Sprawozdanie z listy trzeciej

Karolina Bąk

Listopad 2019

1 Zadanie 1

Celem zadania było zaimplementowanie metody bisekcji do rozwiązywania f(x)=0. Funkcja miała spełniać poniższą specyfikację:

Dane:

- **f** - f(x) jako anonimowa funkcja
- **a**, **b** - końce przedziału początkowego
- **delta**, **epsilon** - dokładności obliczeń

Wyniki:

- **r** - przybliżenie pierwiastka
- **v** - wartość f(r)
- **it** - ilość iteracji
- **err** - błąd
  - **0** - brak
  - **1** - funkcja nie zmienia znaku w [a,b]

Opis algorytmu:

Na początku przypisuję wartości końców odcinka do zmiennych u i w, a długość odcinka do e. Jeśli znaki zmiennych u i w są takie same, zwracam błąd (1), gdyż nie można wtedy zagwarantować, że gdzieś w tym przedziale będzie pierwiastek funkcji. W przeciwnym razie metoda działa dla funkcji ciągłych. Następnie w pętli zmniejszam przeszukiwany odcinek o połowę, sprawdzając środkową wartość w odcinku. Jeśli jest ona poprawna zgodnie z narzuconą dokładnością, zwracam wynik. Jeśli nie, to sprawdzam jej znak ze znakiem wartości funkcji na początku odcinka. Jeśli są różne, to szukam dalej pierwiastka w pierwszej połowie odcinka. Jeśli nie, szukam pierwiastka w drugiej połowie. Powtarzam proces, dopóki nie trafię na poprawny (w granicy błędu) wynik.

Pseudokod:

```
0. M - bezpiecznik przeciwko nieskończonej pętli - typemax(Int32)
1. u = f(a); w = f(b);
2. e = b-a
3. if sign(u) == sign(w)
   return(a,u,0,1)
4. for it in 1:M
4.1 e = e/2
    c = a+e
    v = f(c)
4.2 if |e| < delta or |v| < epsilon
    return(c,v,it,0)
4.3 if sign(v) != sign(u)
    b = c
    w = v
   else
    a = c
    u = v
```

2 Zadanie 2

Celem zadania było zaimplementowanie metody Newtona do rozwiązywania f(x)=0. Funkcja miała spełniać poniższą specyfikację:

Dane:

- **f** - f(x) jako anonimowa funkcja
- **pf** - f'(x) jako anonimowa funkcja
- **x0** - przybliżenie początkowe
- **delta**, **epsilon** - dokładności obliczeń
- **maxit** - maksymalna liczba iteracji

Wyniki:

- **r** - przybliżenie pierwiastka
- **v** - wartość f(r)
- **it** - ilość iteracji

- **err** - błąd
  - **0** - metoda zbieżna
  - **1** - nie znaleziono wyniku w maxit iteracji
  - **2** - pochodna bliska zeru

Opis algorytmu:

Na początku działania funkcji sprawdzam, czy udało się trafić w pierwiastek przy warunkach początkowych. Jeśli tak, to zwracam tą wartość. Jeśli nie, kontynuuję. Rozpoczynam pętle od 1 do maxit, w której na początku sprawdzam czy pochodna w x0 jest mniejsza niż wymagana dokładność. Jeśli tak, to przyjmuję, że jest bliżej zera niż powinna i zwracam błąd (2). Następnie obliczam następny x, czyli x1 (punkt przecięcia stycznej do funkcji w x0 z osią OX) równy $x0 - \frac{f(x0)}{f'(x0)}$. Potem sprawdzam czy x1 jest dobrym przybliżeniem pierwiastka. Jeśli tak, to zwracam uzyskaną wartość. Jeśli nie, to za x0 podstawiam x1 i kontynuuję poszukiwania. Jeśli po maxit iteracji nie uda mi się uzyskać odpowiedniego wyniku, zwracam błąd (1).

