Mechanism of controlling the process of the convergence of the Newton iteration metod

Kazakov D. S. Nikonov E. G.

The **aim** of this work is the analysis of numerical iterative schemes based on Newton's method and the development of a mechanism to control the properties of convergence.

The **result** is the algorithm of the mechanism controlling the properties of convergence of the iterative numerical schemes implemented in the form of software in C++.

The **evaluation criteria**: Increase the rate of convergence of the iterative process.

To achieve this goal it was necessary to solve the following **tasks**:

— to analyze the peculiarities of numerical methods for solving nonlinear equations;

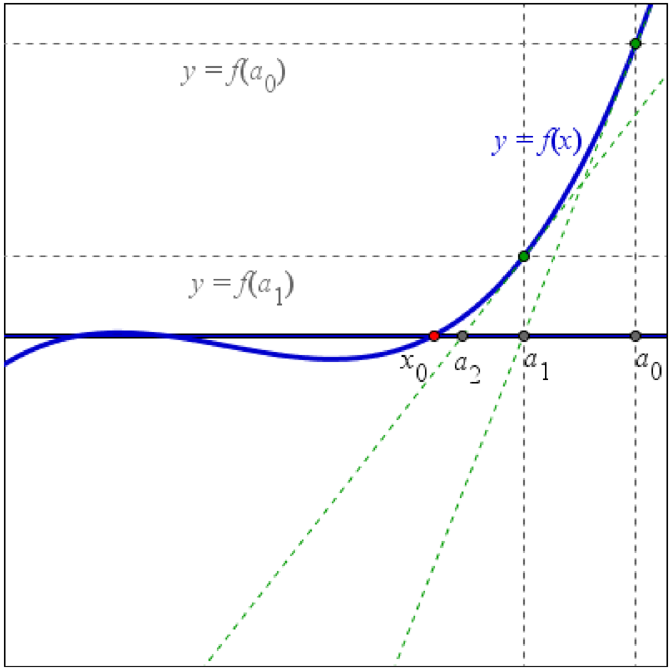
— to study the approaches to solving the problem of choosing a control parameter;

— to develop and test the software.

Introduction

With a wide range of computational problems in various fields of science and technology the need arises for numerical solution of the equations of one or several variables, in particular when using implicit difference schemes for the solution of ordinary differential equations and partial differential equations, integral and other, usually non-linear equations. The use of traditional iterative methods is often inefficient because of the relatively large number of iterations, which inevitably leads to a significant increase in computation time. The question of the construction of efficient algorithms for solving nonlinear equations is still relevant in computational mathematics and its applications, in particular mathematical modeling, for example, in the simulation method of molecular dynamics where at each time step it is necessary to solve numerically from 10^3 to 10^7 - 10^{10} differential equationsIn this case, the price of even one iteration increases by a factor of millions in solving one equation. One of the effective iterative methods for solving nonlinear equations is the Newton method and its various modifications, including on the basis of the continuous analog of the Newton method (CANM).

Newton's method



The derivation of the Newton method formula is based on the formula of the tangent equation drawn to the graph of the function $f(x)$ at the point x^* sufficiently close to the solution of the equation $f(x)=0$.

$$f(x) = f(x_0) + f'(x)(x - x_0)$$

Assuming that the solution x^* of the equation $f(x)$=0 lies in a neighborhood of the point x_0 and $f'(x_0)\neq 0$, substituting $x = x^*$ in formula of the tangent equation

$$0 \approx f(x_0) + f'(x_0)x^* - f'(x_0)\,x_0$$

We obtain the following approximate expression for the unknown solution x^*

$$x^* \approx x_0 - f(x_0)f'(x_0)^{-1}.$$

This algorithm allows us to obtain a new approximation x_1 to the solution of the equation x^* if the derivative exists and is not zero along the entire segment $(x^*;\ x_0)$

$$x_1 = x_0 - f(x_0)f'(x_0)^{-1}$$

Now this is a new approach of $x_1$can be used instead of the old $x_0$in order to find even more accurate approximation x_2 to the solution of the equation x^*. If you know the value of x_{i-1} for the i-th step of the iterative process, then the following approximation can be obtained by the formula

$$x_i = x_{i-1} - f(x_{i-1})f'(x_{i-1})^{-1}$$

A continuous analog of Newton's method

In the Newton method, the value x_i calculated at each step can be considered as a function of the entire variable i: $x_i = x(i)$, and $x(0) = x_0$ is given. It is possible to consider the entire variable i as a continuous variable t, $0 \leq t \leq \infty$. And assuming that the function $x(t)$ is differentiable, we obtain the following expression instead of algorithm of Newton's method:

$$x'(t) = -f(x(t))f'(x(t))^{-1}, 0 \leq t < \infty, x(0) = x_0$$

This equation can usually be solved only approximately. One can use Euler's method, one of the simplest methods of approximate solution of the problem, based on the approximate representation of the derivative $x'(t)$ at a fixed point t in the form:

$$x'(t) \approx \frac{x(t+\Delta t) - x(t)}{\Delta t}$$

Where Δt is a small increment of the argument t.

