ITERATION METHODS FOR SOLVING NONLINEAR EQUATIONS, USING MATLAB AND PYTHON

Tonya P. Mateva
College of Dobrich, Shumen University, Shumen, Bulgaria tonqmatewa@gmail.com

Abstract: In this paper, we compare some iterative methods for finding a simple root α of nonlinear equation $f(x) = 0$ in R. All calculations were made using the paid computer program Matlab 7.6.0 (R2008a) and the free distribution Python programming language. On the one hand, we compare the efficiency of the methods and on the other hand the efficiency of the two computer programs. When the equation is difficult to solve or unsolvable by using analytical methods it is solved with the help of numerical methods. Solving is in two stages - first locate the roots, then find the roots. We are looking for a simple root α of nonlinear equation f (x) = 0, where $f: I \subset R \to R$ for open interval I. In order to improve of convergence of lines and the number of iterations, many authors have modified Newton's formula and developed various iterative scheme using a variety of techniques. We stopped the following iterative methods - Weerakoon and Fernando's method. Homeier's method and the Kou's method . All three methods are a modification of the Newton's method, so to choose an initial approximation, we will impose Newton type: Interval $[a, b]$, in which the root is located; 2. The initial approximation x_0, is this end of the interval $[a, b]$, for which $f(x_0).f''(x_0) > 0$. For experiments, we use the Matlab program, a scientific and engineering computing product program, and Python, an interactive, object-oriented programming language. We will compare the accuracy of the results with the two software programs.
Keywords: Matlab, Python, iterative methods, nonlinear equation

ИТЕРАЦИОННИ МЕТОДИ ЗА РЕШАВАНЕ НА НЕЛИНЕЙНИ УРАВНЕНИЯ, С ПОМОЩТА НА MATLAB И PYTHON

Тоня П. Матева
Колеж - Добрич, Шуменски университет „Епископ Константин Преславски“, България
tonqmatewa@gmail.com

Резюме: В статията ще разгледаме итерационни методи за решаване на нелинейни уравнения. Ще направим експерименти с помощта на платената компютърна програма MATLAB и безплатно разпространеният език за програмиране Python. От една страна ще сравним ефективността на методите, а от друга ефективността на двете компютърни програми. Когато уравнението е трудно решимо или нерешимо с помощта на аналитични методи, се прибягва до решаването им с помощта на числени методи. Решаването се осъществява на два етапа – първо локализиране на корените, след което уточняване на корените. Търсим прост корен α на нелинейно уравнение $f(x) = 0$, където $f: I \subset R \to R$ за отворен интервал I . За подобряване реда на сходимост и броят итерации, много автори са модифицирали формулата на Нютон и разработили различни итерационни схеми с помощта на различни техники. Спрели сме се на следните итерационни метода - методът на Weerakoon and Fernando [1], методът на Homeier [2, 3] и методът на Kou [4]. И трите метода са модификация на Нютъновият метод, затова за избор на първоначално приближение, ще наложим условия Нютънов тип: 1. Интервал $[a, b]$, в който се намира коренът; 2. Първоначалното приближение x_0 , е този край на интервала $[a, b]$, за който $f(x_0).f''(x_0) > 0$.
За експериментите ще използваме програма Matlab - диалогов програмен продукт, предназначен за научни и инженерни изчисления и Python - интерактивен, обектно-ориентиран език за програмиране. Ще сравним точността на резултатите с двете софтуерни програми.
Ключови думи: Matlab, Python, Итеративни методи, Нелинейни уравнения

1. ВЪВЕДЕНИЕ

Python е един от най-популярните езици за програмиране с отворен код. Той е мощен, гъвкав и лесен за прилагане. Основното предимство на езика е, че може да се изпълни задача с написването на по-малко код, в сравнение с други програмни езици. Той разполага със структури от данни от високо ниво и прост, но ефективен подход към обектно-ориентираното програмиране. Едно от основните му предимства, е достъпността-интерпретаторът на Python и обширната му стандартна библиотека могат да се разпространяват безплатно [5].
Matlab е широко разпространена програмна система, популярна сред инженери, изследователи, научни работници. Тя предлага големи възможности за извършване на аналитични преобразувания, изчисления и

висококачествена визуализация на получените резултати. Има вграден програмен език от високо ниво, което я прави гъвкава и адаптивна, както и приспособима към изискванията на всеки потребител [6]. Matlab е платена програма.

