Міністерство освіти і науки, молоді та спорту України
Львівський національний університет імені Івана Франка
Факультет електроніки

Звіт

Про виконання лабораторної №10
з курсу «Чисельні методи» на тему

«*Чисельні методи розв'язування нелінійних рівнянь з одним невідомим*»

Виконала :
Студентка групи ФЕІ-31
Литвин В.В.
Перевірив :
доц. Шувар Р. Я

Львів 2012

Мета: вивчити однокрокові та багатокрокові ітераційні методи розв’язку нелінійних рівнянь з одним невідомим.

Основні теоретичні відомості

Нехай нам задано нелінійне рівняння:

$$F(x)=0, \tag{1}$$

де $F(x)$ – неперервна і достатньо гладка функція дійсної змінної. Для його чисельного розв’язку рівняння (1) виділяють 2 етапи: виділення коренів та уточнення коренів за допомогою ітераційної процедури.

Етап виділення коренів полягає у встановленні числа, характеру і початкового наближення коренів рівняння (1).

Основними методами чисельного розв’язку нелінійних рівнянь є ітераційні методи, які в свою чергу поділяються на однокрокові методи та багатокрокові методи.

В основі однокрокових методів лежить метод простої ітерації. Метод простої ітерації використовує представлення вихідного рівняння у вигляді

$$x=f(x). \tag{2}$$

Суть методу полягає в тому, що ми спочатку вибираємо деяке нульове наближення значення кореня рівняння x_0 і всі послідуючі наближення до точного значення кореня x^* обчислюємо, використовуючи однокрокову ітераційну послідовність:

$$x_{n+1}=f(x_n). \tag{3}$$

Ітераційна послідовність наближених значень деякого кореня $\{x_n\}$ має збіжність p-го порядку до точного значення кореня x^*, якщо похибка наступної ітерації задовольняє умові

$$\left|x_{n+1}-x^*\right|=q\left|x_n-x^*\right|^p, \tag{4}$$

де $q>0$ - деяка стала.

Дослідимо умову збіжності методу простої ітерації. Виконаємо розклад в ряд із залишковим членом в диференціальній формі

$$x_{n+1}-x^*=f(x_{n+1})-f(x^*)=(x_n-x^*)f'(\xi), \qquad \text{де } \xi\in\left[x_n,x^*\right].$$

Отже, якщо $\left|f'(x)\right|<1$ для всіх (x_n), то метод простої ітерації має збіжність першого порядку і ітераційна послідовність є збіжною при довільному початковому наближенні x_0.

Для отримання вихідної формули (2) найчастіше використовують перетворення:

$$x=x+\tau F(x)$$

Метод простої релаксації для якого приймається, що $\tau(x_n)=\tau=const$. Відповідно отримаємо ітераційну процедуру:

$$x_{n+1}=x_n+\tau F(x_n) \tag{5}$$

Як ми вже показали для методу простої ітерації $x_{n+1}=f(x_n)$ достатньою умовою збіжності є $\left|f'(x)\right|<1$ для всіх $\{x_n\}$. Для методу релаксації ця умова досягається шляхом вибору належного значення τ: $-2<\tau F'(x)<0$ для всіх $\{x_n\}$.

Ітерації припиняють, коли виконуються умови

$$\left|F(x_{n+1})\right|<\varepsilon$$

$$\left|x_{n+1}-x_n\right|<\varepsilon$$

$$\left|\frac{(x_n-x_{n-1})^2}{2x_{n-1}-x_n-x_{n-2}}\right|<\varepsilon$$

Метод Ньютона. Для підвищення швидкості збіжності можна використати метод Ньютона. Для отримання формули Ньютона розкладемо функцію $F(x)$ в ряд Тейлора в околі $n+1$ наближення деякого кореня, яке ми вважатимемо достатньо добрим наближенням:

