

Laboratorium z Metod Numerycznych

Sprawozdanie

Autor Łukasz Gut - WFiIS, Informatyka Stosowana, Rok 2.
28 marca 2019

Laboratorium nr 6 - Wyznaczanie pierwiastków równania nieliniowego metodami siecznych i Newtona. Badanie szybkości zbieżności metody.

Cel laboratorium

Celem szóstego laboratorium było zapoznanie się z dwiema podstawowymi metodami wyznaczania pierwiastków równania nieliniowego (metoda Newtona oraz metoda siecznych), zaimplementowanie algorytmu, służącego do rozwiązania zadanego problemu, a następnie zbadanie szybkości zbieżności każdej z tych metod.

Wstęp teoretyczny

Równanie nieliniowe - równanie matematyczne, które można zapisać w postaci:

$$f(x) = 0,$$

przy czym $f(x) = \sum_{i=0}^{\infty} a_i x^i$, dla $i \geq 2$.

W rzeczywistości poszukiwanie rozwiązań tego typu równań jest bardzo potrzebne. Niestety nie istnieją ogólne reguły prowadzące do znalezienia pierwiastków takiego rozwiązania (poza szczególnymi przypadkami), ponieważ równanie w tej formie może zarówno posiadać kilka pierwiastków lub też nie posiadać żadnego. Analityczne rozwiązywanie tego typu zagadnień bardzo często również nie jest najprostsze, dlatego stosuje się różne metody numeryczne, które w pewnym okolicznościach i przy pewnych założeniach są w stanie przybliżyć nam takie rozwiązanie.

Metoda siecznych - numeryczna metoda poszukiwania rozwiązania równania nieliniowego z jedną niewiadomą. W istocie jest to algorytm interpolacji liniowej. Metodę siecznych można zapisać następującym wzorem iteracyjnym:

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

Rząd zbieżności tej metody to:

$$p = \frac{1}{2}(1 + \sqrt{5}) \approx 1.618$$

Prościej mówiąc - algorytm ten wybiera dwa punkty, pomiędzy którymi wiemy, że znajduje się miejsce zerowe, a następnie wykreśla kolejne proste, których kolejne przecięcia się z osią OX to kolejne przybliżenia szukanego pierwiastka.

Metoda Newtona - numeryczna metoda wyznaczania przybliżonej wartości pierwiastka funkcji. Metoda ta również należy do rodzaju algorytmów iteracyjnych. Aby z niej skorzystać spełnione muszą zostać następujące założenia:

- 1) W przedziale $[a,b]$ znajduje się dokładnie jeden pierwiastek.
- 2) Funkcja ma różne znaki na krańcach przedziału, tj. $f(a) * f(b) < 0$. (w istocie jest to twierdzenie Darboux)
- 3) Pierwsza i druga pochodna funkcji mają stały znak w tym przedziale.

Metodę Newtona można zapisać w postaci wzoru iteracyjnego:

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

Rząd zbieżności tej metody to:

