MOWNIT - Laboratorium 13

Rozwiązywanie równań nieliniowych

Mikołaj Wróblewski

1. Metoda siecznych

Metoda siecznych jest metodą numeryczną służącą do rozwiązywania równań nieliniowych jednej zmiennej. Jest to algorytm interpolacji liniowej, polega ona na przyjęciu, że funkcja ciągła na dostatecznie małym odcinku przybliża się w sposób liniowy. Wtedy na odcinku <a,b> możemy funkcję y = f(x) zastąpić sieczną. Za przybliżony pierwiastek przyjmujemy wtedy przecięcie siecznej z osią OX. Generalnie posługujemy się wzorem:

$$\begin{cases} x_0 = a \\ x_1 = b \\ x_{n+1} = \frac{f(x_n)x_{n-1} - f(x_{n-1})x_n}{f(x_n) - f(x_{n-1})} \end{cases}$$

Wzór na kolejne x jest wzorem na miejsce zerowe prostej zadanej dwoma punktami. Generalnie iterujemy do momentu, w którym różnica pomiędzy x0 a x1 jest mniejsza od zadanego epsilon. Pamiętać musimy również o odgórnym ograniczeniu na maksymalną ilość iteracji.

2. Metoda Newtona - Raphsona

Metoda Newtona - Raphsona polega na obliczaniu stycznej do funkcji w punkcie startowym, miejsce w którym styczna do funkcji przecina oś OX jest naszym kolejnym x-em w iteracji, iterujemy do momentu, w którym różnica pomiędzy kolejnymi x-ami jest dostatecznie mała.

Wzór:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

Jest to wzór dla funkcji jednej zmiennej, nic nie stoi na przeszkodzie, by rozszerzyć tę metodę dla układów równań nieliniowych (tutaj przykład dla dwóch zmiennych):

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} - \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix}^{-1}_{(x_n, y_n)} * f\left(\begin{bmatrix} x_n \\ y_n \end{bmatrix}\right)$$

Oczywiście rozszerzenie na więcej niż dwie zmienne jest trywialne. Dodatkowo:

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} f_1(x,y) \\ f_2(x,y) \end{bmatrix}$$

## 3. Rozwiązanie zadania

Najpierw za zadanie mieliśmy obliczenie miejsc zerowych funkcji:

$$2 * x^2 + 1 = 2^x$$

Najpierw przy użyciu metody siecznych, wybieram 3 przedziały:

- a = -1, b = 10
- a = 5, b = 10
- a = 0.3 b = 2

Wyniki:

```
secant method
(-4.0470663985053456e-17, 13)
(6.352344892434116, 16)
(0.39928075666163776, 10)
```

Liczba po przecinku to liczba wykonanych iteracji.

Teraz wyniki dla metody Newtona-Raphsona, po obliczeniu pochodnej wybieram 3 punkty:

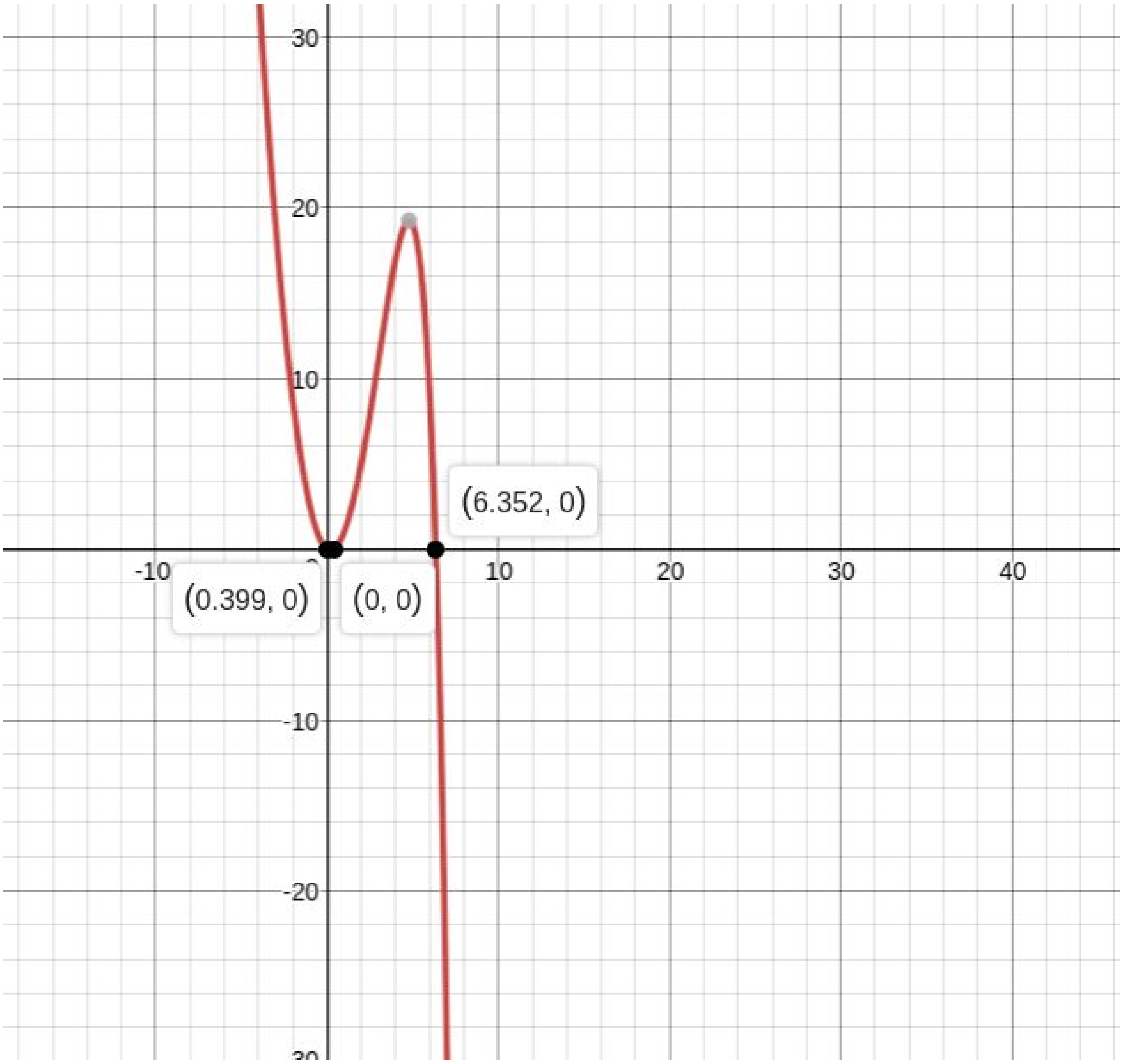
- a = 6
- a = 4
- a = 0

Wyniki:

```
newton-raphson method for first function
(6.352344892434116, 4)
(0.39928075666163754, 6)
(0.0, 1)
```

Jak widzimy potrzebowaliśmy mniej iteracji by uzyskać niemal identyczne wyniki.

Korzystając z narzędzi dostępnych online wykres funkcji wraz z miejscami zerowymi:




Oczywiście wykres ten nie daje tak dokładnych wyników jak powyższe metody.
Następnie używając metody Newtona za zadanie miałem rozwiązanie układu równań:

$x_1^2 + x_2^2 + x_3^2 = 1$

$2x_1^2 + x_2^2 - 4x_3 = 0$

$3x_1^2 - 4x_2 + x_3^2 = 0$

dla przybliżenia początkowego $x = [0, 0, 0]^T$

Wyniki:

```
newton-raphson for system of nonlinear equations
(array([0, 0, 0]), 'Jacobian not invertible at iteration:', 0)
```

Jak widzimy dla tego wektora startowego Jacobian nie jest odwracalny. By skorzystać z metody Newtona musimy mieć odwracalny Jacobian. Zmieniam zatem wektor startowy na [10,0,0].

Wyniki:

```
newton-raphson for system of nonlinear equations
[9.31322575e-09 0.00000000e+00 0.00000000e+00]
```

Jacobian danych funkcji obliczyłem analitycznie, jest on w kodzie. Jak widzimy wektor startowy [0,0,0] był de facto dobrym rozwiązaniem tego układu równań.

## 4. Wnioski

Zarówno metoda Newtona jak i metoda siecznych dobrze radzą sobie z wyznaczaniem miejsc zerowych nieliniowych funkcji, ta pierwsza robi to jednak w mniejszej ilości iteracji. Metodę Newtona da się uogólnić do N-tego wymiaru, natomiast w pewnym momencie możemy otrzymać nieodwracalny Jacobian. Musimy także dobrze dobrać wektor początkowy (initial guess).