

**Wydział:** Fizyki i Informatyki Stosowanej

**Kierunek:** Informatyka Stosowana

**Rok:** 2020/21

**Semestr:** letni

**Typ:** stacjonarne

**Nr albumu:** 401984

**Data:** 12.04.2021



## Sprawozdanie - Laboratorium nr 6

### Wyznaczanie pierwiastków układu równań nieliniowych metodą Newtona

#### Spis treści

1	Wstęp teoretyczny	2
2	Zadanie do wykonania	3
3	Wyniki	4
4	Podsumowanie	5
5	Literatura	6

*opracował:*

*Tomasz Szkaradek*

# 1 Wstęp teoretyczny

Założmy, że mamy do rozwiązania układ równań nieliniowych czyli taki, który przybiera postać.

$$f_j(\xi_1, \xi_2, \dots, \xi_n) = 0, j = (1, 2, 3, \dots, n) \quad (1)$$

zapisujemy w postaci wektorowej:

$$\mathbf{f}(x) = 0$$

$$\mathbf{f} = (f_1, f_2, \dots, f_n)$$

$$\mathbf{x} = (\xi_1, \xi_2, \dots, \xi_n)$$

Dla takiej postaci układu wyprowadza się wzory iteracyjne. Ogólny wzór iteracyjny:

$$\mathbf{F} = (F_1, F_2, \dots, F_n)$$

$$\mathbf{x}_{i+1} = F(\mathbf{x}_i, \mathbf{x}_{i-1}, \dots, \mathbf{x}_{i-n+1})$$

Zakładamy że funkcja wektorowa  $\mathbf{f}$  ma w otoczeniu rozwiązania:

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$$

funkcję odwrotną

$$\mathbf{g} = (g_1, g_2, \dots, g_n)$$

Jeśli punkt

$$\mathbf{y} = (y^{(1)}, y^{(2)}, \dots, y^{(n)})$$

jest odwrotny do punktu  $\mathbf{x}$  (wektora przybliżonych rozwiązań)

Cały problem polega jednak na tym, że nasz układ jest tak duży iż rozwiązanie go "metodą tradycyjną" przy użyciu kartki i długopisu nie wchodzi w grę. Trzeba więc do tego celu skorzystać z dobrodziejstw techniki czyli komputera. Należy jednak przy tym pamiętać, że układ równań nieliniowych charakteryzuje się tym, że może nie mieć żadnego rozwiązania lub też może mieć ich wiele. Nie możemy więc tak po prostu zastosować znanej wszystkim metody Newtona. Będziemy musieli wprowadzić tutaj pewne modyfikacje. Różnica w stosunku do znajdowania pierwiastków równania nieliniowego polega na tym, że w tym przypadku działamy na zmiennych macierzowych, a zamiast pochodnej liczymy jacobian macierzy.

W przypadku równania skalarnego,  $f: \mathbb{R} \rightarrow \mathbb{R}$ , metoda stycznych (zwana też metodą Newtona) rozwiązywania równania  $f(x)=0$  jest zadana wzorem

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (2)$$

Przez analogię, gdy  $F: \mathbb{R}^N \supset D \rightarrow \mathbb{R}^N$  można więc byłoby zdefiniować wielowymiarową metodę Newtona wzorem

$$x_{k+1} = x_k - F'(x_k)^{-1} F(x_k) \quad (3)$$

gdzie  $F'(x_k)$  oznaczałoby macierz pochodnej  $F$  w punkcie  $x_k$ . Jak za chwilę się przekonamy, jest to rzeczywiście bardzo dobry pomysł, a tak określona metoda zachowuje (z zachowaniem właściwych proporcji) wszystkie cechy metody Newtona znane nam z przypadku skalarnego! Dla  $n=2$  otrzymujemy

$$x_{i+1} = x_i - \begin{bmatrix} \frac{\partial f_1}{\partial \xi_1} & \frac{\partial f_1}{\partial \xi_2} \\ \frac{\partial f_2}{\partial \xi_1} & \frac{\partial f_2}{\partial \xi_2} \end{bmatrix}^{-1} \cdot \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}_{x=x_i} \quad (4)$$

Rząd zbieżności metody Newtona wynosi  $p=2$ . Zazwyczaj zbieżność uzyskujemy tylko jeśli startujemy już z dobrym przybliżeniem rozwiązania.