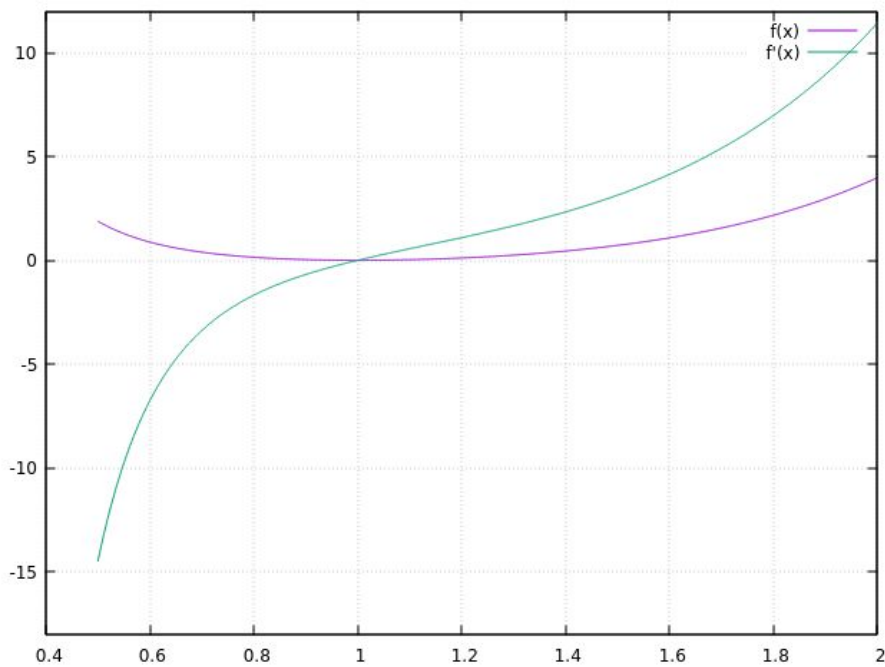
$$p = 2.$$

Metoda ta działa na bardzo podobnej zasadzie, co metoda siecznych, z tym że w przypadku metody Newtona kolejnymi przybliżeniami pierwiastka równania nieliniowego są miejsca zerowe kolejnych stycznych do naszej zadanej funkcji $f(x)$.

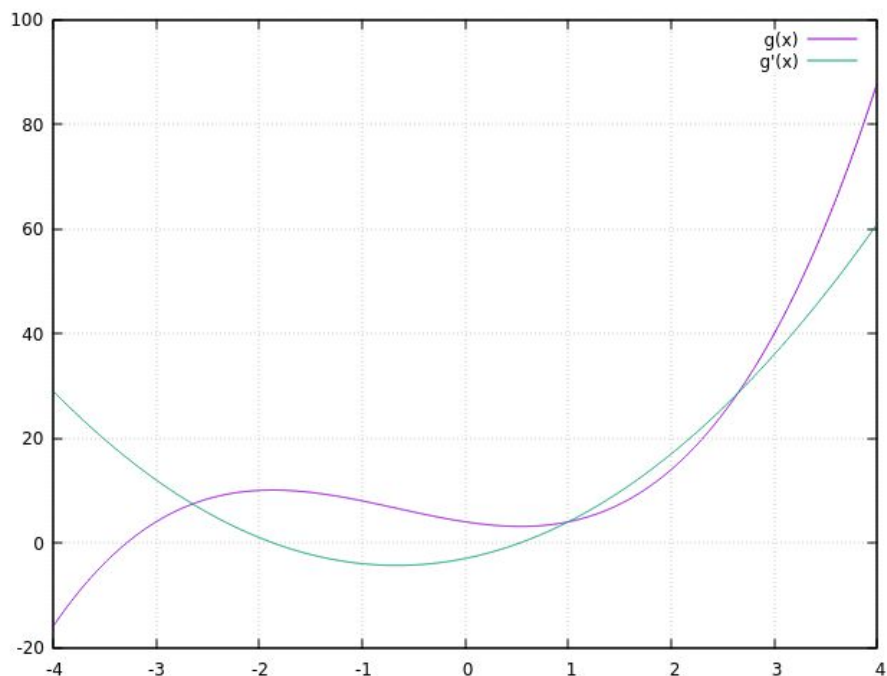
Problem

Problemem, z którym przyszło nam się zmierzyć na szóstym laboratorium był problem rozwiązania dwóch równań nieliniowych:

1) $f(x) = (\ln(x) - x)^6 - 1$



2) $g(x) = x^3 + 2x^2 - 3x + 4$



W przypadku pierwszej funkcji dokładnym rozwiązaniem jest $x_d = 1.0$, zaś w przypadku drugiej $x_d = -3.284277537306950$. Miejscami startowymi dla naszych metod będą następujące punkty na osi OX:

	Metoda siecznych	Metoda Newtona
$f(x)$	$x_0 = 3.0, x_{-1} = 3.01$	$x_0 = 3.0$
$g(x)$	$x_0 = -20.0, x_{-1} = -20.1$	$x_0 = -20.0$

Wyniki

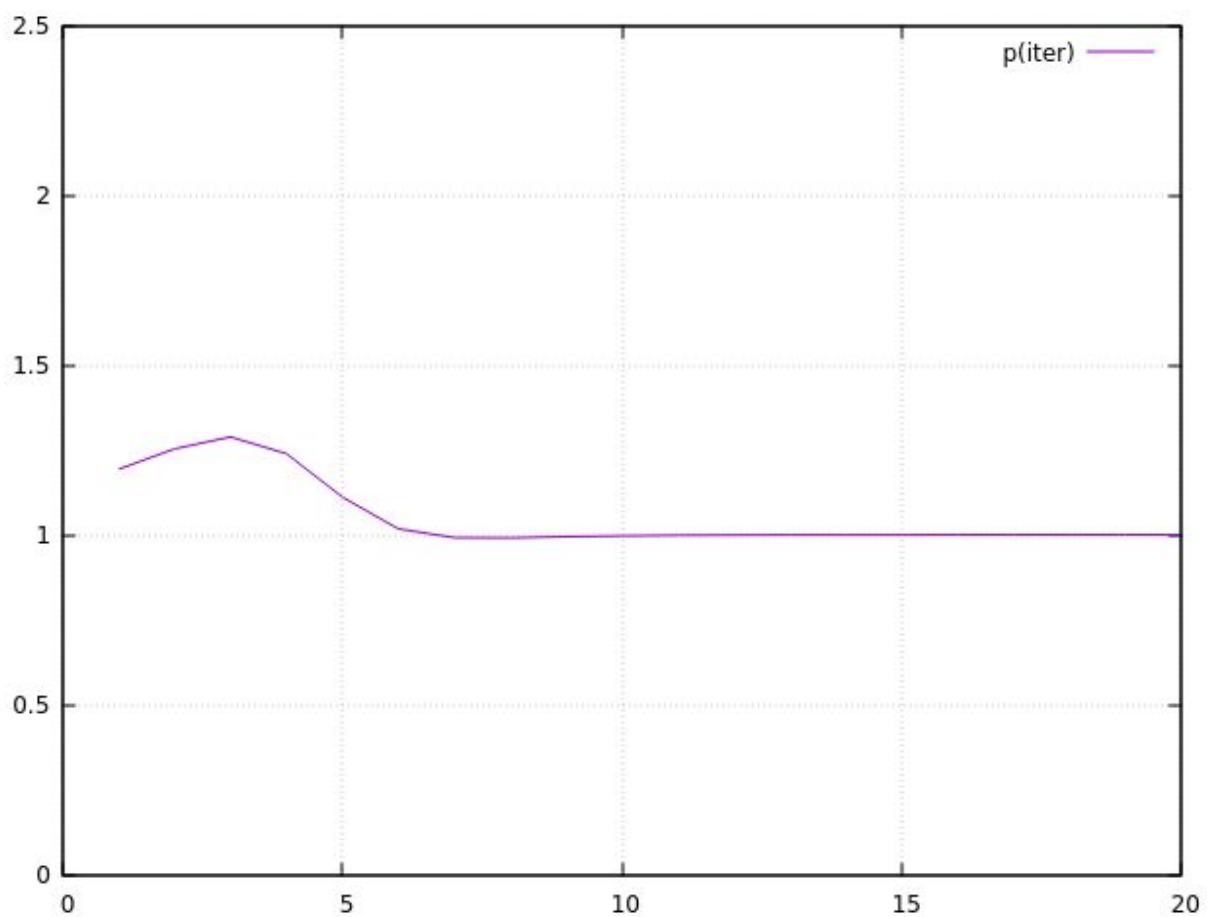
Poniżej przedstawiam wyniki działania programu:

1) Dla funkcji $f(x)$:

Metoda Newtona

k	$f(x_k)$	x_k	ϵ_{k+1}	p
0	16.0707834510016	2.53471280843119	1.53471280843119	1.19336310795865
1	5.55406685494719	2.1188887978375	1.1188887978375	1.25324066596378
2	1.86374362235942	1.75299472925721	0.752994729257214	1.28828040550516
3	0.578803898653489	1.45207882078395	0.452078820783954	1.23866473276679
4	0.159331602728268	1.24029989787832	0.24029989787832	1.11037984992897
5	0.0401210937411285	1.11912367797536	0.119123677975357	1.01757933327116
6	0.00986685640629004	1.05832909029778	0.0583290902977844	0.991506033226283

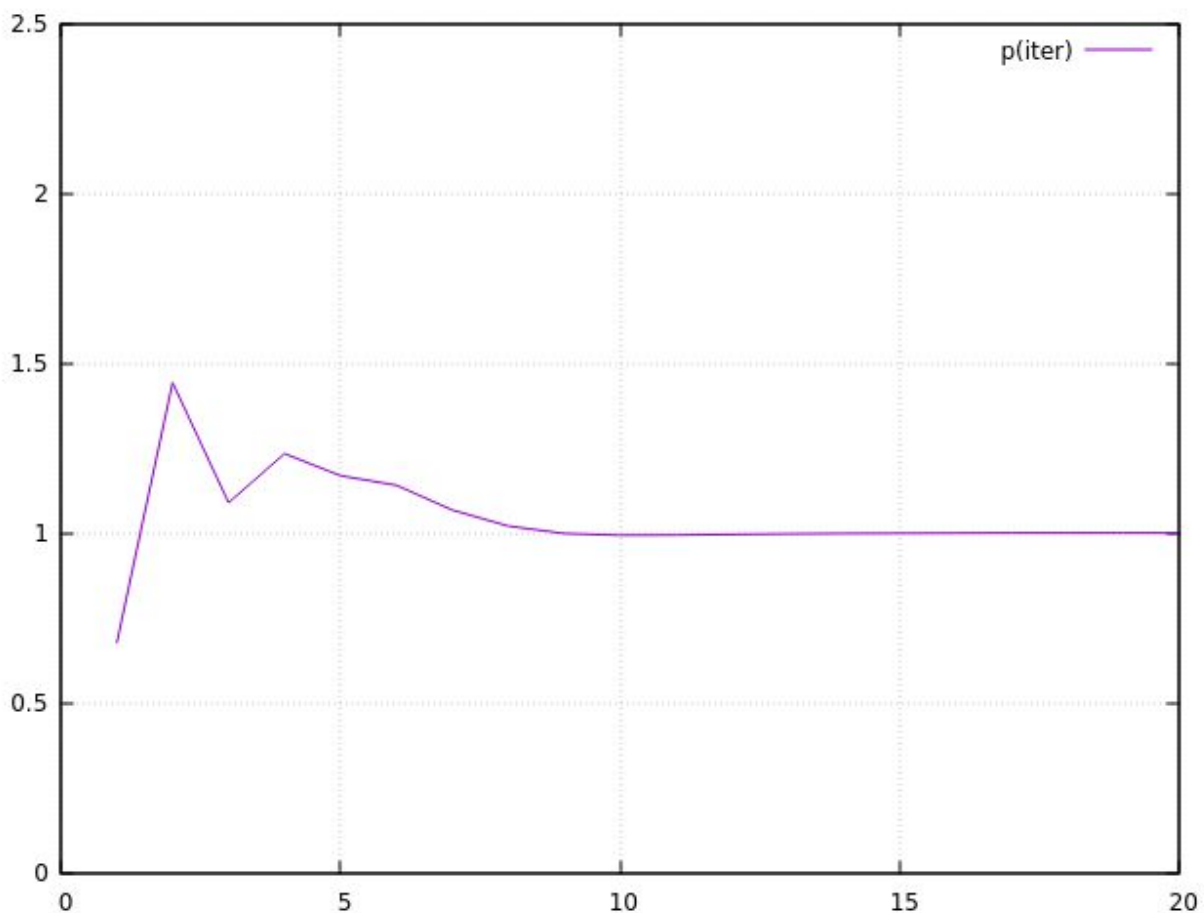
7	0.0024330570730895	1.0287346823868	0.0287346823868002	0.99097217197463
8	0.000603304922160985	1.01424634524328	0.0142463452432757	0.994320919112131
9	0.000150163099294387	1.00709138331188	0.00709138331188019	0.996878792214593
10	3.74553193716043e-05	1.00353756230368	0.00353756230368463	0.998370510464705
11	9.35299216164864e-06	1.00176672674465	0.00176672674465261	0.999168243690196
12	2.33688373985075e-06	1.00088284705542	0.000882847055417857	0.999579896131509
13	5.84049802743181e-07	1.00044129411191	0.000441294111910961	0.999788895108005
14	1.45991022826308e-07	1.00022061466015	0.000220614660148444	0.999894191031412
15	3.64950740738834e-08	1.00011029922544	0.000110299225442034	0.999947008499644
16	9.12343400827353e-09	1.000055147585	5.51475865011586e-05	0.999973589464046
17	2.28081686870496e-09	1.00002757328497	2.7573284967275e-05	0.999987082768898
18	5.70198555038814e-10	1.00001378651179	1.37865117875169e-05	0.999993139668607
19	1.42548639558981e-10	1.00000689322333	6.89322332547704e-06	0.999993575638923



