Obliczenia Naukowe- Lista 3

Szymon Skoczylas

22 listopada 2020

Zadanie 1

Cel zadania

Celem zadania jest zaimplementowanie w języku *Julia* algorytmu bisekcji.

Opis i wykonanie

Algorytm służy do szukania miejsc zerowych danej funkcji $f(x)$, na danym przedziale $[a, b]$. Algorytm do działania potrzebuje następujących warunków;

1. funkcja f jest ciągła na $[a, b]$
2. $f(a)f(b) < 0$, czyli f na zadanym przedziale zmienia znak

Algorytm przedstawia się następująco:

```
function mbisekcji(f, a::Float64, b::Float64, delta::Float64, epsilon::Float64)
    r, v = Float64(0.0), Float64(0.0)
    it, err = 0, 0
    middle_of_rng = b - a
    f_at_a, f_at_b = f(a), f(b)

    if signbit(f_at_a) == signbit(f_at_b)
        return (r, v, it, 1)
    end

    while middle_of_rng > epsilon
        it += 1

        middle_of_rng /= 2.0
        r = a + middle_of_rng
        v = f(r)

        if abs(middle_of_rng) < delta || abs(v) < epsilon
            return (r, v, it, err)
        end

        signbit(v) != signbit(f_at_a) ? (f_at_b, b = v, r) : f_at_a, a = v, r
    end
end
```

Przedstawiony wyżej algorytm znajduje miejsce zerowe funkcji z dokładnością do danego epsilona i delty. Przbieg samego algorytmu można opisać następująco:

1. Sprawdzenie czy $f(x) = 0$, gdzie $x = \frac{a+b}{2}$. Jeśli dany warunek jest spełniony algorytm kończy działanie i x jest szukanym punktem.

2. Jeśli punkt 1 nie został spełniony, dopóki $|a - b| > \epsilon$:

   (a) Dzielimy przedział $[a, b]$ na dwa mniejsze $[a, x]$, oraz $[x, b]$.

   (b) Wybieramy przedział o znaku przeciwnym do dotychczasowego i zastępujemy nim przedział który rozpatrywaliśmy do tej pory.

3. Powyższe kroko wykonujemy do momentu, aż wartość środka przedziału $(\frac{a+b}{2})$ będzie równa 0 (co do zadanej dokładności).

Zadanie 2

Cel zadania

Celem zadania jest zaimplementowanie w języku *Julia* metody stycznych, zwanej również metodoą Newtona.

Opis i wykonanie

Algorytm służy do szukania miejsc zerowych danej funkcji $f(x)$, na danym przedziale $[a, b]$. Algorytm do działania potrzebuje następujących warunków:

1. W zadanym przedziale $[a, b]$ istnieje dokładnie jeden pierwiastek.

2. $f(a)f(b) < 0$, czyli f na zadanym przedziale zmienia znak

3. Znaki pierwszej i drugiej pochodnej funkcji f nie zmieniają się w przedziale $[a, b]$

Poniższy kod przedstawia algorytm stycznych:

```
function mstycznych(f, pf, x0::Float64, delta::Float64, epsilon::Float64, maxit::Int)
    it, err = 0, 0
    r = x0
    v = f(r)

    if abs(pf(r)) < epsilon
        return (r, v, it, 2)
    end

    for i in 1:maxit
        x1 = r - v / pf(r)
        v = f(x1)
        if abs(v) < epsilon || abs(x1 - r) < delta
            r = x1
            return (r, v, i, err)
        end
        r = x1
    end

    return (r, v, it, 1)
end
```

Na początku metody Newtona wybierany jest pewien punkt startowy- x. W punkcie tym następnie konstruujemy styczną do wykresu naszej funkcji f. Przecięcie stycznej z osią X jest przybliżeniem pierwiastka funkcji. Jeśli przybliżenie to nie odpowiada zadanej przez nas dokładności to powtarzamy cały proces, aż do otrzymania satysfakcjonującego wyniku. Przybliżenia dane są następującym wzorem $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$. Warunkami kończącymi poszukiwanie miejsca zerowego są (musi być spełniony chociaż jeden):

1. $|f(x_n)| \leqslant \epsilon$, wartość funkcji w x_n jest bliska 0 na tyle, że odpowiada zadanej przez nas dokładności.

2. $|x_{n+1} - x_n| \leqslant \delta$

Zadanie 3

Cel zadania

Celem zadania jest zaimplementowanie w języku *Julia* metody siecznych, zwanej również metodoą Eulera.