$$F(x_{n+1}) = F(x_n) + F'(x_n)(x_{n+1} - x_n) + \ldots = 0. \tag{6}$$

Якщо ми обмежимось першими двома членами розкладу, то отримаємо наступну ітераційну послідовність:

$$x_{n+1} = x_n - \frac{F(x_n)}{F'(x_n)}. \tag{7}$$

Знайдемо порядок методу Ньютона. З формули (7) знаходимо, що

$$f(x) = x - \frac{F(x)}{F'(x)}.$$

Знайдемо вираз для першої похідної:

$$f'(x) = 1 - \frac{(F')^2 - FF''}{(F')^2} = \frac{FF''}{(F')^2}.$$

В околі деякого кореня вихідне рівняння можна представити як $F(x) \approx a(x - x^*)^p$. Знаходимо похідну:

$$f' = \frac{a(x-x^*)^p \, p(p-1) a (x-x^*)^{p-2}}{p^2 a^2 (x-x^*)^{2p-2}} = \frac{p-1}{p}$$

Для простого кореня $p = 1$ і тому $f'(x^*) = 0$. Виконавши розклад в ряд і обмежившись першим ненульовим доданком, отримаємо:

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

Отже метод Ньютона є методом другого порядку.

На практиці для збіжності методу Ньютона початкове наближення x_0 вибирають з умови, щоб $F(x_0)F''(x_0) \geq 0$.

В модифікованому методі Ньютона використовується наступна ітераційна послідовність:

$$x_{n+1} = x_n - \frac{F(x_n)}{F'(x_p)}. \tag{8}$$

В цьому методі значення похідної обчислюється не на кожному ітераційному кроці, а тільки на кроках оновлення x_p.

Метод Чебишева. Аналогічно методу Ньютона розкладемо функцію $F(x)$ в ряд в околі $n+1$ наближення деякого кореня, тільки на відміну від методу Ньютона залишимо вже три члени розкладу:

$$F(x_{n+1}) = F(x_n) + F'(x_n)(x_{n+1} - x_n) + \frac{1}{2}F''(x_n)(x_{n+1} - x_n)^2 = 0 \tag{9}$$

Розв'яжемо отримане квадратне рівняння відносно x_{n+1}:

$$(x_{n+1}^2 - 2x_{n+1}x_n + x_n^2)F'' + 2(x_{n+1} - x_n)F'(x_n) + 2F(x_n) = 0,$$

$$F_n' x_{n+1}^2 + (2F_n' - 2x_n F_n'')x_{n+1} + (x_n^2 F_n'' - 2x_n F_n' + 2F_n) = 0,$$

$$x_{n+1} = \frac{-(F_n' - x_n F_n'') \pm \sqrt{(F_n' - x_n F_n'')^2 - F_n''(x_n^2 F_n'' - 2x_n F_n' + 2F_n)}}{F_n''} =$$

$$= \frac{-\left(F_n' - x_n F_n''\right) \pm \sqrt{F_n' - 2F_n F_n''}}{F_n''} = \frac{-\left(F_n' - x_n F_n''\right) \pm F_n' \left(1 - \dfrac{2F_n F_n''}{F_n'^2}\right)^{\frac{1}{2}}}{F_n''} =$$

$$= \frac{-\left(F_n' - x_n F_n''\right) + F_n' \left(1 - \dfrac{2}{2}\dfrac{F_n F_n''}{F_n'^2} - \dfrac{4}{8}\left(\dfrac{F_n F_n''}{F_n'^2}\right)^2\right)}{F_n''} = x_n - \frac{F_n}{F_n'} - \frac{1}{2}\frac{F_n^2 F_n''}{F_n'^3}$$

Отримаємо наступну уточнену ітераційну процедуру:

$$x_{n+1} = x_n - \frac{F(x_n)}{F'(x_n)} - \frac{1}{2}\frac{F^2(x_n)F''(x_n)}{\left(F'(x_n)\right)^3}. \tag{10}$$