Ще направим експерименти със следните итерационни схеми:

1.1 Метод на Newton:

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

1.2. Метод на Weerakoon and Fernando [1]:

$$x_{n+1} = x_n - \frac{2.f(x_n)}{f'(x_n) + f'\left(x_n - \frac{f(x_n)}{f'(x_n)}\right)} \quad (2)$$

1.3. Метод на Homeier [2,3]:

$$x_{n+1} = x_n - \frac{f(x_n)}{2}.\left(\frac{1}{f'(x_n)} + \frac{1}{f'\left(x_n - \frac{f(x_n)}{f'(x_n)}\right)}\right) \quad (3)$$

1.4. Метод на Kou [4]

$$x_{n+1} = y_n - \frac{f(y_n)}{\frac{f'(x^*_{n+1} + y_n)}{2}}$$

където

$$x^*_{n+1} = y_n - \frac{f(y_n)}{f'(x_n)}$$
$$y_n = x_n + \frac{f(x_n)}{f'(x_n)} \quad (4)$$

За всяка една итерационна схема ще напишем програмен код на Matlab и Python.
Както вече споменахме, тези схеми са модификация на Нютъновата итерационна формула, затова за избор на начално приближение, ще следваме следните условия:
1. Ще изследваме функциите в интервал $[a, b]$, за който $f(a).f(b) < 0$, което гарантира наличието на корен в съответният интервал.
2. Ще изискваме в съответният интервал, първата и втора производна да са непрекъснати за $\forall x \in [a, b]$: $f'(x) \neq 0$ и $f''(x) \neq 0$
3. За първоначално приближение ще използваме този край на интервала $x_0 = a$ или $x_0 = b$, за който $f(x_0).f''(x_0) > 0$.

2. ЧИСЛЕНИ ЕКСПЕРИМЕНТИ

Изчисления са направени с помощта на компютърна програма Matlab 7.6.0(R2008a) и програмният език Python. За стоп критерии използваме 2 условия - абсолютната стойност на функцията от намереният приближен корен трябва да е по-малка от зададената точност $|f(x_n)| < \varepsilon$ и абсолютната стойност на разликата на последното и предпоследното приближение $|x_n - x_{n-1}| < \varepsilon$ и, когато се достигне стоп критерият, x_{n+1} приема стойността на корена α. За стоп критерии вземаме $\varepsilon = 10e - 15$.
В таблици от 1 до 5 са представени експериментите, направени с описаните по-горе методи. Никой от избраните методи не изисква изчисляване на втора производна.

Пример 1. Разглеждаме нелинейната функция $f_1(x) = (x-1)^3 - 1$.

Таблица 1

$f_1(x) = (x-1)^3 - 1$		$x_n = 2.00000e+00$				
		it	$\|f(x_n)\|$	δ	ρ	Чрез графичният метод, установяваме, че функцията има реален корен в интервала $(1, +\infty)$. Първоначално приближение намираме надясно по оста Ox , след корена, където функцията и втората й производна имат един и същ знак. В тази таблица са представени експерименти с три първоначални приближения, отговарящи на условията описани по-горе. С тях, методът на Homeier се отличава с най-малък брой.
Newton	$x_0 = 3.4$	7	0.00e+00	2.88e-11	2.00e+00	
WF		5	0.00e+00	7.24e-14	2.99e+00	
Homeier		4	0.00e+00	6.51e-09	3.05e+00	
Kou		5	0.00e+00	3.20e-13	2.99e+00	
Python		7	0.00e+00	4.58e-12	2.00e+00	
		5	0.00e+00	7.24e-14	2.99e+00	
		4	0.00e+00	6.51e-09	3.06e+00	
		5	0.00e+00	3.20e-13	2.99e+00	
Newton	$x_0 = 3.8$	7	0.00e+00	2.38e-09	2.00e+00	
WF		5	0.00e+00	1.26e-10	2.97e+00	
Homeier		4	0.00e+00	3.50e-07	3.05e+00	
Kou		5	0.00e+00	4.14e-10	2.96e+00	
Python		7	0.00e+00	1.38e-09	2.00e+00	
		5	0.00e+00	1.26e-10	2.97e+00	
		4	0.00e+00	3.50e-07	3.08e+00	
		5	0.00e+00	4.14e-10	2.96e+00	
Newton	x_0 =6.7	9	0.00e+00	8.25e-11	2.00e+00	
WF		6	0.00e+00	5.12e-09	2.94e+00	
Homeier		5	0.00e+00	8.63e-08	3.05e+00	
Kou		6	0.00e+00	1.90e-08	2.92e+00	
Python		9	0.00e+00	8.25e-11	2.00e+00	
		6	0.00e+00	5.12e-09	2.95e+00	
		5	0.00e+00	8.63e-08	3.07e+00	
		6	0.00e+00	1.91e-08	2.93e+00	