Opis i wykonanie

Kod implementujący metodę siecznych:

```
function msiecznych(f, x0::Float64, x1::Float64, delta::Float64, epsilon::Float64, maxit::Int)
    r, v = Float64(0.0), Float64(0.0)
    it, err = 0, 0
    f_x0, f_x1 = f(x0), f(x1)

    for it in 1:maxit
        if abs(f_x0) > abs(f_x1)
            x0, x1 = x1, x0
            f_x0, f_x1 = f_x1, f_x0
        end

        s = (x1 - x0) / (f_x1 - f_x0)
        f_x1, x1 = f_x0, x0
        x0 -= f_x0 * s
        f_x0 = f(x0)

        if abs(x1 - x0) < delta || abs(f_x0) < epsilon
            r, v = x0, f_x0
            return(r, v, it, err)
        end
    end

    v, r = f_x0, x0
    return(r, v, it, 1)
end
```

Metoda ta działa w oparciu o założenie, że funkcja ciągła na małym odcinku zmienia się liniowo. Na danym $[a, b]$, krzywą f zastępujemy sieczną. Pierwiastkiem jest punkt przecięcia siecznej z osią X. Sposób działania tego algorytmu jest analogiczny do algorytmu Newtona, z tym, że styczne zastąpiono siecznymi. Metodę tą opisuje następujący wzór:

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{x-1}}{f(x_n) - f(x_{n-1})}$$

Jeśli przybliżenie to nie odpowiada zadanej przez nas dokładności to powtarzamy cały proces, aż do otrzymania satysfakcjonującego wyniku. Aby metoda siecznych działała muszą zachodzić następujące warunki:

1. $f(a)f(b) < 0$, czyli funkcja musi zmieniać znak na danym przedziale.

2. funkcja f jest ciągła na $[a, b]$

Zadanie 4

Cel zadania

Celem tego zadania było obliczenie miejsc zerwych funkcji $sinx - \left(\frac{1}{2}x\right)^2 = 0$ za pomocą wcześniej zapimplementowanych metod, dla danych argumentów:

1. Dla metody bisekcji: na przedziale $[1.5, 2]$, $\delta = \frac{1}{2}10^{-5}$, $\epsilon = \frac{1}{2}10^{-5}$

2. Dla metody Newtona: $x_0 = 1.5$, $\delta = \frac{1}{2}10^{-5}$, $\epsilon = \frac{1}{2}10^{-5}$

3. Dla metody siecznych: $x_0 = 1$, $x_1 = 2$, $\delta = \frac{1}{2}10^{-5}$, $\epsilon = \frac{1}{2}10^{-5}$

Wykonanie i wyniki

Wyniki wywołania każdej z metod z danymi wyżej parametrami (kolejno: $x0$, $f(x_0)$, liczba iteracji, err):

```
[szymo@szymoHost lista_3]$ julia zad4.jl
Metoda bisekcji: (1.9337539672851562, -2.7027680138402843e-7, 16, 0)
Metoda siecznych: (1.933753644474301, 1.564525129449379e-7, 4, 0)
Metoda stycznych: (1.933753779789742, -2.2423316314856834e-8, 4, 0)
```

Jak widać wyniki zwracane przez każdą z metod są bardzo zbliżone do siebie. Różnice zaczynają się na siódmym miejscu po przcinku. Można też zawuażyć, że metody stycznych i siecznych zbiegają szybciej, niż metoda bisekcji. Metoda ta potrzebowała aż 16 iteracji, czyli czterokortnie więcej niż pozostałe.

Zadanie 5

Cel zadania

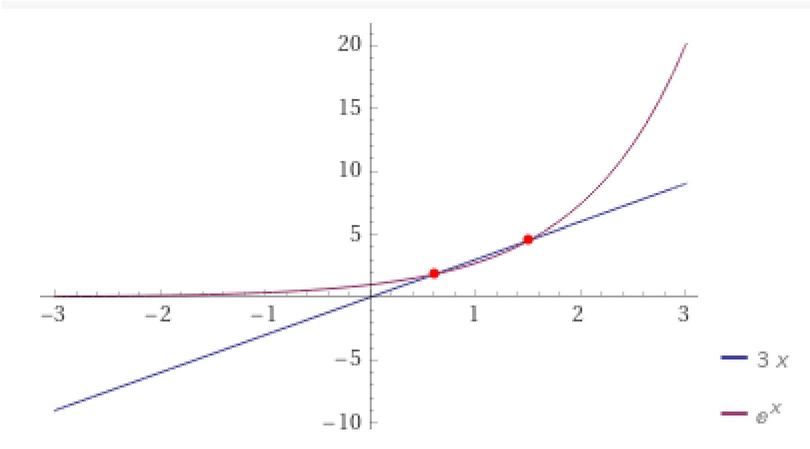
Celem tego zadania jest znalezienie wartości x, za pomocą metody bisekcji, takiej, że przecinają się wykresy dwóch funkcji $y = 3x$ oraz $y = e^x$. Zadana dokładność obliczeń: $\delta = 10^{-4}$, $\epsilon = 10^{-4}$.