## 2 Zadanie do wykonania

Podczas naszych 6 laboratoriów byliśmy zobowiązani do wyznaczenia wektora rozwiązań zadanego układu równań nieliniowych metodą Newtona:

$$\begin{cases} 2xy^2 - 3x^2y - 2 = 0 \\ x^2y^3 + 2xy - 12 = 0 \end{cases} \quad (5)$$

Do otrzymania wyniku posługujemy się metodą opisaną na wstępie tj. metodą Newtona w dowolnej k-tej iteracji otrzymujemy przybliżony wektor rozwiązań  $[x_k, y_k]$  który jest zależny od poprzedniego przybliżenia:

$$r_k = r_{k-1} + \Delta r \quad (6)$$

gdzie w tym przypadku  $\Delta r$  jest odwrotną macierzą Jacobiego pomnożoną przez wektor kolumnowy ze znakiem minus.

$$\Delta r = - \begin{bmatrix} 2y^2 - 6xy & 4xy - 3x^2 \\ 2xy^3 + 2y & 3x^2y^2 + 2x \end{bmatrix}_{r=r_{k-1}}^{-1} \cdot \begin{bmatrix} 2xy^2 - 3x^2y - 2 \\ x^2y^3 + 2xy - 12 \end{bmatrix} \quad (7)$$

w naszym przypadku rozmiar macierzy Jacobiego jest równy 2 więc możemy ręcznie policzyć odpowiednie pochodne bez wykorzystywania ogólnego algorytmu na obliczenie macierzy Jacobiego. Kolejne wartości wektora kolumnowego to wartości kolejnych 2 równań dla poprzedniego przybliżenia wartości wektora rozwiązań. Następnie wykonujemy ten algorytm dla dwóch punktów początkowych:

- $r_0 = [10, 10]$
- $r_0 = [10, -4]$

natomiast naszym  $\epsilon$  czyli warunkiem zbieżności będzie liczba  $\epsilon = 10^{-6}$  tj.

$$\|\Delta r\| = \|r_k - r_{k-1}\| < 10^{-6} \quad (8)$$

Oczekiwaną wartością wektora rozwiązań w obu przypadkach to  $r=[1,2]$ .

Następnie narysowaliśmy 2 wykresy dla każdego z punktów startowych. Pierwszy z nich to wykres normy długości różnicy wektorów położenia punktów w zależności od numeru iteracji  $k$  (na wykresie  $\Delta x \equiv \|\Delta r\|$ ).

Drugi wykres reprezentuje punkty pośrednie, przez które przechodzi algorytm, zmierzając do rozwiązania układu równań (dla lepszego zilustrowania zbiegania się algorytmu na wykresie umieściliśmy strzałki).

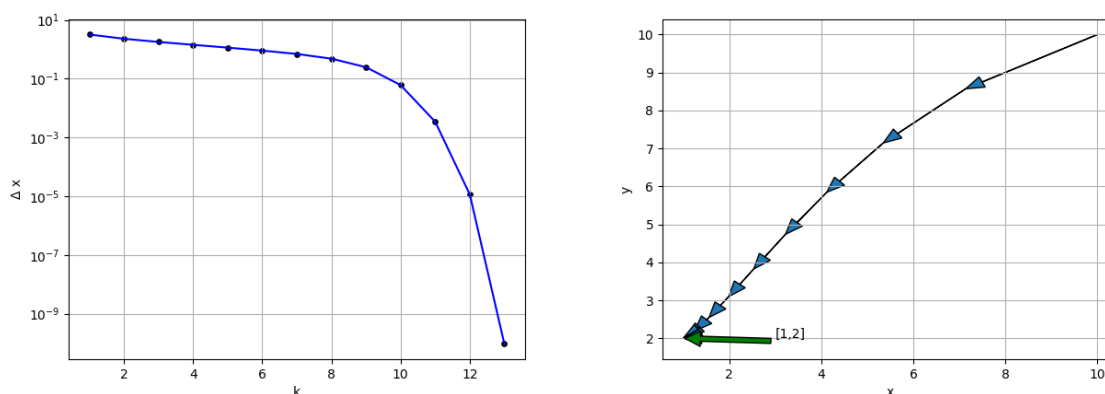
Punkt końcowy także zaznaczymy na wykresie.