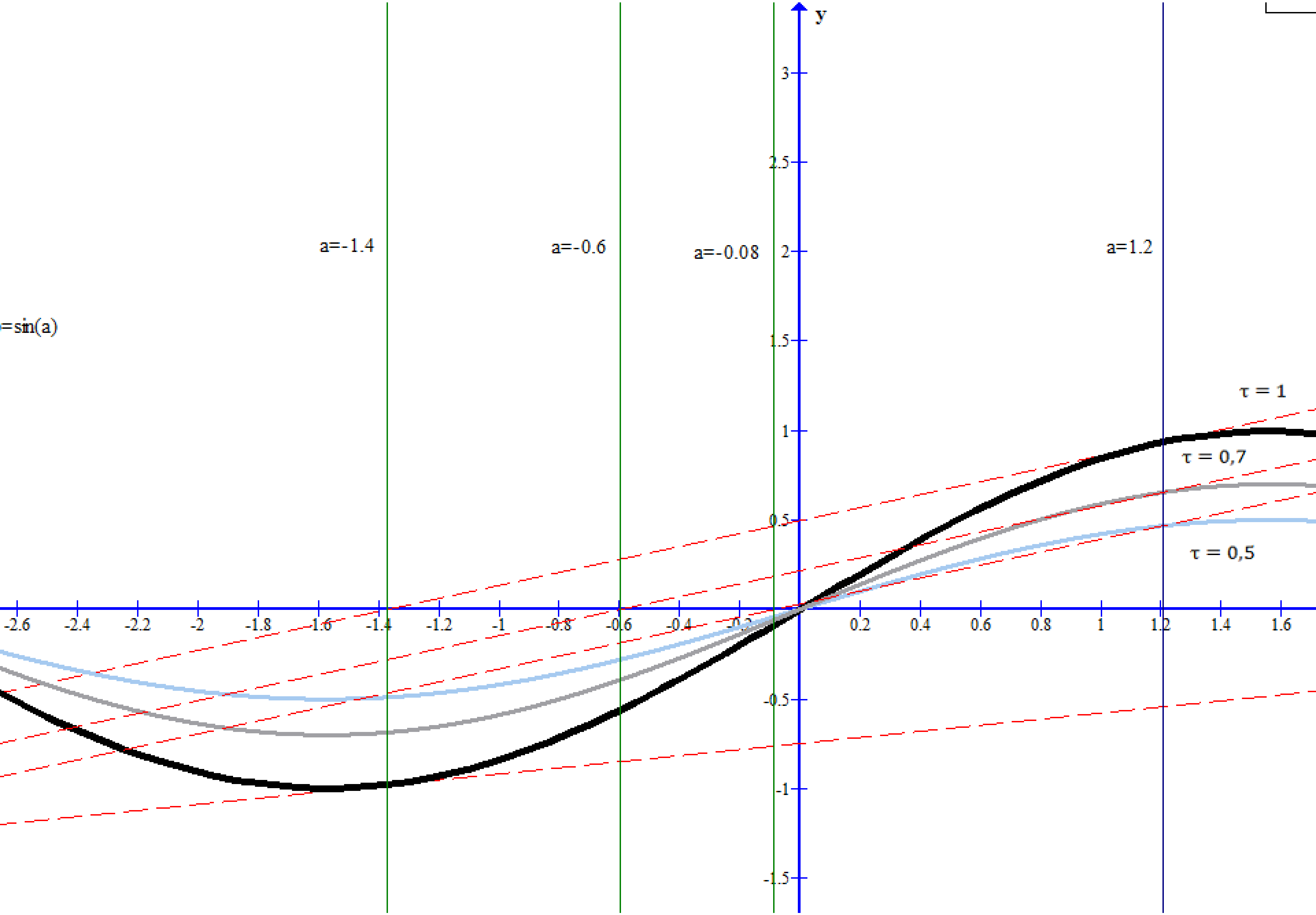
We define a system of points:

$$t = t_i,\ i = 0,1,2,3\ldots,\ \tau_i = t_{i+1} - t_i,\ t_0 = 0,\ \tau > 0$$

Where τ is the distance between neighboring points.