Rys. 1 - Wykres rzędu zbieżności w zależności od iteracji dla metody Newtona

Metoda Siecznych

k	$f(x_k)$	x_k	ε_{k+1}	p
0	16.243526613871	2.539161186718	1.539161186718	0.67510857990286
1	8.73672388111268	2.28971358725792	1.28971358725792	1.44230179715147
2	3.97460044318323	1.99939627416551	0.999396274165512	1.08879934668289
3	1.88923747692646	1.75708938390511	0.757089383905113	1.23319118531318
4	0.843952802938374	1.5375711199843	0.537571119984295	1.16824041446147
5	0.36015868155423	1.36033418277244	0.360334182772441	1.13985081258396
6	0.144071888821004	1.22839082338167	0.228390823381668	1.06677282838165
7	0.055376436648966	1.14042001885274	0.140420018852742	1.02004050893679
8	0.0209341505178076	1.08549601131168	0.0854960113116829	0.997756935305062
9	0.00790074437280075	1.05211299380445	0.052112993804446	0.992910778977912
10	0.00298877558945865	1.03187648120114	0.0318764812011418	0.993229054925017
11	0.0011339350431161	1.01956321180708	0.019563211807077	0.995004460331757
12	0.000431191543194709	1.01203563878265	0.0120356387826515	0.99658155343566
13	0.000164228616168938	1.00741684710507	0.00741684710506774	0.997774865484773
14	6.26159606686993e-05	1.00457548716782	0.00457548716781719	0.998578105962385
15	2.38899933551195e-05	1.00282457847243	0.00282457847243256	0.999105232988801
16	9.1187010291538e-06	1.00174444542477	0.00174444542476837	0.999440293839441
17	3.48150128703928e-06	1.00107765134095	0.00107765134094651	0.999651774478704
18	1.32945233466408e-06	1.00066584325054	0.000665843250544196	0.999783811717035
19	5.07719816411623e-07	1.0004114441921	0.000411444192095933	0.999866053201173



