

Zadanie II.9

Damian Soliński

Piotr Kulas

Marek Skiba

04.12.2012

1 Treść zadania

Dla równania $f(x) = 0$, gdzie $f(x) = e^x + x + 2$, wczytać $a, b \in \mathbf{R}$ takie, by $a < b$ oraz $f(a) * f(b) < 0$. Następnie, dopóki "użytkownik się nie znudzi", wczytywać wartość $0 < \epsilon < 1$ i metoda połowienia na $[a, b]$ przybliżyć z dokładnością ϵ rozwiązanie tego równania. Rozwiązanie to przybliżyć również metoda Newtona z $x_0 = a$, przy czym x_k będzie dobrym przybliżeniem, gdy $|x_k - x_{(k-1)}| \leq \epsilon$. Porównać ilość kroków wykonanych metoda połowienia i metoda Newtona.

2 Schemat rozwiązywania zadania

Najpierw pobieramy od użytkownika dwie liczby: a i b , należące do liczb rzeczywistych, które spełniają: $a < b$ oraz $f(a) * f(b) < 0$. Następnie w petli aż użytkownik się nie znudzi pobieramy przybliżenie (ϵ) i liczymy dwoma sposobami: metoda równego podziału (metoda połowienia) oraz metoda Newtona.

Metoda równego podziału (metoda połowienia)

1. Sprawdzić, czy pierwiastkiem równania jest punkt $x_1 = \frac{a+b}{2}$, czyli czy $f(x_1) = 0$.
2. Jeżeli tak jest, algorytm kończy się, a punkt jest miejscem zerowym. W przeciwnym razie x_1 dzieli przedział $[a, b]$ na dwa mniejsze przedziały $[a, x_1]$ i $[x_1, b]$.
3. Wybierany jest ten przedział, dla którego spełnione jest drugie założenie, tzn. albo $f(x_1)f(a) < 0$ albo $f(x_1)f(b) < 0$. Cały proces powtarzany jest dla wybranego przedziału.

Działanie algorytmu kończy się w punkcie 2 albo po osiągnięciu żądanej dokładności przybliżenia pierwiastka.

Metoda Newtona

W pierwszym kroku metody wybierany jest punkt startowy x_0 (w naszym wypadku wartość liczby a). Wyliczamy x_1 ze wzoru $x_0 - \frac{f(x_0)}{df(x_0)}$. Następnie podkładamy wynik x_1 pod naszą funkcję $f(x) = e^x + x + 2$. Wynikiem funkcji będzie przybliżenie, jeśli nie jest satysfakcjonujące, wówczas punkt x_2 jest wybierany jako nowy punkt startowy i wszystkie czynności są powtarzane. Proces jest kontynuowany, aż zostanie uzyskane wystarczająco dobre przybliżenie pierwiastka.

3 Przykładowe rozwiązania:

Metoda równego podziału (metoda połowienia)

Ustalamy, że: $a = -3$, $b = -2$, $\epsilon = 0.01$

$$f(x) = e^x + x + 2$$

$$f(a) = -0,950212932$$

$$f(b) = 0,135335283$$

$$x_1 = \frac{a+b}{2} = -2.5$$

$$f(s) = e^{-2.5} - 2.5 + 2 = -0.41791$$

$$a = -2.5$$

$$b = -2$$

$$x_1 = \frac{a+b}{2} = -2.25$$

$$f(s) = e^{-2.25} - 2.25 + 2 = -0.144600775$$

$$a = -2.25$$

$$b = -2$$

$$x_1 = \frac{a+b}{2} = -2.125$$

$$f(s) = e^{-2.125} - 2.125 + 2 = -0,00556703173$$

Osiągneliśmy oczekiwane przybliżenie, pierwiastkiem równania $f(x)$ w podanym przedziale jest $x = -2.125$

Metoda Newtona

Ustalamy, że: $a = -3$, $b = -2$, $\epsilon = 0.01$

$$f(x) = e^x + x + 2$$

$$df(x) = e^x + 1$$

$$x_0 = a = -3$$

$$x_1 = -3 - \frac{f(-3)}{df(-3)} = -2,09485175$$

$$f(x_1) = -0,950212932$$

$$x_2 = -2,09485175 - \frac{f(-2,09485175)}{df(-2,09485175)} = -2,11999379$$

$$f(x_2) = 0,000038583910$$

Osiągneliśmy oczekiwane przybliżenie, pierwiastkiem równania $f(x)$ w podanym przedziale jest $x = -2,11999379$