Для побудови багатокрокових ітераційних методів функцію $F(x)$ заміняють інтерполяційним многочленом, побудованим на вузлах $x_n, x_{n-1}, \ldots$ Нове наближення x_{n+1} знаходять як нуль побудованої інтерполяційної функції. У загальному ми можемо записати:

$$\begin{gathered} F(x) = F(x_n) + (x - x_n)F(x_n, x_{n-1}) + \\ + (x - x_n)(x - x_{n-1})F(x_n, x_{n-1}, x_{n-2}) + \ldots \end{gathered} \tag{11}$$

Залежно від того, на якому члені ми обриваємо ряд, ми можемо отримати різні типи багатокрокових ітераційних формул. У найпростішому випадку, коли обмежитись двома доданками:

$$F(x_{n+1}) = F(x_n) + (x_{n+1} - x_n)F(x_n, x_{n-1}) = 0, \tag{12}$$

ми отримаємо формулу методу хорд:

$$x_{n+1} = x_n - \frac{F(x_n)}{F(x_n, x_{n-1})} = x_n - F(x_n)\frac{x_n - x_{n-1}}{F(x_n) - F(x_{n-1})}.$$

Знайдемо швидкість збіжності методу хорд. Для цього перепишемо вихідне рівняння у вигляді

$$x_{n+1} - x^* = x_n - x^* - \frac{\left(\left(x_n - x^*\right) - \left(x_{n-1} - x^*\right)\right)F\left(x^* + \left(x_n - x^*\right)\right)}{F\left(x^* + \left(x_n - x^*\right)\right) - F\left(x^* + \left(x_{n-1} - x^*\right)\right)}$$

Виконаємо розклад в околі точного значення кореня x^* по малій величині $\left(x_n - x^*\right)$

$$x_{n+1} - x^* = x_n - x^* -$$
$$- \frac{\left(\left(x_n - x^*\right) - \left(x_{n-1} - x^*\right)\right)\left(F\left(x^*\right) + \left(x_n - x^*\right)F'\left(x^*\right) + \frac{1}{2}\left(x_n - x^*\right)^2 F''\left(x^*\right)\right)}{\left(F\left(x^*\right) + \left(x_n - x^*\right)F'\left(x^*\right) + \frac{1}{2}\left(x_n - x^*\right)^2 F''\left(x^*\right)\right) - \left(F\left(x^*\right) + \left(x_{n-1} - x^*\right)F'\left(x^*\right) + \frac{1}{2}\left(x_{n-1} - x^*\right)^2 F''\left(x^*\right)\right)}$$

Отримаємо далі після нескладних перетворень

$$x_{n+1} - x^* = x_n - x^* - \frac{\left(\left(x_n - x^*\right) - \left(x_{n-1} - x^*\right)\right)\left(\left(x_n - x^*\right)F'\left(x^*\right) + \frac{1}{2}\left(x_n - x^*\right)^2 F''\left(x^*\right)\right)}{\left(\left(x_n - x^*\right) - \left(x_{n-1} - x^*\right)\right)F'\left(x^*\right) + \frac{1}{2}\left(\left(x_n - x^*\right)^2 - \left(x_{n-1} - x^*\right)^2\right)F''\left(x^*\right)} =$$

$$= x_n - x^* - \frac{\left(\left(x_n - x^*\right)F'\left(x^*\right) + \frac{1}{2}\left(x_n - x^*\right)^2 F''\left(x^*\right)\right)}{F'\left(x^*\right) + \frac{1}{2}\left(\left(x_n - x^*\right) + \left(x_{n-1} - x^*\right)\right)F''\left(x^*\right)} =$$

$$= \left(x_n - x^*\right) - \left(x_n - x^*\right)\frac{\left(F'\left(x^*\right) + \frac{1}{2}\left(x_n - x^*\right)F''\left(x^*\right)\right)}{F'\left(x^*\right) + \frac{1}{2}\left(\left(x_n - x^*\right) + \left(x_{n-1} - x^*\right)\right)F''\left(x^*\right)} =$$