Rys. 2 - Wykres rzędu zbieżności w zależności od iteracji dla metody siecznych

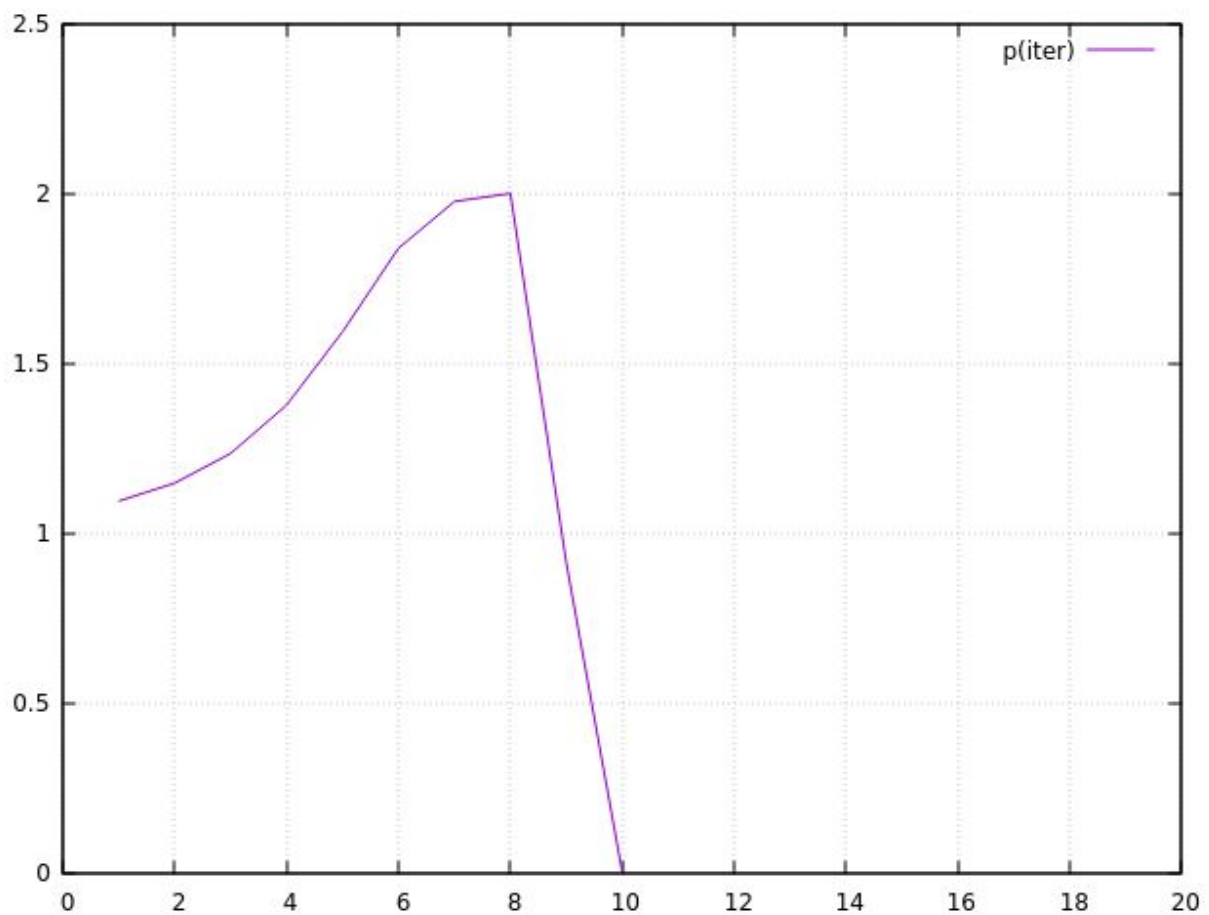
Na podstawie powyższych wyników można zauważyć, że metoda Newtona jest znacznie szybsza w porównaniu do metody siecznych, co potwierdza założenia teoretyczne (rzęd zbieżności jest większy).