Wykonanie i wyniki

W celu znalezienia miejsc przecięcia dwóch funkcji posłużono się następującym równaniem:

$$3x - e^x = 0$$

Wykres $y = e^x$ i $y = 3x$ uzystany za pomocą $WolframAlpha$:



Równianie $3x - e^x = 0$ przekazano do funkcji realizującą metodę bisekcji jako parametr f. Przeprowadzono testy dla czterech przedziałów: $[0.5, 1]$, $[0, 1]$, $[1, 2]$, $[1.5, 2]$. Otrzymane wyniki (kolejno: $x0$, $f(x_0)$, liczba iteracji, err):

```
[szymo@szymoHost lista_3]$ julia zad5.jl
----------------------------------------
Przedzial [0.5, 1.0]:
(0.619140625, 9.066320343276146e-5, 8, 0)
----------------------------------------
Przedzial [0.0, 1.0]:
(0.619140625, 9.066320343276146e-5, 9, 0)
----------------------------------------
Przedzial [1.0, 2.0]:
(1.5120849609375, 7.618578602741621e-5, 13, 0)
----------------------------------------
Przedzial [1.5, 2.0]:
(1.5120849609375, 7.618578602741621e-5, 12, 0)
----------------------------------------
```

Jak widać po wykresie i wynikach algorytmu funkcje te przecinają się w dwóch miejscach. Można również zauważyć, że im mniejszy przedział, tym szybciej miesce zerowe jest znajdowane. Trzeba też mieć też jakiekolwiek pojęcie o samym przebiegu funkcji, aby dobrze dobrać przedział w którym będziemy szukać miejsca zerowego, w przeciwnym wypdaku może to być poszukiwanie na ślepo.

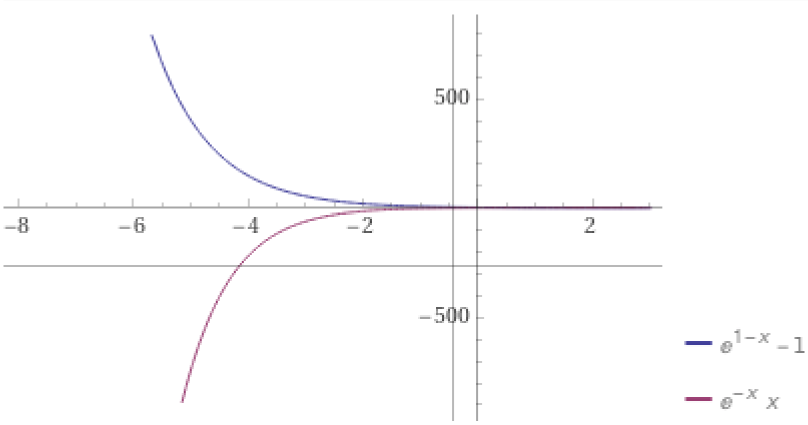
Zadanie 6

Cel zadania

Znaleźć miejsca zerowe funkcji $f_1(x) = e^{1-x} - 1$ oraz $f_2(x) = xe^{-x}$ za pomocą każdej z trzech zaimplementowancyh metod. Zadana dokładność obliczeń: $\delta = 10^{-5}$, $\epsilon = 10^{-5}$.