$$=\left(x_n-x^*\right)\left(1-\frac{\left(1+\frac{1}{2}\left(x_n-x^*\right)\frac{F''\left(x^*\right)}{F'\left(x^*\right)}\right)}{1+\frac{1}{2}\left(\left(x_n-x^*\right)+\left(x_{n-1}-x^*\right)\right)\frac{F''\left(x^*\right)}{F'\left(x^*\right)}}\right)=$$

$$=\frac{1}{2}\frac{F''\left(x^*\right)}{F'\left(x^*\right)}\left(x_n-x^*\right)\left(x_{n-1}-x^*\right)$$

Ми отримали різницеве рівняння для похибки $\varepsilon_n=x_n-x^*$

$$\varepsilon_{n+1}=C\varepsilon_n\varepsilon_{n-1},\ \text{де}\ C=\frac{1}{2}\left|\frac{F''\left(x^*\right)}{F'\left(x^*\right)}\right|$$

Нехай швидкість збіжності методу рівна $\varepsilon_{n+1}=A\varepsilon_n^p$. Тоді $\varepsilon_{n-1}=A^{-1/p}\varepsilon_n^{1/p}$. Підставимо цей вираз у рівняння для похибки $A\varepsilon_n^p=C\varepsilon_n A^{-1/p}\varepsilon_n^{1/p}$. З цього рівняння знаходимо, що $p=1,618$ і

$$A=\left(\frac{1}{2}\left|\frac{F''\left(x^*\right)}{F'\left(x^*\right)}\right|\right)^{1/1,618}$$

Швидкість збіжності методу хорд визначається співвідношення:

$$\left|x_{n+1}-x^*\right|\approx\left|\frac{F''(x^*)}{2F'(x^*)}\right|^{1/1,618}\left|x_n-x^*\right|^{1,618}.$$

Якщо залишити наступний доданок, то ми отримаємо метод парабол:

$$F(x)=F(x_n)+(x-x_n)F(x_n,x_{n-1})+(x-x_n)(x-x_{n-1})F(x_n,x_{n-1},x_{n-2}).$$

Розв'яжемо отримане квадратне рівняння відносно $Z=x-x_n$:

$$(Z+x_n-x_{n-1})Z\cdot F\left(x_n,x_{n-1},x_{n-2}\right)+Z\cdot F\left(x_n,x_{n-1}\right)+F\left(x_n\right)=0.$$

З двох коренів вибираємо найменший за модулем. Ми отримаємо ітераційну послідовність:

$$x_{n+1}=x_n+Z$$

Швидкість збіжності методу парабол $p=1,84$:

$$\left|x_{n+1}-x^*\right|\approx\left|\frac{F'''\left(x^*\right)}{6F'\left(x^*\right)}\right|^{0,42}\left|x_n-x^*\right|^{1,84}$$

Відмітимо, що метод парабол дозволяє знайти комплексні корені.

В методі зворотної інтерполяції використовують інтерполяційний многочлен Лагранжа. Нехай нам задана послідовність наближених значень коренів рівняння: $x_n,x_{n-1},x_{n-2}\ldots$ Використовуючи ці значення, побудуємо інтерполяційний многочлен Лагранжа $x=L\left(y\right)$. У найпростішому випадку

$$x=L_1\left(y\right)=\frac{y-y_n}{y_{n-1}-y_n}x_{n-1}+\frac{y-y_{n-1}}{y_n-y_{n-1}}x_n.$$

Отримаємо ітераційну послідовність:

$$x_{n+1}=L_1(0)=-\frac{y_n}{y_{n-1}-y_n}x_{n-1}-\frac{y_{n-1}}{y_n-y_{n-1}}x_n. \tag{13}$$