2) Dla funkcji $g(x)$:

Metoda Newtona

k	$f(x_k)$	x_k	ε_{k+1}	p
0	-2106.44185211338	-13.6114592658908	10.3271817285838	1.09310321750078
1	-618.256506231519	-9.38479352879782	6.10051599149087	1.14633660414141
2	-178.692284141625	-6.62081911497943	3.33654157767248	1.23347081511055
3	-49.4243963769105	-4.86931966338243	1.58504212607548	1.37592960963354
4	-11.9623496020198	-3.85347594418826	0.569198406881308	1.59208255496759

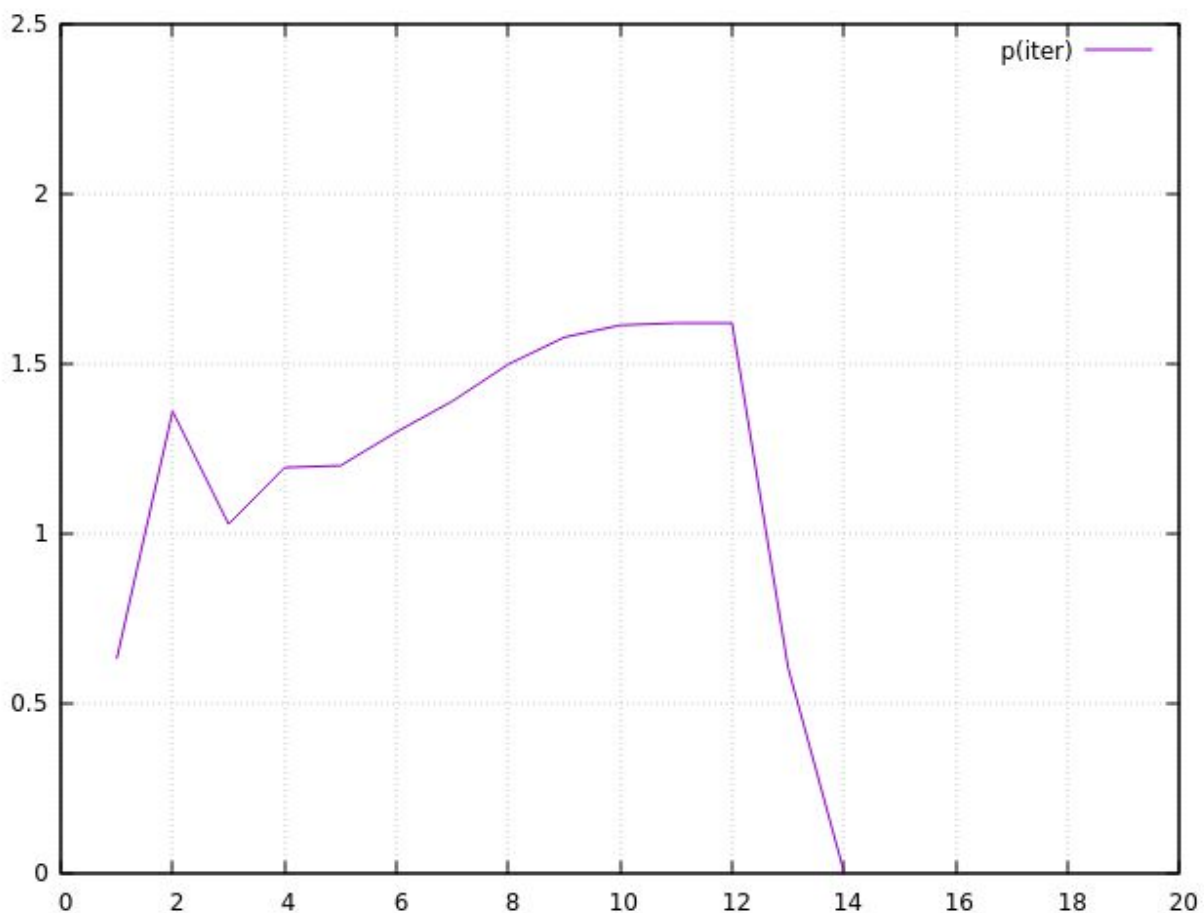
5	-1.90718861152192	-3.39574336911583	0.111465831808883	1.83752855766463
6	-0.0906214805942689	-3.28984872154702	0.00557118424006919	1.97577124069402
7	-0.000242773707709532	-3.28429250260471	1.49652977565751e-05	1.99932763857961
8	-1.75870518148713e-09	-3.28427753741536	1.08415054711486e-10	0.912189343856486
9	0	-3.28427753730695	2.22044604925031e-15	0
10	0	-3.28427753730695	2.22044604925031e-15	-nan
11	0	-3.28427753730695	2.22044604925031e-15	-nan
12	0	-3.28427753730695	2.22044604925031e-15	-nan
13	0	-3.28427753730695	2.22044604925031e-15	-nan
14	0	-3.28427753730695	2.22044604925031e-15	-nan
15	0	-3.28427753730695	2.22044604925031e-15	-nan
16	0	-3.28427753730695	2.22044604925031e-15	-nan
17	0	-3.28427753730695	2.22044604925031e-15	-nan
18	0	-3.28427753730695	2.22044604925031e-15	-nan
19	0	-3.28427753730695	2.22044604925031e-15	-nan