Further we obtain:

$$\frac{x(t_{i+1}) - x(t_i)}{\tau_i} \approx -f(x(t_i))f'(x(t_i))^{-1},\ x(t_0) = x_0,\ i = 0,1,2,3\ldots.\qquad 0 < \tau \leq 1$$


y
3
2.5
2
1.5
1
0.5
-0.5
-1
-1.5
-2.6
-2.4
-2.2
-2
-1.8
-1.6
-1.4
-1.2
-1
-0.8
-0.6
-0.4
-0.2
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
a=-1.4
a=-0.6
a=-0.08
a=1.2
=sin(a)
τ = 1
τ = 0,7
τ = 0,5

The method of specifying a control parameter for the CANM

Given the fact that the function must be monotonic close to the root as a parameter which determines the precision used, the module value of the function at the point defined in each iteration step, the value is called inviscid. In search of an algorithm that provides the minimum value of the residual at each step, it was decided to use a recurrence formula based on $\frac{\tau_i}{\tau_{i-1}}$.

$$\frac{\tau_i}{\tau_{i-1}} = \frac{\|f(x_{i-1})\|}{\|f(x_i)\|},\ \ i = 1,2,\ldots.$$

The valueτ_0 is chosen, many of the calculations using this formula provided a stable and relatively fast convergence of a sequence of approximations to the desired solution $0.01 < \tau \leq 0.1$. How can we lead from the formula, if you increase the value $\|f(x_i)\|$ of the residual step size τ_i increases, while increasing the residual is reduced to the iterative process is not degenerated it is necessary to clarify the interval.

$$0 < \tau \leq 1$$

Investigations of the step size factor for NAMS.

In the process of calculating CANM, it is worthwhile to consider in detail the moment when $f''(x_i)$ changes its sign on the segment $[x_{i-1}; x_i]$. In view of our assumption that $f(x_i)$ has a unique root x^* on the interval $[x_{i-1}; x_i]$ and $f'(x) \neq 0$, it is obvious that the value $\|x_{i-1} - x^*\|$ Must be greater than $\|x_{i+1} - x^*\|$, in this case there is no need to calculate the coefficient τ_{i+1}, using the CANM algorithm, since we know the value x_{i-1}, which Can be used to calculate the coefficient τ_{i+1}. We set the coefficient k, which denotes an x_{i-1} such that $\frac{\varphi(x_{i-1})}{\varphi(x_i)} < 0$. Using CANM, we can calculate the value τ_{i+1} required In order that $x_{i+1} = k$.

$$v = f(x_{i-1}) f'(x_{i-1})^{-1}$$

$$\tau_{i+1} = \frac{(k - x_{i-1})}{v}$$

If the computed value is $\tau > 1$, then we set $\tau = 1$ and use the CANM algorithm to calculate x_i while maintaining the quadratic convergence rate. The most interesting case is when $\tau \leq 1$, it is obvious that it is necessary to decrease the step value. Based on the data obtained during a series of test calculations, it was decided to use the following algorithm:

$$\tau_i = \tau - \frac{\frac{\varphi(k)}{\varphi(m)} + \left(1 - \frac{\varphi(x_i)}{\varphi(l)}\right)}{2}$$

In the section where the function changes its sign, an extremely low rate of convergence of the NAMN is possible, in order to avoid divergence, under the following conditions:

$$\frac{\|l\|-\|m\|}{\|l-x_n\|+\|m-k\|} > 5;\ c > 4;\ c = 1,2,3,\ldots.$$

Where c - is the number of successive iterations made under bilaterally convergent conditions, use the algorithm for a certain step x_{i+1} :

$$x_{i+2} = \frac{\|x_i\| - \|k\|}{2}.$$

It is obvious that the algorithm for calculating the step size coefficient of the CANM can be represented as:

$$\tau_i = \frac{\tau_{i-1} * \|f(x_{i-1})\|}{\|f(x_i)\|} = \frac{\tau_0 * \|f(x_0)\|}{\|f(x_i)\|}, \qquad \tau_0 * \|f(x_0)\| < \|f(x_i)\|,$$

As a result, the formula for calculating the iteration of CANM can take the form:

$$x_i = x_{i-1} - \tau_0 * \|f(x_0)\| * \|f(x_{i-1})\|^{-1} * f(x_{i-1}) * f'(x_{i-1})^{-1}$$

While running the algorithm, you can use the value $f(x_i)$ instead of $\|f(x_i)\|$, if you define the sign $f(x_0)$ and use this algorithm only until the function changes sign, as a result we get:

$$x_i = x_{i-1} - \tau_0 * \|f(x_0)\| * f'(x_{i-1})^{-1} \quad \text{если} \qquad \tau_0 * \|f(x_0)\| > f(x_i),$$

$$\text{то}\ x_i = x_{i-1} - \|f(x_i)\| * f'(x_{i-1})^{-1}$$

It should be noted that using this algorithm reduces the total number of operations on 30%

modified CAMN

```
do
{
      if (i / f(c)<0)
      {
            i = -i;
      }
      if (fabs(i) > fabs(f(c)))
      {
            i = f(c);
      }
      c = c - i / f1(c);
      N = N + 1;
} while (fabs(f(c)) >= eps);
```