4 Opis działania:

Na początku program zapyta o przedział (a i b). A powinno być mniejsze od B , wartość funkcji w tych punktach musi mieć różny znak. Przy wprowadzeniu błędnych wartości, program będzie pytał aż do skutku. Następnie program spyta o podanie przybliżenia, z przedziału od 0 do 1. Po tym kroku wyświetli się wynik znaleziony dwiema metodami (połowa i Newtona). W kolejnym etapie program zapyta się o chęć wprowadzenia kolejnej wartości przybliżenia. Można odmówić wpisując *nie*, i wtedy program kończy działanie, albo wpisać *tak* i podać kolejne przybliżenie.

5 Źródło programu:

```
#include <stdio.h>
#include <math.h>
double f(double x);
double df(double x);
void bisection(double a, double b, double e);
void newton(double x, double e);

int main() {
    double a, b, e;
    int status;
    char chce_podac_x[2];
    do {
        printf("Podaj_a:_");
        scanf("%lf", &a);
        printf("Podaj_b:_");
        scanf("%lf", &b);
        if(a < b) {
            if (f(a) * f(b) < 0) {
                do {
                    printf("Podaj_przyblizenie:_");
                    status = scanf("%lf", &e);
                    if(e > 0 && e < 1) {
                        bisection(a, b, e);
                        newton(a, e);
                    }
                }
                do {
                    printf("Chcesz_podać_kolejne_przyblizenie?(tak/nie):_");
                    scanf("%s", chce_podac_x);
                    if(strcmp(chce_podac_x, "tak") == 0 || strcmp(chce_podac_x, "nie") == 0)
                    } while(1);
                } while (strcmp(chce_podac_x, "tak") == 0);
            } else {
```

```

        printf("bład: f(a) i f(b) musza mieć różny znak\n");
    }
    } else {
        printf("bład: a musi być mniejsze od b\n");
    }
} while(a >= b || f(a) * f(b) >= 0);

return 0;
}

double f(double x) {
    return exp(x) + x + 2;
}

double df(double x) {
    return exp(x) + 1;
}

void bisection(double a, double b, double e) {
    double m = (a + b) / 2;
    double fm = f(m);
    int step = 0;
    while (fabs(fm) > e) {
        if ((f(a) * fm) < 0) {
            b = m;
        } else if ((f(b) * fm) < 0) {
            a = m;
        }
        m = (a + b) / 2;
        fm = f(m);
        step += 1;
    }
    printf("Metoda połowienia:\n");
    printf("x=%.16lf, f(x)=%.16lf, Kroki: %d\n", m, fm, step);
}

void newton(double x, double e) {
    int step = 0;
    while (fabs(f(x)) > e) {
        x -= f(x) / df(x);
        step += 1;
    }
    printf("Metoda Newtona:\n");
    printf("x=%.16lf, f(x)=%.16lf, Kroki: %d\n", x, f(x), step);
}

```

6 Wynik programu:

```
[mskiba@sigma] ~/MetodyObliczeniowe $ g++ zad2.cpp -o zad2
[mskiba@sigma] ~/MetodyObliczeniowe $ ./zad2
Podaj a: 10
Podaj b: 11
bład: f(a) i f(b) musza mieć różny znak
Podaj a: 11
Podaj b: 10
bład: a musi być mniejsze od b
Podaj a: -3
Podaj b: -2
Podaj przybliżenie: 2
Chcesz podać kolejne przybliżenie? (tak/nie): tak
Podaj przybliżenie: 0.01

Metoda połowienia:
x = -2.1250000000000000, f(x) = -0.0055670317332805, Kroki: 2

Metoda Newtona:
x = -2.1199937939314832, f(x) = 0.0000385795067956, Kroki: 2

Chcesz podać kolejne przybliżenie? (tak/nie): tak
Podaj przybliżenie: 0.0001

Metoda połowienia:
x = -2.1201171875000000, f(x) = -0.0000996243708498, Kroki: 9

Metoda Newtona:
x = -2.1199937939314832, f(x) = 0.0000385795067956, Kroki: 2

Chcesz podać kolejne przybliżenie? (tak/nie): nie
[mskiba@sigma] ~/MetodyObliczeniowe $
```