych parametrów:

1. dla f_1 gdy $x_0 > 1$
2. dla f_2 gdy $x_0 > 0$
3. dla f_2 gdy $x_0 = 1$

Analiza wywołań:

1. W tym wypadku wartości zwracane przez metodę były akceptowalne do $x_0 = 7.4$. W kolejnych iteracjach testu $x_0 \in [7.6, 12.4]$ pojawił się błąd $err = 1$ (nieosiągnięto dokładności po `maxit` iteracjach), a wartości zwracane przez metodę to `NaN`. Powodem pojawienia się `NaN` tj. (not a number) był fakt, że w pewnym momencie pochodna, bliska zeru, przechodziła przez test $|f'(x_0)| < \epsilon$, ale w momencie, gdy algorytm wyliczał $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$, to dochodziło do dzielenia przez zero. Dalsze iteracje zwracały błąd $err = 2$ świadczący o pochodnej bliskiej zeru - wtedy warunek $|f'(x_0)| < \epsilon$ zwracał `true` i metoda kończyła działanie.
2. Sprawdzając kolejne wywołania metody Newtona dla parametrów $x \in (0, \infty)$ zauważyłem, że do osiągnięcia $x = 1.0$, oscylowały w granicach realnej wartości. Przy $x = 1.0$ funkcja zwróciła `err = 2`, kolejne obliczenia, dla $x_0 > 1.0$ zwracały wartości znacznie odbiegające od realnej.
3. Po wywołaniu metody Newtona na f_2 z zadanymi paramterami otrzymujemy $(1.0, -0.0, 0, 2)$. Takie zachowanie (będące odstępstwem od punktów w otoczeniu tego punktu) związane jest z pochodną tej funkcji, tj. $f'_2(x) = -e^{-x}(x-1)$, która w $x_0 = 1$ osiąga swoje miejsce zerowe, a więc styczna $g(x) \parallel OX$, tj. nie spełnia warunków metody Newtona.

Powyższe zestawy pokazują, że metody Newtona i siecznych są jedynie zbieżne lokalnie, a źle dobrane parametry początkowe uniemożliwiają osiągnięcie rozsądnych wyników.