LAB 2 - Zagadnienie numerycznego wyznaczania zer nieliniowych funkcji

Maciej Pestka, Damian Szopiński

28 października 2022

Ćwiczenie polega na tym, aby wyznaczyć punktu x_0, które dążą do 0.

Metodę Newtona:

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

a kod funkcji:

```
for(int i=0;i<M;++i)
{
    xn=xn-(f(xn)/ff(xn));
    cout<<xn<<'\n';
    MetodaNewtonaF<<fixed<<setprecision(64)<<xn<<'\n';
}

```

Natomiast metoda siecznych:

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

ta metoda została następująca napisania w kodzie:

```
for(int i=0;i<M;++i)
{
  if((f(xn)-f(x0))==0){
    cout<<"mianownik jest rowny 0!\n";
  }else{
    xn=x1-f(x1)*((x1-x0)/(f(xn)-f(x0)));
  }
  cout<<xn<<'\n';
  MetodaSiecznychF<<fixed<<setprecision(64)<<xn<<'\n';
  x0=x1;
  x1=xn;
}

```

A metoda bisekcji została napisania w kodzie:

```
for(int i=0;i<M;++i)
{
  for(int k=1;k<M+1;++k)
  {
    e=e/2;
    c=a+e;
    w=f(c);
    if (abs(e)<sigma || abs(w)<epsilon){
      cout<<"Przerwano w M="<<k<<"\n";
      break;
      //return c;
    }
    if(signbit(w)!=signbit(v)){
      b=c;
```

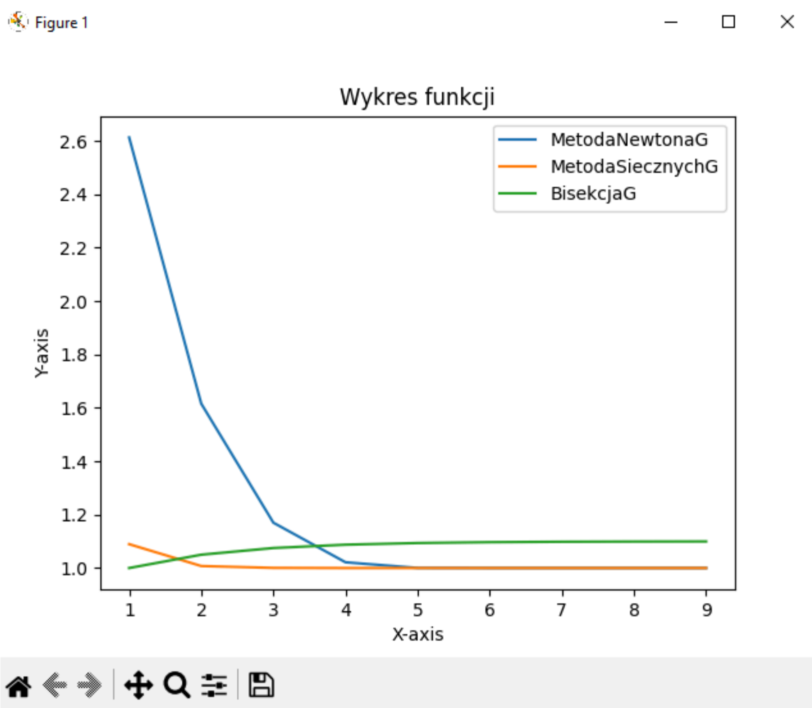
```
            v=w;
        }else{
            a=c;
            u=w;
        }
        cout<<c<<'\n';
        BisekcjaF<<fixed<<setprecision(64)<<c<<'\n';
    }
    cout<<c<<'\n';
}
```

Pierwszą funkcją jest wielomian $f(x) = x^3 + 2x^2 + x - 2$, który osiąga $y = 0$ dla 0.6956. Przyjmiemy $x_0 = 0.6$ oraz $x_1 = 0.7$.
Funkcja $g(x) = x^2 - x$ ten wielomian osiąga 2 punkty zerowe (0,0) oraz (1.0). przyjmujemy $x_0 = 1.4$ oraz $x_1 = 1.1$.
Funkcja $k(x) = \sin(x)$ osiąga zerowe cykliczne, ale my braliśmy pod uwagę (π,0) i przyjmujemy $x_0 = 4$ oraz $x_1 = 3.1$. Za σ i ϵ przyjęliśmy liczbę 0.0000000000000002220446049250313080847263336181640625, czyli najmniejszą liczbę jaką udało nam się odczytać z double z poprzedniego zadania laboratoryjnego.