T=0,984

CAMN

```
do
{
      a = c;
      c = c - b*f(c) / f1(c);
      b = b*fabs(f(a)) / (f(c));
      if (1 < b)
      {
            b = 1;
      }
      N = N + 1;
} while (fabs(f(c)) >= eps);
```

T=1,11

Calculations were performed using a processor: intel core i3-5005U SPU@ 2.00GHz 2.00GHz

In the course of comparing the time required for the spreading of values using two algorithms, it was established that the proposed algorithm for calculating the function without changing the sign, without limiting the value of the derivative, is performed with less time.

It is obvious that using the algorithm, where $\tau_0 * \|f(x_0)\|$ is a constant, only $f'(x_{i-1})$ affects $\|x_i - x_{i-1}\|$, taking into account that $x_{i-1} - x^* > x_i - x^*$, the probability of divergence of CANM increases after each iteration. To reduce the degree of divergence, the authors proposed the following algorithm:

$$x_i = x_{i-1} - \tau_0 * \|f(x_0)\| * {m_i}^{-1}$$

Provided that $\|m_i\| > \|f'(x_i)\|$

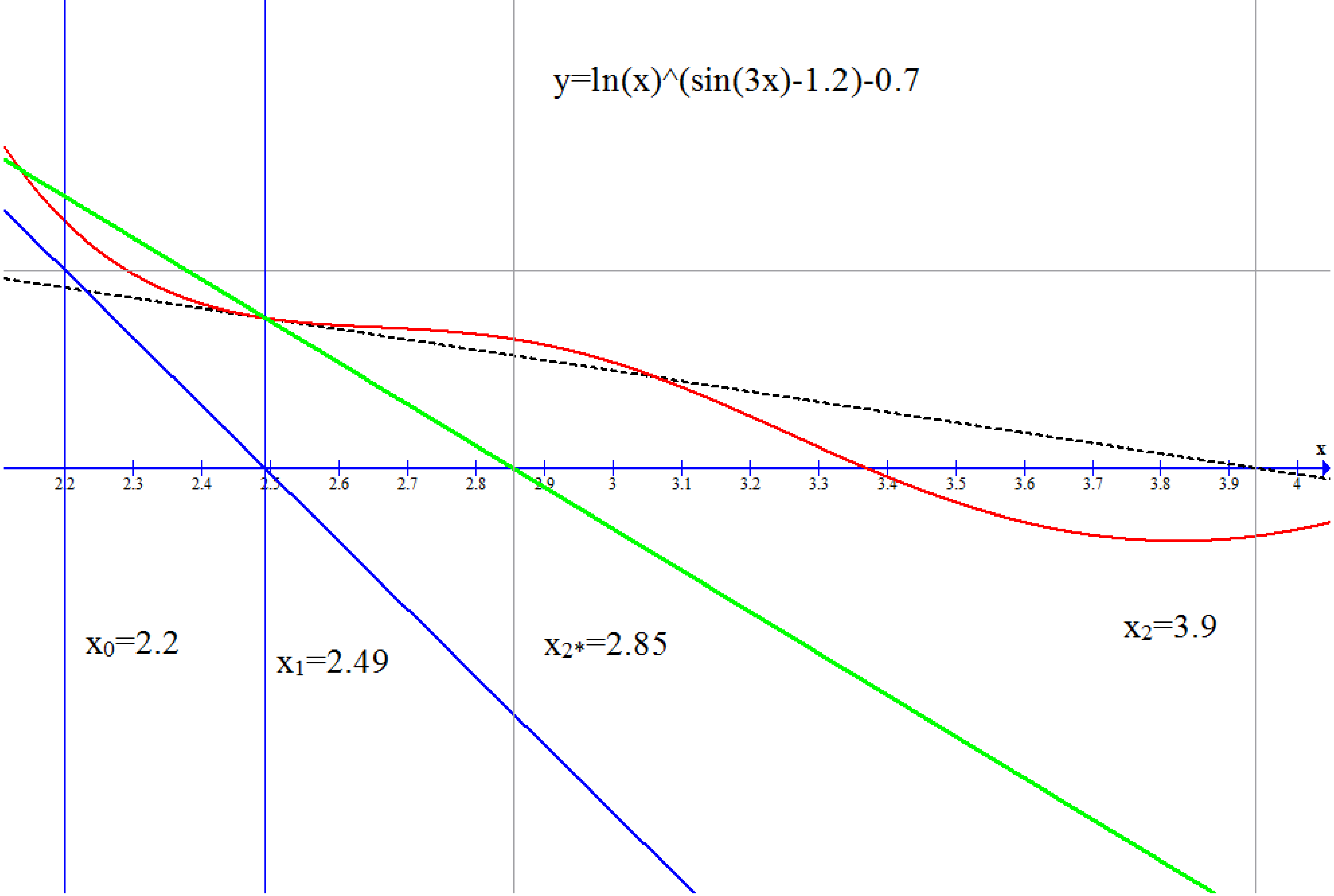
$$m_i = m_i - \frac{(m_i - f\prime(x_i))}{\sqrt{(i+1)*g}}$$