Можна записати більш точнішу формулу для випадку трьох вузлів

$$x = L_2(y) = \frac{(y - y_{n-1})(y - y_n)}{(y_{n-2} - y_{n-1})(y_{n-2} - y_n)} x_{n-2} + \frac{(y - y_{n-2})(y - y_n)}{(y_{n-1} - y_{n-2})(y_{n-1} - y_n)} x_{n-1} +$$
$$+ \frac{(y - y_{n-2})(y - y_{n-1})}{(y_n - y_{n-2})(y_n - y_{n-1})} x_n$$

Знаходимо ітераційну послідовність

$$x_{n+1} = \frac{y_{n-1} y_n}{(y_{n-2} - y_{n-1})(y_{n-2} - y_n)} x_{n-2} + \frac{y_{n-2} y_n}{(y_{n-1} - y_{n-2})(y_{n-1} - y_n)} x_{n-1} +$$
$$+ \frac{y_{n-2} y_{n-1}}{(y_n - y_{n-2})(y_n - y_{n-1})} x_n \qquad (14)$$

Для покращення швидкості збіжності ітераційних методів можна використати метод Ейткена:

Нехай ми маємо ітераційний метод з лінійною збіжністю. Для трьох послідовних наближень значень коренів можемо записати, що

$$x_n - x^* = q^n \left(x_0 - x^* \right) + \theta \left(q^{n+1} \right)$$
$$x_{n-1} - x^* = q^{n-1} \left(x_0 - x^* \right) + \theta \left(q^n \right)$$
$$x_{n-2} - x^* = q^{n-2} \left(x_0 - x^* \right) + \theta \left(q^{n-1} \right)$$

Використовуючи ці рівняння, отримаємо квадратне рівняння відносно уточненого значення кореня

$$\left(x_{n-1} - x^* \right)^2 = \left(x_n - x^* \right)\left(x_{n-2} - x^* \right),$$

$$x_{n-1}^2 - 2x_{n-1}x^* = x_n x_{n-2} - x^* x_{n-2} - x^* x_n\,,$$
$$x^* = \frac{x_{n-1}^2 - x_n x_{n-2}}{2x_{n-1} - x_{n-2} - x_n} = \frac{\left(x_n - x_{n-1} \right)^2 - x_n^2 + 2x_n x_{n-1} - x_n x_{n-2}}{2x_{n-1} - x_{n-2} - x_n}.$$

Отримаємо ітераційну послідовність з на порядок швидшою збіжністю:

$$x_{n+1} = x_n + \frac{\left(x_n - x_{n-1} \right)^2}{2x_{n-1} - x_{n-2} - x_n} + \theta \left(q^{n+1} \right).$$

На практиці це метод реалізують наступним чином. Спочатку знаходимо за методом простої ітерації два послідовні наближення:

$$x_1 = f\left(x_0 \right), \quad x_2 = f\left(x_1 \right).$$

Наступне наближення знаходимо по методу Ейткена:

$$x_3 = x_2 + \frac{\left(x_2 - x_1 \right)^2}{2x_1 - x_2 - x_0};$$

Далі ітераційний процес повторюється знову до досягнення заданої точності:

$$x_4 = f\left(x_3 \right), \quad x_5 = f\left(x_4 \right),$$
$$x_6 = x_5 + \frac{\left(x_5 - x_4 \right)^2}{2x_4 - x_5 - x_3}.$$

Вхідні дані

Задано нелінійне рівняння

$$F(x) = \cos(x^2)$$

Для знаходження розв’язку цього рівняння задано початкове наближення, яке рівне

X = -1.05

Також задано точність обчислень *eps = 1E-12*.