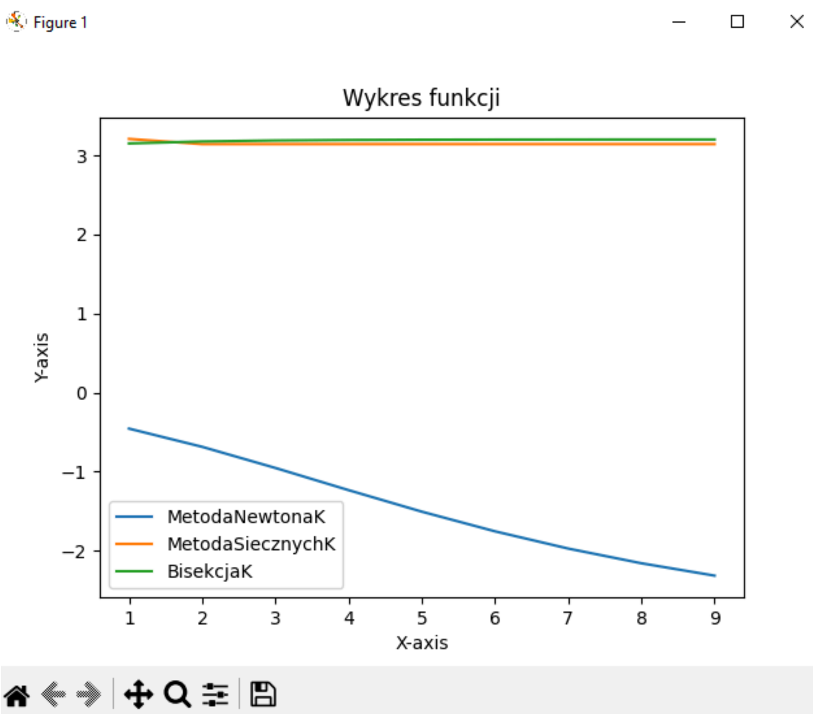
Przyjęliśmy 9 próbek, dla każdego z metod (nie licząc xn wstępnego w metodzie newtona)

Do rysowania wykresów wykorzystaliśmy bibliotekę Matplotlib języka python (Wersja python na którym wykonywane były testy to 3.10.6). Program cpp uruchamia ten skrypt z odpowiednimi argumentami (argumentami uruchomienia są nazwy plików z wynikami (.txt), nie ma limitu ilości argumentów, kolejne z nich będą rysowane na tym samym wykresie). W pliku cpp każdą metodę każdej funkcji oddzieliliśmy nawiasami klamrowymi by mieć pewność, że wyniki nie zostały odczytane z poprzedniego testu. Wersja kompilatora z jakiego korzystałem było mingw 12.2.0 -std=c++20. Oś x przedstawia liczbę iteracji, a y przybliżony wynik.

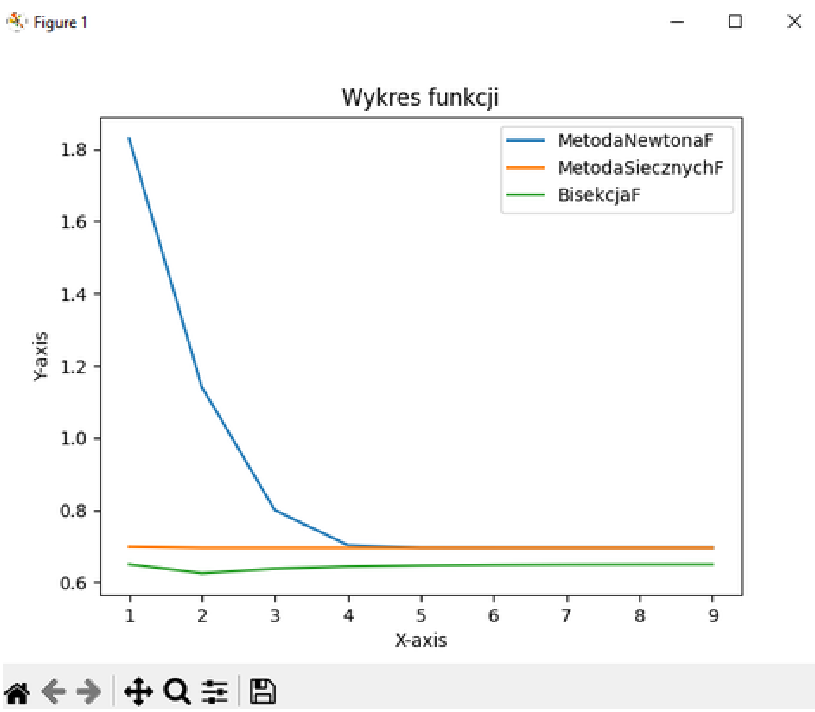
Z naszych pomiarów wynika, że najoptymalniejszą metodą jest kolejno:
dla f 1.metoda siecznych 2.metoda newtona, 3.bisekcja
dla g 1. metoda siecznych 2. metoda newtona 3. bisekcja
k 1.metoda siecznych 2. bisekcja 3.metoda newtona
Ranking polegał na tym że najpierw sprawdzano odchylenie przybliżenia od faktycznego wyniku, a gdy oba zbiegały do tej samej liczby, sprawdzano która z nich doszła do wyniku pierwsza



Rysunek 1: wykres1



Rysunek 2: wykres2



Rysunek 3: wykres3