

# Rozwiązanie zadania N16

Krzysztof Waniak

Dane jest równanie

$$(x^2 - 1)\sinh^3 x = 0$$

Zastosuj algorytm siecznych i algorytm opracowany w zadaniu 8. z poprzedniego zestawu do znalezienia rozwiązania równania (1), startując, odpowiednio, z dwu i trzech losowych punktów z przedziału (0, 1). Punkty początkowe dla metody siecznych mają być dwoma z trzech punktów początkowych dla algorytmu opartego o iterację odwrotną. Wyznacz miejsce zerowe z dokładnością do  $10^{-8}$ . Powtórz zadanie dla kilku(nastu) różnych zestawów punktów początkowych.  
Porównaj wyniki.

Kod programu:

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<time.h>
#include<ctype.h>          /* zawiera F_OK itp.    */
#include<unistd.h>         /* zawiera funkcje access(), usleep() */
#define wyp(a) printf(#a "\n")
#define wyp2(a) printf(#a)
#define wypisz(a) printf("%.16f",a)
#define wypisz2(a) printf("%d",a)
#define wypisz3(a) printf("%i",a)
#define karetka printf("\n")
#define karetka2 printf("\n\n")
#define space printf(" ")

double losuj()
{
    return (double)((rand()% 9999)+1) /10000.0;
}

double f( double x )
{
    return (pow(x,2.0)-1)*pow(sinh(x),3.0);
}

double interpolacja_lagrangea(double x[], double y[], double x0 )
{
    double t;
    double wynik = 0.0;
    int k , i ;
    for( k = 0; k< 3; k++)
    {
        t = 1.0;
        for(i = 0; i < 3 ; i++)
        {
            if(i != k )
            {
                t = t*((x0-x[i])/(x[k]-x[i]));
            }
        }
    }
}
```

```

    }
    wynik += t*y[k];
}
return wynik;
}

double metoda_siecznych ( double x_1 , double x_2 , double eps)
{
    double f1 ,f0 ,f2 , x1=x_1 , x2=x_2 , x0;
    int i ,k=0;
    f1 = f(x1);
    f2 = f(x2);
    i = 64;

    while(i && ((fabs(x1 - x2) > eps)))
    {
        if(f1 - f2 == 0)
        {
            wyp(Zle punkty startowe);
            i = 0;
            break;
        }
        x0 = x1 - f1 * (x1 - x2) / (f1 - f2);
        f0 = f(x0);
        if(fabs(f0) < eps)
        {
            break;
        }
        x2 = x1; f2 = f1;
        x1 = x0; f1 = f0;
        if(!(--i))
        {
            wyp(Przekroczony limit obiegow);
        }
        k++ ;
    }

    karetka;
    wyp2(Metoda siecznych: iteracja);
    space;
    wypisz2(k);
    karetka;
    return x0;
}

double odwrocona_interpolacja(double a , double b , double c ,double
epsilon )
{
    int i = 0;
    double x0 = 100;
    while(fabs(f(x0)) > epsilon)
    {
        double x[4];
        double y[4];
        x[0] = a;
        x[1] = b;
        x[2] = c;
        y[0] = f(a);
        y[1] = f(b);
        y[2] = f(c);
    }
}

```

```

        x0 = (interpolacja_lagrangea(y,x,0.0));
        a = b;
        b = c;
        c = x0;
        i++;
    }

    karetka;
    wyp2(Odwrocona interpolacja: iteracja);
    space;
    wypisz2(i);
    karetka;
    return x0;
}

int main()
{
    int i;
    const double ypsilon = 0.00000001 ;
    double x0 , x1 , x2 ;
    srand(time(0));
    i = 0;
    while(i < 16)
    {
        wypisz3(i+1);
        space;
        wyp(LOSOWANIE);
        x0 = losuj () ;
        x1 = losuj () ;
        x2 = losuj () ;
        if ( x0 == x1 )
        {
            continue;
        }
        karetka2;
        wyp2(Metoda siecznych);
        space;
        wypisz(metoda_siecznych( x0 , x1 , ypsilon ));
        karetka;

        wyp2(Odwrocona interpolacja);
        space;
        wypisz(odwrocona_interpolacja ( x0 , x1 , x2 , ypsilon ));
        karetka;
        karetka2;
        i++ ;
    }
    /*system("pause");*/
    return 0;
}

```

Wynik działania programu:

```
C:\Documents and Settings\Wrzysiek\Pulpit\zadania_numerki_do_wyslania\N16\N16.exe
1 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 18
0.0018432068028955
Odwocona interpolacja
Odwocona interpolacja: iteracja 13
0.0020234794506416

2 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 18
0.0017139733973062
Odwocona interpolacja
Odwocona interpolacja: iteracja 8
0.0017058583037501

3 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 11
0.0016935116819379
Odwocona interpolacja
Odwocona interpolacja: iteracja 11
0.0018776736814356

4 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 17
0.0017120749082811
Odwocona interpolacja
Odwocona interpolacja: iteracja 16
0.0016194651509242

5 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 16
0.0017551597171683
Odwocona interpolacja
Odwocona interpolacja: iteracja 13
0.0015931335447210

6 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 14
0.0017675916261181
Odwocona interpolacja
Odwocona interpolacja: iteracja 13
0.0021139003396702

7 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 8
0.0016590704033527
Odwocona interpolacja
Odwocona interpolacja: iteracja 8
0.0018707998783372

8 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 12
0.0019823656560435
Odwocona interpolacja
Odwocona interpolacja: iteracja 12
0.0020846426282619

9 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 14
0.0018948520655265
Odwocona interpolacja
Odwocona interpolacja: iteracja 13
0.0016211862142459

10 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 11
0.0020340462591920
Odwocona interpolacja
Odwocona interpolacja: iteracja 12
0.0016217161397211
```

```

11 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 17
0.0019908866326490
Odwrócona interpolacja
Odwrócona interpolacja: iteracja 16
0.0017740906953026

12 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 18
0.0019920442591203
Odwrócona interpolacja
Odwrócona interpolacja: iteracja 16
0.0016288632543121

13 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 16
0.0016272130408993
Odwrócona interpolacja
Odwrócona interpolacja: iteracja 6
1.00000000000003952

14 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 19
0.9999999999998337
Odwrócona interpolacja
Odwrócona interpolacja: iteracja 12
0.0020673651659511

15 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 17
0.0017361344447718
Odwrócona interpolacja
Odwrócona interpolacja: iteracja 15
0.0017459239278462

16 LOSOWANIE

Metoda siecznych
Metoda siecznych: iteracja 17
0.0017601606008594
Odwrócona interpolacja
Odwrócona interpolacja: iteracja 15
0.0017301204300472

Aby kontynuować, naciśnij dowolny klawisz . . . _

```

Wyniki zgodne z przewidywaniami; metoda odwróconej interpolacji jest bardziej efektywna od metody siecznych. Metoda w przeważającej większości przypadków dla zadanej dokładności obliczeń znajduje miejsce zerowe bliższe rzeczywistemu w mniejszej ilości kroków niż metoda siecznych. Na dodatek, metoda siecznych nie zawsze musi zbiegać w przeciwieństwie do metody odwróconej interpolacji.