

Глава 5

Числено решаване на уравнения

Накрая ще разгледаме няколко метода за приближеното решаване на уравнения. Както знаем, в общия случай е трудно или дори невъзможно да намерим точно решение за дадено нелинейно уравнение. Ето защо в тези случаи трябва да се обърнем към числени методи за намиране на приближено решение (с произволна точност). Тук ще разгледаме само алгоритмичната страна на въпроса, като ще предполагаме, че задачите, които ще решаваме, могат да бъдат решени със съответните числени методи. За повече информация – виж лекциите и учебника.

И така, разглеждаме следната задача – търсим приближено решение на уравнението

$$f(x) = 0$$

с грешка, не по-голяма от ϵ . Първият метод, който ще разгледаме, е т.нар. **метод на разполовяването**. Идеята е следната:

1. Нека сме ограничили корена на уравнението в даден интервал $[a, b]$ и за него имаме $f(a) \cdot f(b) < 0$.
2. Оценяваме стойността на функцията $f\left(\frac{a+b}{2}\right)$.
 - Ако $f(a) \cdot f\left(\frac{a+b}{2}\right) < 0$, това означава че функцията има различни знаци в точките a и $\frac{a+b}{2}$ и коренът се намира в интервала, определен от тези две точки. Вземаме $b = \frac{a+b}{2}$ и търсим корена в така получения (два пъти по-малък) интервал $[a, b]$.
 - В противен случай, вземаме $a = \frac{a+b}{2}$.
3. Повтаряме тези стъпки, докато интервалът, в който е заключен коренът стане с дължина, по-малка от ϵ .

Да демонстрираме този алгоритъм, като решим следната

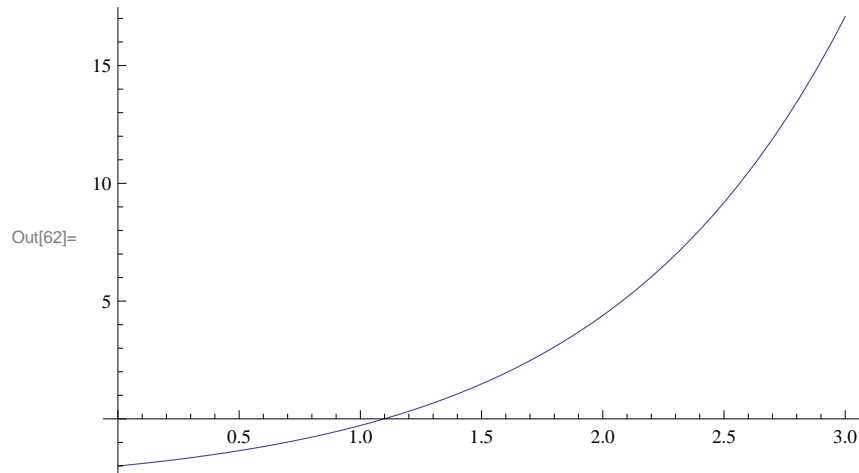
Задача 1. Да се намери приближено решение на уравнението

$$e^x - 3 = 0$$

в интервала $[0, 3]$ с грешка, ненадминаваща $\epsilon = 0.000001$.

Решение. Първо, да изчертаем графиката на функцията $f(x) = e^x - 3$, за да видим какви корени има тя в искания интервал.

```
In[62]:= Plot[f[x], {x, 0, 3}]
```



Оказва се, че уравнението има един корен и този корен е ограничен например в интервала $[1, 3]$. Нека сега реализираме описания по-горе алгоритъм в Mathematica, като за начални стойности на a и b вземем съответно 1 и 3:

```
In[52]:= f[x_] = E^x - 3;
ε = 0.000001;
a = 1;
b = 3;
iter = 0;
While[b - a > ε,
  If[f[a] f[(a + b)/2] < 0, b = (a + b)/2, a = (a + b)/2];
  iter++;
]
Print["x≈", b // N]
Print["iterations≈", iter]
x≈1.09861
iterations≈21
```

□

Вторият метод, който ще разгледаме е т.нар. метод на Нютон, който е може би най-широко използваният за приближено решаване на уравнения, тъй като (в случаите, когато е сходящ към точното решение) дава резултат с исканата точност след сравнително малък брой итерации.

1. Нека имаме едно първоначално приближение x_0 (например избрано от графиката на функцията). Да построим допирателната към графиката на функцията $f(x)$ в точката $(x_0, f(x_0))$. Имаме

$$f'(x_0) = \operatorname{tg} \varphi = \frac{f(x_0) - 0}{x_0 - x_1}$$

Тогава

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

2. Аналогично намираме следващите приближения по формулата

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad i = 0, 1, \dots$$

3. Правим това, докато разликата в две съседни приближения стане по-малка от ϵ .

```
In[22]:= f[x_] := E^x - 3;
epsilon = 0.000001;
x0 = 3;
x1 = x0 - f[x0]/f'[x0];
i = 1;
While[Abs[xi - xi-1] > epsilon,
  xi+1 = xi - f[xi]/f'[xi]; i++;]
Print["x≈", xi // N];
Print["Iterations:", i];
```

x≈1.09861
Iterations:7

Библиография

- [1] Боянов, Б.: Лекции по числени методи. Дарба, 2008
- [2] Сборник по числени методи – <http://www.fmi.uni-sofia.bg/econtent/nummeth>
- [3] Сендов, Бл., Попов, В.: Числени методи. Първа част. Университетско издателство „Св.Климент Охридски”, 1996
- [4] Kiusalaas, J.: Numerical Methods in Engineering. Cambridge University Press, 2010
- [5] Chapra, S.: Applied Numerical Methods with Matlab for Engineers and Scientists. McGraw Hill, 2012
- [6] Бахвалов, Н.С., Лапин, А.В., Чижонков, Е.В.: Численные методы в задачах и упражнениях. Высшая школа, 2000
- [7] Hollis, S: Manual for Stewart’s Single Variable Calculus. Brooks/Cole, 2008
- [8] Antia, H. M.: Numerical Methods for Scientists and Engineers. McGraw-Hill Publishing Company Ltd., 1991