Пример 2. Разглеждаме нелинейната функция $f_2(x) = x^2 - e^{-x} - 3.x + 2$.

Таблица 2

$f_2(x) = x^2 - e^{-x} - 3.x + 2$		$x_n = 2.109356e+00$				
		it	$\|f(x_n)\|$	δ	ρ	Тази таблица представя експериментите, направени с функция f_2. Изследваме функцията в интервала $(1.5, +\infty)$, където коренът е $x_n = 2.109356e+00$. За първоначално приближение избираме точки в дясно от корена, където функцията и втората и производна са с еднакъв знак. Методът на Homeier прави най-малък брой итерации, независимо колко първоначалното приближение се отдалечава от корена. С първоначално приближение $x_0 = 3$, методът на Kou, представен чрез програмата Python, прави една итерация повече в сравнение с Matlab.
Newton	$x_0 = 3$	6	8.88e-16	9.73e-14	2.00e+00	
WF		4	0.00e+00	1.27e-11	2.97e+00	
Homeier		3	8.88e-16	6.77e-07	3.31e+00	
Kou		4	0.00e+00	1.53e-09	2.93e+00	
Python		6	0.00e+00	9.73e-14	2.00e+00	
		4	0.00e+00	1.27e-11	2.97e+00	
		3	8.88e-16	6.78e-07	3.35e+00	
		5	2.22e-16	2.80e-06	2.93e+00	
Newton	$x_0 = 3.5$	6	1.78e-15	3.01e-10	2.00e+00	
WF		4	8.88e-16	1.09e-08	2.91e+00	
Homeier		3	0.00e+00	2.71e-05	3.04e+00	
Kou		4	0.00e+00	5.66e-07	2.81e+00	
Python		6	0.00e-00	3.01e-10	2.00e+00	
		4	8.88e-16	1.09e-08	2.92e+00	
		3	0.00e+00	2.71e-05	3.13e+00	
		4	0.00e+00	5.66e-07	2.84e+00	
Newton	$x_0 = 6$	7	8.88e-16	5.11e-09	2.00e+00	
WF		5	0.00e+00	3.44e-11	3.43e+00	
Homeier		4	0.00e+00	1.77e-08	3.40e+00	
Kou		5	0.00e+00	3.87e-08	2.88e+00	

Python	7	0.00e+00	5.11e-09	2.00e+00	
	5	0.00e+00	3.43e-11	2.97e+00	
	4	0.00e+00	1.77e-08	3.42e+00	
	5	0.00e+00	3.87e-08	2.89e+00	

Пример 3. Разглеждаме нелинейната функция $f_3(x) = e^x - 4.x^2$.