### 3 Wyniki

Cały program został napisany w języku Python obliczenia zostały prowadzone na liczbach zmiennoprzecinkowych. Tak jak było to napisane powyżej aby zacząć program musimy obrać punkt od którego zaczynamy iteracje Chcemy sprawdzić różnice w działaniu programu dla dwóch różnych punktach początkowych. Algorytm zapewnia nas że dla zadanego układu znajdziemy dokładnie jeden i ten sam (w obu przypadkach) wektor rozwiązań

- Wektor startowy  $r_0 = [10, 10]$

Dla tego wektora pocztowego od którego zaczynamy iterować otrzymaliśmy wektor rozwiązań równy dokładnie  $r=[1,2]$ . Potrzebował do tego 13 iteracji co możemy zauważyć na wykresach poniżej



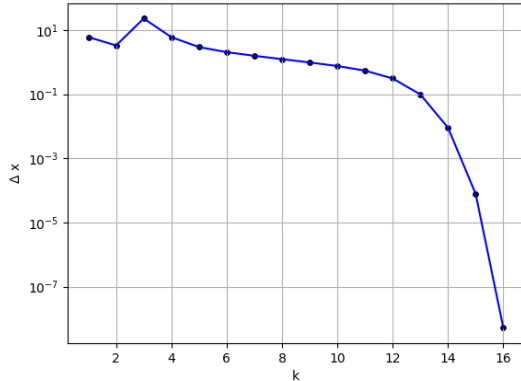
Rysunek 1: Z lewej: wykres normy długości różnicy wektorów położenia punktów w zależności od numeru iteracji  $k$ ;

z prawej: wykres punktów pośrednich, przez które przechodzi algorytm, zmierzając do rozwiązania układu równań.

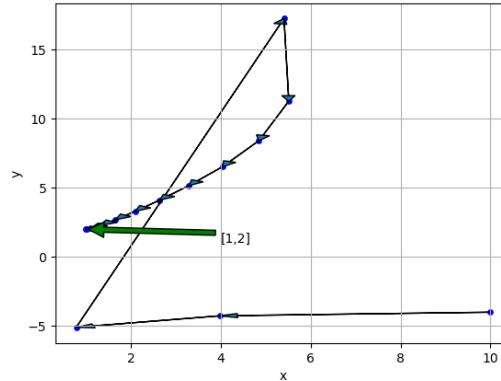
Punkt końcowy zaznaczony jest czerwonym markerem.

- Wektor startowy  $r_0 = [10, -4]$

Dla tego wektora pocztowego od którego zaczynamy iterować otrzymaliśmy wektor rozwiązań równy dokładnie  $r=[1,2]$ . Potrzebował do tego 16 iteracji co możemy zauważyć na wykresach poniżej