Pseudokod:

---

1. v = f(x0);
2. **if** |v| < epsilon
   return(x0,v,0,0)
3. **for** it in 1:maxit
3.1 fprim = pf(x0)
3.2 **if** |fprim| < delta
   return(x0,v,it,2)
3.3 x1 = x0-v/fprim
   v = f(x1)
3.4 **if** |x1-x0| < delta **or** |v| < epsilon
   return(x1,v,it,0)
3.5 x0 = x1
4. **return**(x1,v,maxit,1)

---

3 Zadanie 3

Celem zadania było zaimplementowanie metody siecznych do rozwiązywania f(x)=0. Funkcja miała spełniać poniższą specyfikację:

Dane:

- **f** - f(x) jako anonimowa funkcja

- $\mathbf{x0}, \mathbf{x1}$ - przybliżenia początkowe
- $\mathbf{delta}, \mathbf{epsilon}$ - dokładności obliczeń
- **maxit** - maksymalna liczba iteracji

Wyniki:

- **r** - przybliżenie pierwiastka
- **v** - wartość f(r)
- **it** - ilość iteracji
- **err** - błąd
  - **0** - metoda zbieżna
  - **1** - nie znaleziono rozwiązania w maxit iteracji

Opis algorytmu:

Zaczynam od ustalenia wartości funkcji dla punktów przybliżających. Potem zaczynam pętle od 1 do maxit. Jeśli |f(x0)| jest większe od |f(x1)| to zamieniam miejscami wartości zmiennych x0 oraz x1. Następnie wyliczam s równe $\frac{x1-x0}{f(x1)-f(x0)}$, które jest przybliżeniem pochodnej f(x), co pozwoli mi obliczyć następne x0 równe $x0-f(x0)*s$, które jest miejscem gdzie sieczna przechodząca przez x0 i x1 przecina oś OX. Następne x1 to stare x0. W kolejnym kroku sprawdzam, czy udało mi się przybliżyć pierwiastek. Jeśli tak, to zwracam uzyskany wynik. W przeciwnym razie kontynuuję obliczenia. Jeśli nie uda się znaleźć rozwiązania w maxit iteracji, zwracam błąd (1).

Pseudokod:

```
1. fx0 = f(x0); fx1 = f(x1)
2. for it in 1:maxit
2.1 if |fx0| > |fx1|
    swap(x0,x1)
    swap(fx0,fx1)
2.2 s = (x1-x0)/(fx1-fx0)
    x1 = 0; fx1 = fx0
    x0 = x0-fx0*s; fx0 = f(x0)
2.3 if |x1-x0| < delta or |fx0| < epsilon
    return(x0,fx0,it,0)
3. return(x0,fx0,maxit,1)
```

4 Zadanie 4

Celem zadania było zastosowanie zaimplementowanych wcześniej metod do znalezienia pierwiastka równania $\sin(x) - (\frac{1}{2}x)^2 = 0$. Deltę i epsilon w każdej metodzie ustawiłam na $\frac{1}{2}10^{-5}$.

1. Metoda bisekcji dla przedziału [1.5, 2] zwraca:

$$(r = 1.9337539672851562, v = -2.7027680138402843e - 7, it = 16, err = 0)$$

2. Metoda Newtona z x0 = 1.5 zwraca:

$$(r = 1.933753779789742, v = -2.2423316314856834e - 8, it = 4, err = 0)$$

3. Metoda siecznych z x0 = 1 i x1 = 2 zwraca:

$$(r = 1.933753644474301, v = 1.564525129449379e - 7, it = 4, err = 0)$$

Wszystkie metody zwróciły ten sam pierwiastek z dokładnością do 10^{-6},więc każda z nich uzyskała poprawny wynik dla podanych dokładności. Najszybsza i najbardziej dokładna była metoda Newtona, a najwolniejsza i najmniej dokładna metoda bisekcji. Dzieje się tak, ponieważ metoda bisekcji zawsze zgaduje, że pierwiastkiem jest środek zawężanego przedziału, w przeciwieństwie do innych metod, które przybliżają pierwiastek, upraszczając wykres funkcji. Metoda bisekcji jest jednak bezpieczniejsza od innych metod z tego zadania, ponieważ nie może być rozbieżna.

5 Zadanie 5