Текст програми.

```
program lab10;
var
  iter, itermax, err : integer;
  xn1, x1, x2, xn, eps, tau, xm1 : real;
  outFile : text;

function f(x:real):real;
begin
     f := cos(x * x);
end;

function fp(x:real):real;
begin
     fp := - 2 * x * sin(x * x);
end;

function f2p(x:real):real;
begin
     f2p := -2 * sin(x * x )-4 * x * x * cos(x * x);
end;
begin
     assign(outFile, 'out.txt');
     rewrite(outFile);

     xn1 := -1.05;
     itermax := 10000;
     eps := 1e-12;
     tau := 0.25;

     //проста ітерація
     writeln ('Simple iterations: ');
     iter := 0;
     err := 0;
     repeat
           xn := xn1;
           xn1 := xn + tau * f(xn);
           iter := iter + 1;
     until  ((abs(xn1 - xn) < eps) AND (abs(f(xn1)) < eps) or (iter >
itermax));
     writeln ('iter= ' ,iter);
    writeln ('x= ',xn1);
    writeln();
    writeln(outFile, 'Simple iterations: iterations = ', iter, ' x= ',
xn1);

     //Ньютона
     writeln (' Nutona:');
     xn1 := -1.05;
     iter := 0;
```

```
            err := 0;
            repeat
                  xn:=xn1;
                  xn1 := xn-f(xn)/fp(xn);
                  iter := iter + 1;
            until  ((abs(xn1  -  xn)<eps)  AND  (abs(f(xn1))  <  eps)  or  (iter  >
itermax));
           writeln ('iter= ' ,iter);
           writeln ('x= ',xn1);
           writeln;
           writeln(outFile, 'Method Newton: iterations = ', iter, ' x= ', xn1);

         //метод чебишева
           writeln ('Method of Chebusheva:');
            xn1 := -1.05;
            iter := 0;
            err := 0;
            repeat
                  xn := xn1;
                  xn1                   :=                   xn-(f(xn)/fp(xn))-
(1/2)*(sqr(f(xn))*f2p(xn))/(fp(xn)*fp(xn)*fp(xn));
                  iter := iter+1;
            until ((abs(xn1-xn)<eps) AND (abs(f(xn1))<eps) or (iter > itermax));
           writeln ('iter= ' ,iter);
           writeln ('x= ',xn1);
           writeln;
           writeln(outFile, 'Method of Chebusheva: iterations  = ', iter, ' x=
', xn1);

           writeln ('Metod of chords:');
            iter:=0;
            err:=0;
            xn1 := -1.05;
            xm1 := xn + tau * f(xn);
            repeat
                  xn := xn1;
                  xn1 := xn - f(xn) * ((xn-xm1) / (f(xn)-f(xm1)));
                  iter := iter + 1;
            until  ((abs(xn1  -  xn)  <  eps)  AND  (abs(f(xn1))<eps)  or  (iter  >
itermax));
           writeln ('iter= ' ,iter);
           writeln ('x= ',xn1);
           writeln;
           writeln(outFile,  'Method  of  Hord:  iterations   =  ',  iter,  '  x=  ',
xn1);

            writeln('Method of simple relaxation:');
            iter := 0;
            err := 0;
            xn1 := -1.05;
            repeat
                  xn := xn1;
                  xn1 := xn + tau * f(xn);
                  iter := iter + 1;
            until  ((abs(xn1  -  xn)  <  eps)  AND  (abs(f(xn1))<eps)  or  (iter  >
itermax));
            writeln ('iter= ' ,iter);
           writeln ('x= ',xn1);
```

```
            writeln;
            writeln(outFile, 'Method of simple relaxation: iterations  = ', iter,
' x= ', xn1);

            writeln('Eytkin method:');
            iter := 0;
            err := 0;
            xn1 := -1.05;
            repeat
                  xn := xn1;
                  x1 := xn + tau * f(xn);
                  x2 := x1 + tau * f(x1);
                  xn1 := x2 +  sqr(x2 - x1) / (2 * x1 - x2 - xn);
                  iter := iter + 1;
            until  ((abs(xn1  -  xn)  <  eps)  AND  (abs(f(xn1))<eps)  or  (iter  >
itermax));
             writeln ('iter= ' ,iter);
            writeln ('x= ',xn1);
            writeln;
            writeln(outFile, 'Method of Eytkin: iterations  = ', iter, ' x= ',
xn1);

             writeln('Method of back interpolation:');
            iter := 0;
            err := 0;
            xn1 := -1.05;
            repeat
                  xn := xn1;
                  x1 := xn + tau * f(xn);
                  xn1 := - f(xn) * x1 / (f(x1) - f(xn)) - f(x1) * xn / (f(xn) -
f(x1));
                  iter := iter + 1;
            until  ((abs(xn1  -  xn)  <  eps)  AND  (abs(f(xn1))<eps)  or  (iter  >
itermax));
             writeln ('iter= ' ,iter);
            writeln ('x= ',xn1);
            writeln;
            writeln(outFile,  'Method  of  back  interpolation:  iterations   =  ',
iter, ' x= ', xn1);

            close(outFile);
          readln;
        end.
```