Rys. 3 - Wykres rzędu zbieżności w zależności od iteracji dla metody Newtona

Metoda Siecznych

k	$f(x_k)$	x_k	ϵ_{k+1}	p
0	-2122.95923856256	-13.644516881752	10.360239344445	0.628835360348858
1	-1037.23212887045	-10.9530503161492	7.66877277884222	1.35763438784097
2	-419.205315564356	-8.38180053275238	5.09752299544543	1.02537229732745
3	-180.422816327562	-6.6377312567325	3.35345371942555	1.19202839396237
4	-73.9992130606969	-5.31992156944118	2.03564403213423	1.1981229327454
5	-29.3995543744629	-4.40361281057695	1.11933527327	1.29700748736471
6	-10.5817823405733	-3.79959318411419	0.515315646807241	1.38742700506673
7	-3.09730231728959	-3.45993532211265	0.175657784805701	1.49633372272304
8	-0.579074266939768	-3.31937471138349	0.0350971740765358	1.5756980195935
9	-0.0450746901442738	-3.28705236765958	0.00277483035262627	1.61124554415726
10	-0.000754586837253157	-3.28432405158705	4.65142801027163e-05	1.61759754065846
11	-1.01253992212946e-06	-3.28427759972339	6.24164431251018e-08	1.6177941367567
12	-2.27977636768628e-11	-3.28427753730836	1.4077627952247e-12	0.603018772083633
13	0	-3.28427753730695	2.22044604925031e-15	0
14	0	-3.28427753730695	2.22044604925031e-15	nan
15	-nan	-nan	nan	nan
16	-nan	-nan	nan	nan
17	-nan	-nan	nan	nan
18	-nan	-nan	nan	nan
19	-nan	-nan	nan	nan



Rys. 4 - Wykres rzędu zbieżności w zależności od iteracji dla metody

W przypadku szukania miejsca zerowego funkcji g możemy zauważyć, że 20 iteracji to za dużo (i to w przypadku obu metod). Oprócz tego warto również zauważyć, że obie metody osiągnęły swoje teoretyczne rzędy zbieżności (zaznaczone kolorem jasnoczerwonym).

Wnioski

W obu badanych przypadkach metoda Newtona okazała się szybsza od metody siecznych, co zgadza się z założeniami teoretycznymi. Co ciekawe jest to również znacznie optymalniejsza metoda w sensie ilości potrzebnych operacji, które musimy wykonać w trakcie jednej iteracji. Na przykładzie drugiej metody widać również, że zdecydowanie lepszym warunkiem zatrzymania pętli byłoby np. sprawdzenie czy kolejne wartości funkcji nie różnią się od siebie bardziej niż jakiegoś zadane ε . Dzięki temu moglibyśmy uniknąć błędów w postaci symbolu nan (not a number).