g=τ_0*d, g>0.5

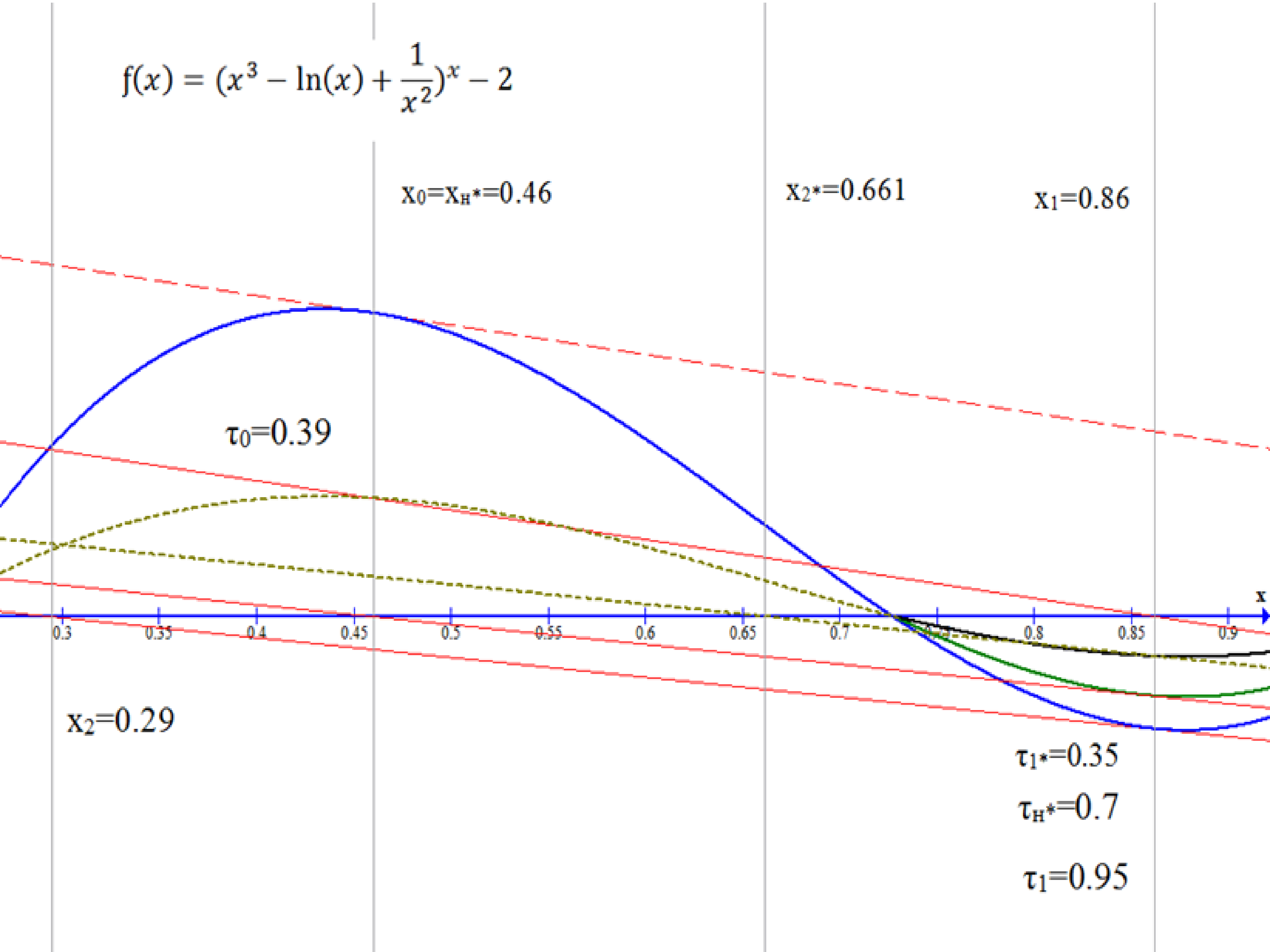
Where is d set initially

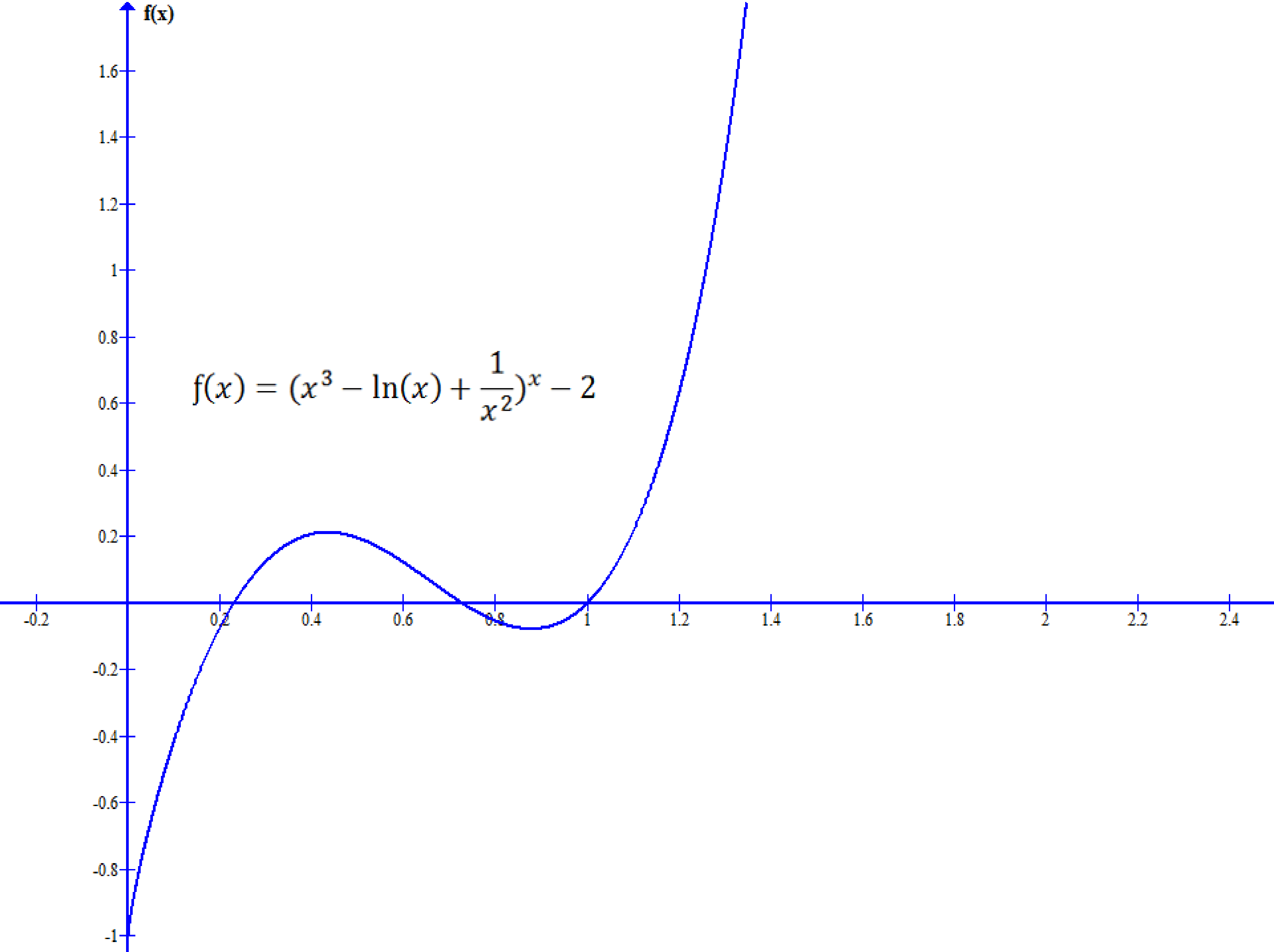