Хід роботи

1. Розглянемо графік заданої функції і оцінимо приблизні корені рівняння (рис. 1).

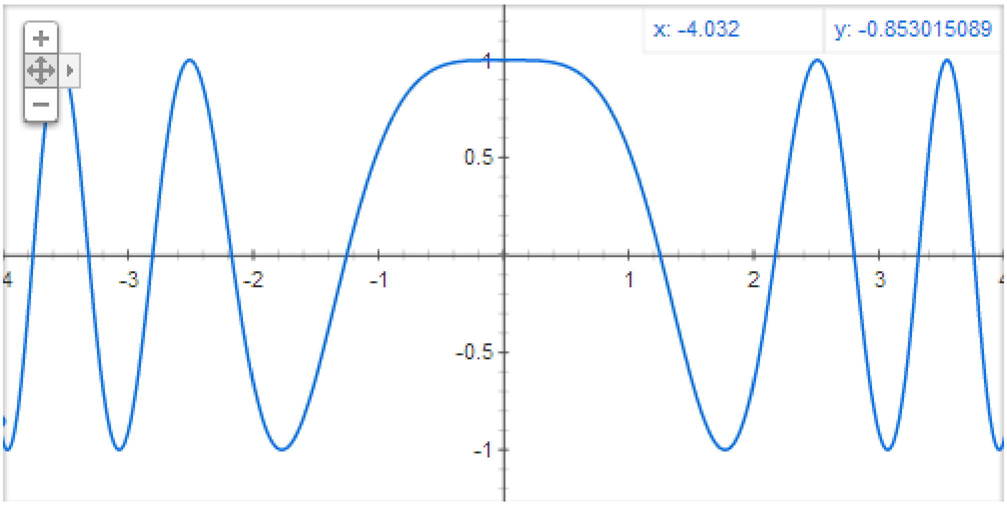

Рис. 2. *Графік функції* $cos(x^2)$

З графіка функції можна вибрати корінь, який приблизно рівний

$$x = -1.25$$

2. Проведемо дослідження чисельних методів розв'язку нелінійних рівнянь. Усі запропоновані методи досягли бажаного результату, тобто ми отримали розв'язок заданого рівняння. Але кожен метод відрізнявся кількістю здійснених ітерацій для знаходження розв'язку. Ми отримали, що знайдений розв'язок відносно заданого наближення рівний

$$x = -1.253314137$$

Наведемо табличку, яка міститиме назву методу, за яким здійснювався пошук розв'язку, і кількість ітерацій.

Назва методу	*Кількість ітерацій*
Простої ітерації	38
Ньютона	5
Чебишева	4
Хорд	2
Релаксації	38
Ейткена	5
Зворотньої інтерполяції	5

Висновок: на даній лабораторній роботі я вивчила однокрокові та багатокрокові методи розв’язку нелінійних рівнянь з одним невідомим. Можу сказати, що найкращу швидкість має метод хорд і Чебишева. Більш повільніші методи, простої ітерації та релаксації, показали порівняно гірші результати, адже знайшли розв’язок за значно більшу кількість ітерацій.