Wyniki i wykonanie

Wykres $f_1(x) = e^{1-x} - 1$ oraz $f_2(x) = xe^{-x}$ wygenerowany za pomocą $WolframAlpha$:



Poniżej wyniki poszukiwania pierwiastków funkcji różnymi metodami (kolejno: bisekcji, stycznych i siecznych) na danych przedziałach, bądź punktów początkowych:

```
[szymo@szymoHost lista_3]$ julia zad6.jl
--------------------METODA BISEKCJI--------------------
f1:
[0, 2] (1.0, 0.0, 1, 0)
[0.5, 1.5] (1.0, 0.0, 1, 0)
[-2, 2] (1.0, 0.0, 2, 0)
[-200, 200] (0.9999990463256836, 9.536747711536009e-7, 24, 0)
[0.4, 1.5] (1.0000091552734374, -9.155231528001906e-6, 15, 0)
f2:
[0, 2] (0.0, 0.0, 0, 1)
[0.5, 1.5] (0.0, 0.0, 0, 1)
[-2, 2] (0.0, 0.0, 1, 0)
[-200, 200] (0.0, 0.0, 1, 0)
[0.4, 1.5] (-1.5258789062909029e-6, -1.525881234599116e-6, 16, 0)
--------------------METODA STYCZNYCH--------------------
f1:
x0=0 (0.9999984358892101, 1.5641120130194253e-6, 4, 0)
x0=-1 (0.9999922654776594, 7.734552252003368e-6, 5, 0)
x0=0.5 (0.9999999998878352, 1.1216494399945987e-10, 4, 0)
x0=1 (1.0, 0.0, 1, 0)
x0=3 (0.9999999710783241, 2.892167638712806e-8, 9, 0)
x0=4 (0.9999999995278234, 4.721765201054495e-10, 21, 0)
x0=6 (-78.4131591025766, 3.081036113119517e34, 0, 1)
x0=8 (NaN, NaN, 0, 1)
x0=15 (15.0, -0.9999991684712809, 0, 2)
f2:
x0=0 (0.0, 0.0, 1, 0)
x0=-1 (-3.0642493416461764e-7, -3.0642502806087233e-7, 5, 0)
x0=0.5 (-3.0642493416461764e-7, -3.0642502806087233e-7, 5, 0)
x0=1 (1.0, 0.36787944117144233, 0, 2)
x0=3 (14.787436802837927, 5.594878975694858e-6, 10, 0)
x0=4 (14.398662765680003, 8.036415344217211e-6, 9, 0)
x0=6 (14.97432014974184, 4.699833827208111e-6, 8, 0)
x0=8 (14.636807965014, 6.438155219843286e-6, 6, 0)
x0=15 (15.0, 4.588534807527386e-6, 0, 2)
--------------------METODA SIECZNYCH--------------------
f1:
x0=0, x1=2 (1.0000017597132702, -1.7597117218937086e-6, 6, 0)
x0=0.5, x1=1.5 (0.9999999624498374, 3.755016342310569e-8, 5, 0)
x0=-2, x1=2 (1.0000000080618678, -8.061867839970205e-9, 8, 0)
x0=-200, x1=200 (200.0, -1.0, 1, 0)
x0=0.4, x1=1.5 (0.999997196847466, 2.8031564627273298e-6, 5, 0)
f2:
x0=0, x1=2 (0.0, 0.0, 1, 0)
x0=0.5, x1=1.5 (5.372227830220525e-6, 5.372198969466189e-6, 9, 0)
x0=-2, x1=2 (14.294924723787231, 8.85064549833867e-6, 15, 0)
x0=-200, x1=200 (200.0, 2.767793053473475e-85, 1, 0)
x0=0.4, x1=1.5 (14.258897288992923, 9.152200589944569e-6, 8, 0)
```

Wnioski

Można zauważyć, że dla $f1$, metoda bisekcji niezależnie od danego przedziału zawsze znajduje pierwiastek (co do określonej dokładności). Dla przedziału w którym miejsce zerowe jest blisko środka przedziału, algorytm ten potrzebuje bardzo mało iteracji aby to miejsce znaleźć. Dla $f2$ natomiast wyniki są niejednoznaczne, dla części danych funkcja praktycznie odrazu zwracała błąd.
Dla metody stycznych, dla $f1$ przekazanie $x_0 \in (1, \infty]$ powoduje wzrost liczby iteracji potrzebnych do znalezienia pierwiastka. Dla $f2$ gdy przekażemy $x_0 = 1$ funkcja kończy działanie, ponieważ wartość $f2$ dla 1 jest wystarczająco bliska zera. Im większe x_0 tym mniej dokładne wyniki są zwracane przez algorytm. Znaczy to, że metoda ta ma coraz gorszą dokładnośc dla miejsc położonych coraz dalej od faktycznego pierwiastka.
W przypadku metody siecznych, analogiczne jak w metodzie stycznych, mi dalsze od faktycznego pierwiastka są x_0 i x_1, tym gorsze przybliżenia zwraca funkcja.