Otherwise $m_i = f'(x_i)$

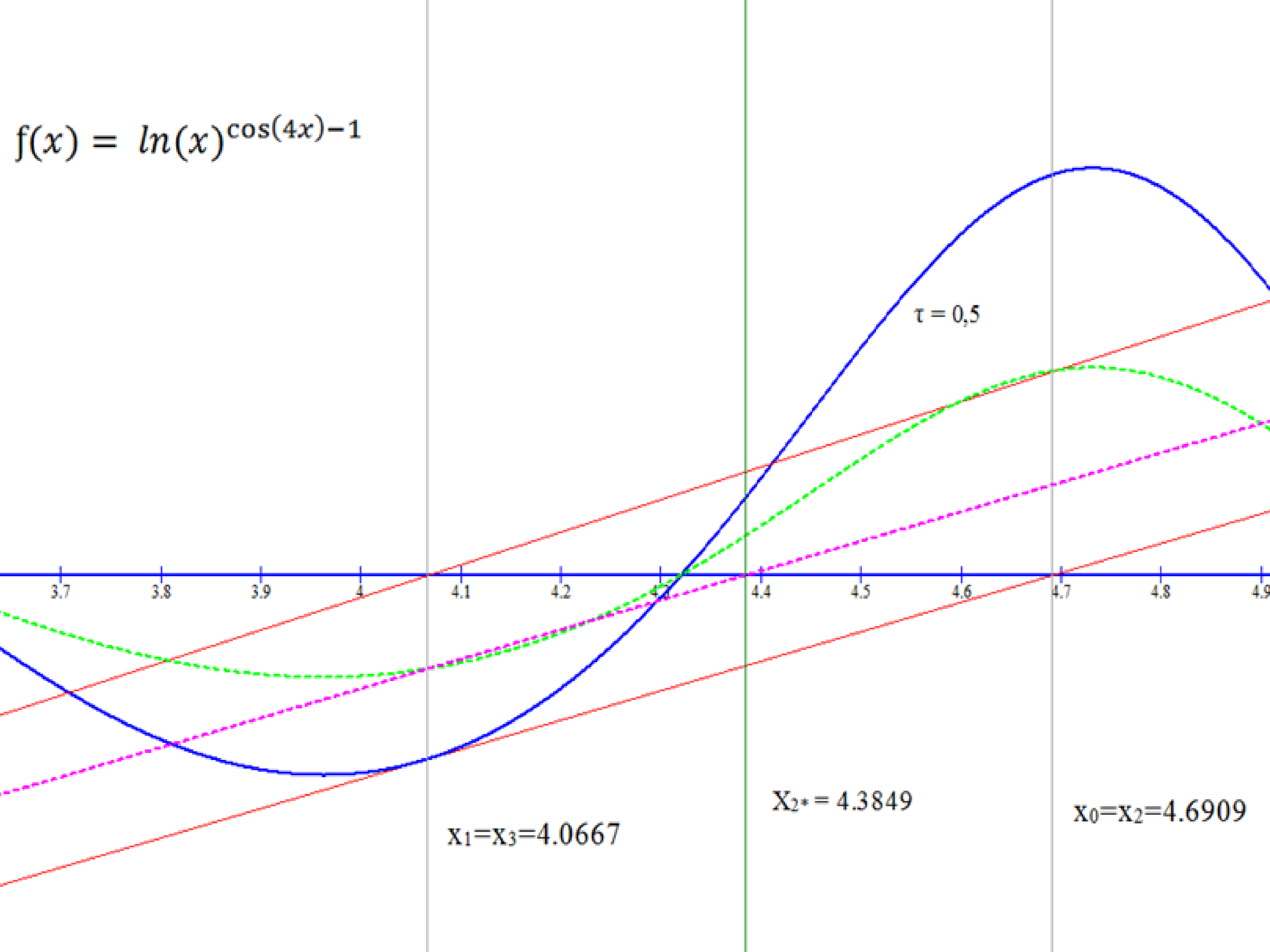
The presented modification reduces the probability of divergence, but at the same time a much larger number of iterations is required, if $f'(x_i)$ strongly decreases from iteration to iteration.