(a) wykres normy  $\|r_k - r_{k-1}\|$  od numeru iteracji  $k$



(b) wykres punktów pośrednich

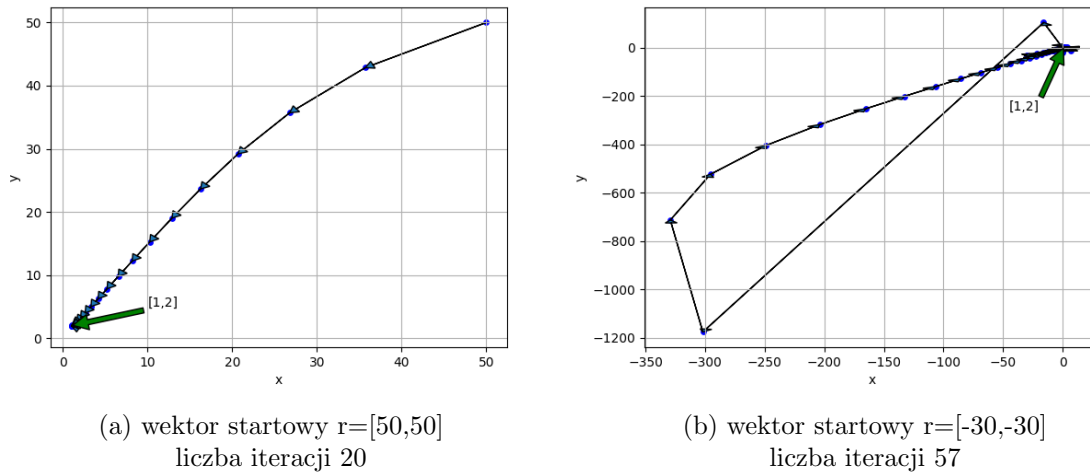
## 4 Podsumowanie

Korzystając z metody Newtona znaleźliśmy wektor rozwiązań układu równań nieliniowych (bądź też dokładne przybliżenie) w niewielkiej ilości iteracji. W celu uzyskania poprawnego wektora rozwiązań algorytm po przekroczeniu przez normę (z wektora różnicy kolejnych przybliżeń) pewnej zadanej z góry liczby zatrzymuje się z błędem.

$$\|\Delta r\| = \|r_k - r_{k-1}\| < 10^{-6} \quad (9)$$

Jednak zastosowany przez nas algorytm w tym zadaniu jest mało ogólny tj. w momencie obliczania macierzy Jacobiego skorzystaliśmy z faktu iż z góry znamy nasz układ równań który dodatkowo był stosunkowo niewielkich rozmiarów  $n=2$  dzięki czemu mogliśmy skorzystać z gotowych przepisów na pochodne cząstkowe w macierzy Jacobiego na etapie pisania programu a nie na etapie wywołania. To uproszczenie mocno wpływa na czas oraz wydajność algorytmu. Porównując wykres 2 oraz 4 możemy wywnioskować iż na ilość iteracji potrzebnych do uzyskania wektora rozwiązań ma dobór punktu startowego dla punktu  $[10,10]$  jest to 13 iteracji natomiast dla  $[10,-4]$  16 iteracji.

Wykresy pokazują nam iż najlepszym punktem startowym jest oczywiście ten najbliższy rozwiązaniu ale też taki który jest w tej samej ćwiartce układu współrzędnych. Widzimy to na podstawie wektora startowego  $r=[10,-4]$  którego kierunek zbieżności nie jest tak jednoznaczny jak dla wektora  $r=[10,10]$  Podsumowując możemy stwierdzić iż jedną z wad metody Newtona przy rozwiązywaniu układu równań nieliniowych jest mocna zależność od punktu początkowego dla którego zaczynamy iterować. Musimy zapewnić by był on jak najbliższy prawidłowemu rozwiązaniu oraz aby odpowiednie współrzędne były tego samego znaku.



Rysunek 3: wykres normy  $\|r_k - r_{k-1}\|$  od numeru iteracji  $k$

Charakterystyczną cechą dla układu równań nieliniowych jest to iż może być sprzeczny przez co nie posiada rozwiązania czy też nieoznaczony czyli posiada więcej niż jedno rozwiązanie. Nie możemy więc tak po prostu zastosować metody Newtona. Jednak metoda Newtona to stabilny algorytm w przeciwieństwie to wielu innych którego, rząd zbieżności jest równy  $p = 2$  więc jest większy od rzędu zbieżności np. metody siecznych ( $p \approx 1.618$ ).

## 5 Literatura

- [1] Tomasz Chwiej, Rozwiązywanie równań nieliniowych i ich układów. Wyznaczanie zer wielomianów.  
*[http : //home.agh.edu.pl/ chwiej/mn/ukl\\_nieliniowe1819.pdf](http://home.agh.edu.pl/~chwiej/mn/ukl_nieliniowe1819.pdf)*
- [2] Wikipedia, Metoda Newtona  
*[https : //pl.wikipedia.org/wiki/Metoda\\_Newtona](https://pl.wikipedia.org/wiki/Metoda_Newtona)*
- [3] Układy równań nieliniowych. Metoda Newtona  
*[http : //mst.mimuw.edu.pl/lecture.php?lecture=partCh9](http://mst.mimuw.edu.pl/lecture.php?lecture=partCh9)*