Таблица 3

$f_3(x) = e^x - 4.x^2$ $x_n = -4.077767094044\text{e}-01$						
		it	$\lvert f(x_n)\rvert$	δ	ρ	В таблицата са представени резултатите, при изследване на функция f_3. В интервала $(-\infty, 0)$ се намира един от корените на функцията - $x_n = -4.077767094044\text{e}-01$. Първоначалното приближение избираме в ляво от корена, където $f(x_0).f''(x_0) > 0$. От резултатите се вижда, че методът на Homeier прави най-малък брой итерации, следван от Kou и Weerakoon and Fernando. Отдалечавайки се от корена, наляво по числовата ос, броят итерации расте, но се запазва същата тенденция в резултатите. И с двете компютърни програми, итерационни методи достигат корена с един и същ брой итерации, с малки разлики в стойностите на δ – разликата между последните две приближения и ρ- реда на сходимост.
Newton	$x_0 = -1$	6	1.11e-16	6.44e-15	2.00e+00	
WF		4	1.11e-16	1.26e-12	2.98e+00	
Homeier		3	1.11e-16	1.83e-07	3.34e+00	
Kou		4	1.11e-16	1.80e-10	2.95e+00	
Python		6	1.11e-16	6.44e-15	2.00e+00	
		4	1.11e-16	1.26e-12	2.98e+00	
		3	1.11e-16	1.83e-07	3.38e+00	
		4	1.11e-16	1.80e-10	2.95e+00	
Newton	$x_0 = -2.5$	7	1.11e-16	7.26e-12	2.00e+00	
WF		5	1.11e-16	1.03e-14	2.98e+00	
Homeier		4	1.11e-16	1.08e-10	3.37e+00	
Kou		5	1.11e-16	5.08e-11	2.96e+00	
Python		7	1.11e-16	7.26e-12	2.00e+00	
		5	1.11e-16	1.03e-14	2.98e+00	
		4	2.22e-16	1.08e-10	3.37e+00	
		5	1.11e-16	5.08e-11	2.96e+00	
Newton	$x_0 = -3$	7	1.11e-16	5.30e-10	2.00e+00	
WF		5	1.11e-16	2.25e-12	2.98e+00	
Homeier		4	1.11e-16	2.93e-09	3.40e+00	
Kou		5	1.11e-16	3.87e-09	2.91e+00	
Python		7	1.11e-16	5.30e-10	2.00e+00	
		5	2.22e-16	2.25e-12	2.98e+00	
		4	1.11e-16	2.93e-09	3.40e+00	
		5	1.11e-16	3.87e-09	2.91e+00	

Пример 4. Разглеждаме нелинейната функция $f_4(x) = 0.5.e^x - 5x + 2$.

Таблица 4

$f_4(x) = 0.5.e^x - 5x + 2$ $x_n = 3.401795803e+00$						
		it	$\lvert f(x_n)\rvert$	δ	ρ	В таблицата са представени резултатите от експеримента с функция f_4. Търсим корена $x_n = 3.401795803e+00$ и според условията за избор на начално приближение, ще търсим такова в интервала $(2.5, +\infty)$. Тъй като в ляво от корена функцията и втората и производна имат различни знаци, което противоречи на наложеното от нас условие
Newton	$x_0 = 5$	7	0.00e+00	2.97e-11	2.00e+00	
WF		5	0.00e+00	2.91e-12	2.99e+00	
Homeier		4	3.55e-15	8.92e-08	3.05e+00	
Kou		5	0.00e+00	7.11e-15	3.00e+00	
Python		7	0.00e+00	2.97e-11	2.00e+00	
		5	0.00e+00	2.91e-12	2.99e+00	
		4	3.55e-15	8.92e-08	3.07e+00	
		5	0.00e+00	7.11e-15	3.00e+00	
Newton	$x_0 = 7$	9	0.00e+00	6.84e-10	2.00e+00	

WF		6	0.00e+00	2.23e-07	2.92e+00	$f(x_0).f''(x_0) > 0$, то избираме точки $x_0 = 5$ и $x_0 = 7$ за първоначални приближения. И за тази функция методът на Homeier прави една итерация по-малко от повечето методи. С най-много итерации е методът на Нютон. И в този пример, двете компютърни програми представят еднакви резултати като брой итерации, с малки разминавания в стойността δ и ρ, които са незначителни и не оказват влияние на цялото изследване.
Homeier		5	3.55e-15	2.03e-06	3.05e+00	
Kou		6	0.00e+00	5.34e-11	2.97e+00	
Python		9	0.00e+00	6.84e-10	2.00e+00	
		6	0.00e+00	2.23e-07	2.94e+00	
		5	3.55e-15	2.03e-06	3.09e+00	
		6	0.00e+00	5.34e-11	2.97e+00	