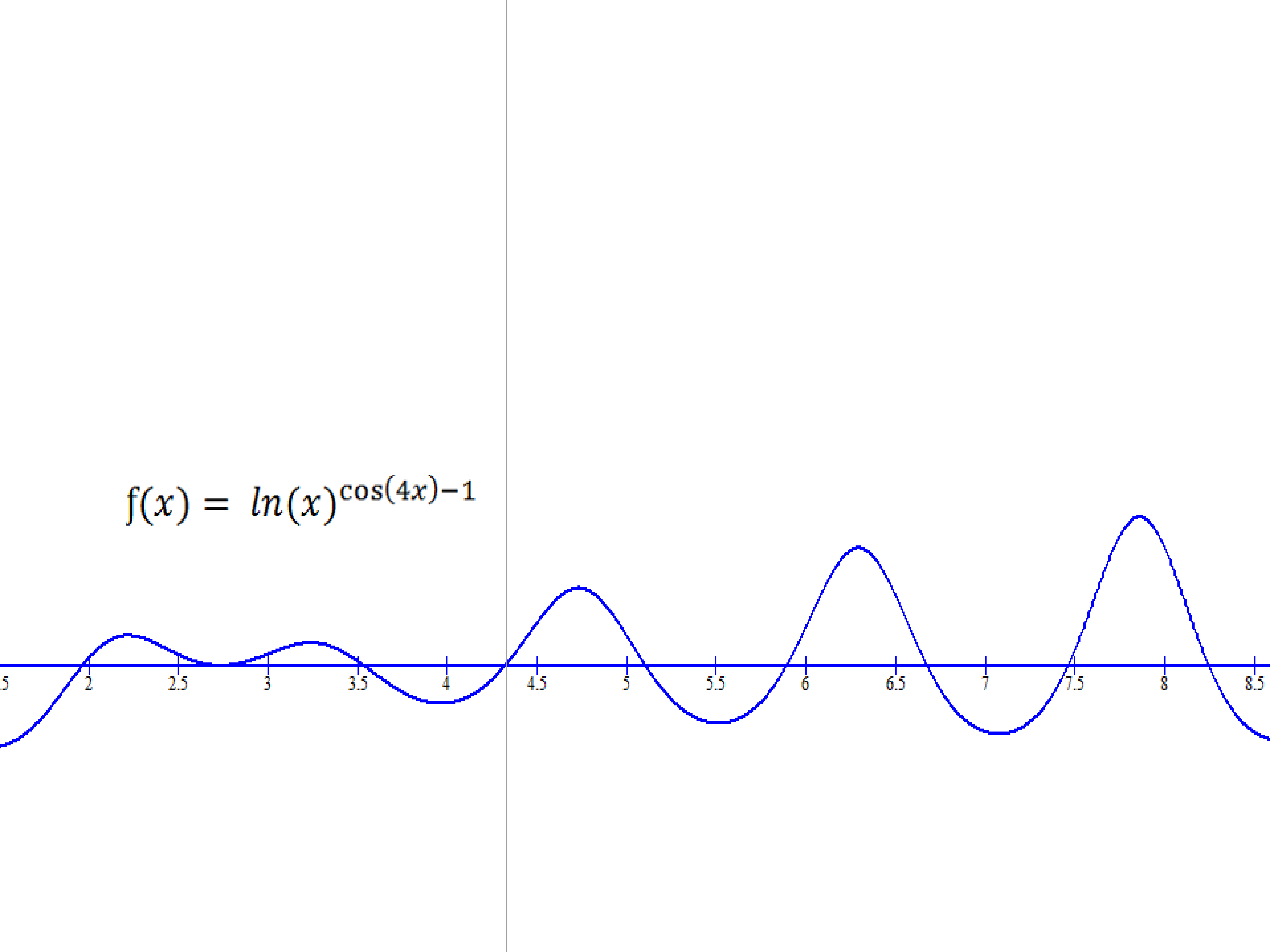


№	f’(x)	m	f(x)	x
3	-3.83561	-4.79009	1.54692	1.88704
4	-3.10683	-4.47193	1.19028	1.95740
5	-2.39804	-4.13065	9.05684e-001	2.03357
6	-1.75019	-3.78413	6.85665e-001	2.11671


$f(x) = (x^3 - \ln(x) + \frac{1}{x^2})^x - 2$
$x_0=x_{H*}=0.46$
$x_{2*}=0.661$
$x_1=0.86$
$\tau_0=0.39$
$x_2=0.29$
$\tau_{1*}=0.35$
$\tau_{H*}=0.7$
$\tau_1=0.95$
x
0.3
0.35
0.4
0.45
0.5
0.55
0.6
0.65
0.7
0.8
0.85
0.9


f(x)
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
-0.2
-0.4
-0.6
-0.8
-1
-0.2
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
2.2
2.4
$f(x) = (x^3 - \ln(x) + \frac{1}{x^2})^x - 2$


$f(x) = \ln(x)^{\cos(4x)-1}$
$\tau = 0{,}5$
3.7
3.8
3.9
4
4.1
4.2
4.3
4.4
4.5
4.6
4.7
4.8
4.9
$x_1 = x_3 = 4.0667$
$X_{2*} = 4.3849$
$x_0 = x_2 = 4.6909$



$$f(x) = \ ln(x)^{\cos(4x)-1}$$

Conclusion

Using the example of a developed algorithm for a modified ANMN, it was shown that it is possible to control the rate of convergence of ANMN using the step change coefficient of the difference scheme for the numerical solution of the differential equation of NAMS as a control parameter. Further development involves the use of interpolation methods to consider cases not provided for by the modification of NAMS. The development and implementation of control mechanisms for iterative processes of solving equations will significantly reduce the time required to calculate the desired value, which will significantly improve the efficiency of algorithms for solving nonlinear equations.