Пример 5. Разглеждаме нелинейната функция $f_5(x) = x.\exp(x) - 1.$

Таблица 5

$f_5(x) = x.\exp(x) - 1$		$x_n = 567.143290e-003$				
		it	$\lvert f(x_n)\rvert$	δ	ρ	Таблицата представя резултатите при изследване на функция f_5, с реален корен $x_n = 567.143290e-003$, с три първоначални приближения, избрани в дясно от корена, за да отговарят на условието $f(x_0).f''(x_0) > 0$. Ще изследваме функцията в интервала $(-1, +\infty)$. И тук с най-малък брой итерации е методът на Homeier, значително повече от метода на Нютон. Отдалечавайки се от началното приближение, броят итерации за всички методи расте, но методът на Homeier остава стабилен с най-малък брой итерации. И с двете програми Matlab и Python, резултатите са едни и същи.
Newton	$x_0 = 1$	5	0.00e+00	6.12e-09	2.00e+00	
WF		4	2.22e-16	1.60e-13	2.99e+00	
Homeier		3	0.00e+00	1.01e-06	3.03e+00	
Kou		3	6.00e-15	1.56e-05	2.84e+00	
Python		5	2.22e-16	6.12e-09	2.00e+00	
		4	2.22e-16	1.60e-13	2.99e+00	
		3	0.00e+00	1.01e-06	3.07e+00	
		3	6.00e-15	1.56e-05	2.92e+00	
Newton	$x_0 = 2$	7	2.22e-16	2.01e-10	2.00e+00	
WF		5	2.22e-16	4.11e-11	2.98e+00	
Homeier		4	0.00e+00	4.24e-07	3.03e+00	
Kou		5	0.00e+00	1.63e-14	2.99e+00	
Python		7	2.22e-16	2.01-10	2.00e+00	
		5	0.00e+00	4.11e-11	2.98e+00	
		4	0.00e+00	4.24e-07	3.06e+00	
		5	0.00e+00	1.63e-14	2.99e+00	
Newton	$x_0 = 5$	11	0.00e+00	2.22e-16	2.00e+00	
WF		7	4.44e-16	5.67e-01	2.87e+00	
Homeier		6	0.00e+00	1.28e-06	3.03e+00	
Kou		7	2.22e-16	5.73e-11	2.98e+00	
Python		11	2.22e-16	1.69e-11	2.00e+00	
		7	4.44e-16	5.91e-06	2.93e+00	
		6	0.00e+00	1.28e-06	3.07e+00	
		7	0.00e+00	5.73e-11	2.98e+00	

3. ЗАКЛЮЧЕНИЕ

Направихме експерименти с четири итерационни формули за намиране прост корен на нелинейно уравнение. Методите, които сравнихме не изискват изчисляване на втора производна и три от тях са с ред на сходимост 3, освен метода на Newton, с ред на сходимост 2. Предвид направените експерименти по-горе, можем да кажем, че при сравнение на избраните итерационни методи, методът на Homeier се отличава като най-стабилен – прави най-малко итерации за намиране корена на функция, независимо колко, първоначалното приближение се отдалечава от корена. След него може да наредим методът на Weerakoon and Fernando и методът на Kou, които правят по-малко итерации от метода на Newton. От друга страна, при

сравнение на двата програмни продукта, с които са направени изчисленията, можем да кажем, че резултатите са идентични, с малки разлики при изчисление на реда на сходимост и разликата в последните две приближения. На общият фон на изследването тези разлики са незначителни.

ЛИТЕРАТУРА

[1] S. Weerakoon, G.I. Fernando, A variant of Newton's method with accelerated third-order convergence, Appl. Math. Lett. 17 (2000) 87-93.
[2] H.H.H. Homeier, On Newton – type methods with cubic convergence, J. Comput. Appl. Math. 176 (2005) 425-432.
[3] H.H.H. Homeier, A modified Newton's method with cubic convergence: the multivariate case, J. Comput. Applied Mathematics. 169 (2004) 161-169.
[4] J. Kou, Y. Li, X. Wang, A modifcation of Newton's method with third-order convergence, Appl. Math. Comput. 181 (2) 1106-1111
[5] Rosum, G., Ръководство по Python, изд. 2.0.1 http://www.daskalo.com/pgitv/files/2016/01/tut-2.0.pdf
[6] Й. Тончев, Matlab 7, Техника, София, 2009