БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра информатики

Факультет НиДО

Специальность ИиТП

Практическая работа № 1

по дисциплине «Методы численного анализа»

Выполнил студент: Дегтярев А.А.

группа 393551

Зачетная книжка № 902021-26

Минск 2015

**Методы решения нелинейных уравнений**

**Многочлен наилучшего средне квадратичного приближения.**

**Метод наименьших квадратов.**

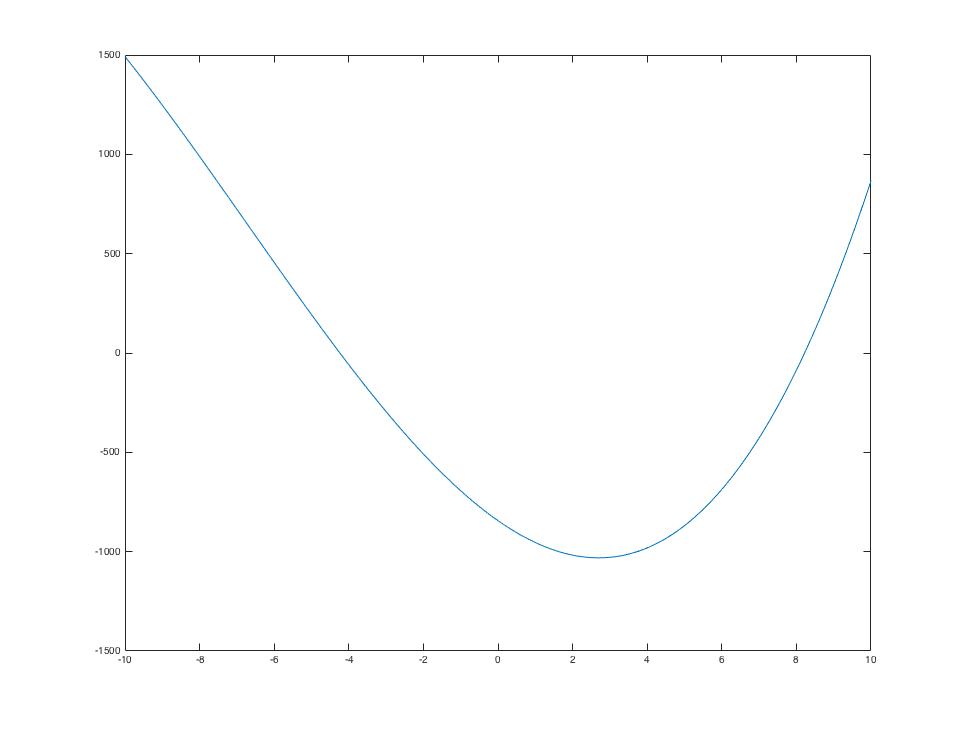
1. Используя теорему Штурма определить число корней уравнения:

x3+ax2 +bx + c=0 на отрезке [-10,10].

За решение данной задачи отвечает метод из модуля EQSolver:

|  |
| --- |
| function res = SturmMethod(f0,lb,ub)  res = 0;  syms x;    f1 = diff(f0);  [q,r] = quorem(f0,f1,x);  f2 = -r;  [q,r] = quorem(f1,f2,x);  f3 = -r;    if (0 >= double(subs(f0,x,lb))) ~= (0 >= double(subs(f1,x,lb)))  res = res + 1;  end    if (0 >= double(subs(f1,x,lb))) ~= (0 >= double(subs(f2,x,lb)))  res = res + 1;  end    if (0 >= double(subs(f2,x,lb))) ~= (0 >= double(subs(f3,x,lb)))  res = res + 1;  end      if (0 >= double(subs(f0,x,ub))) ~= (0 >= double(subs(f1,x,ub)))  res = res - 1;  end    if (0 >= double(subs(f1,x,ub))) ~= (0 >= double(subs(f2,x,ub)))  res = res - 1;  end    if (0 >= double(subs(f2,x,ub))) ~= (0 >= double(subs(f3,x,ub)))  res = res - 1;  end  %here we got a number of roots  end |

Данное уравнение имеет 2 реальных корня в промежутке от [-10,10]

1. Отделить все корни, лежащие на данном отрезке.  
   Отделение корней уравнения можно выполнить графически, либо используя итерационный алгоритм, приведенный нижe
2. Вычислить наименьший из корней сначала методом половинного деления, а за затем методом хорд и методом Ньютона. Сравнить число необходимых итераций в обоих методах. Точность до 0.0001. a=20,2374 b=-131,210 c=-843,923

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Метод | Бис. x1 | Бис. x2 | Хорд x1 | Хорд x2 | Ньютонх1 | Ньютонх2 |
| Итерации | 17 | 17 | 9 | 8 | 4 | 4 |

**Корни уравнения x = -4.240, 8.21901**

|  |
| --- |
| function res = BisectionMethod(f0,a,b,iter,e)  syms x  xn = a  xk = b;  xi = a;  dx = xk-xn  while(abs(xk-xn)>e)  iter = iter + 1  disp(iter)  dx = dx/2  xi = xn + dx  if(sign(subs(f0,x,xn)) ~= sign(subs(f0,x,xi)))  xk = xi  else  xn = xi  end  end  res = [xi,iter]  end |
| function res = ChordMethod(f0,a,b,iter,e)  %% for ) curve  syms x  fa = subs(f0,x,a);  fb = subs(f0,x,b);  newa = a - fa / (fb - fa) \* ( b - a )  iter = iter+1  if abs(newa-a) < e  res = [a,iter]  return  else  res = ChordMethod(f0,newa,b,iter,e)  return  end  end |
| function res = NewtonMethod(f0,a,b,iter,e)  syms x  fa0 = subs(f0,x,a);  fa1 = subs(diff(f0),x,a);  newa = a - fa0 / fa1  iter = iter+1  if abs(newa-a) < e  res = [a,iter]  return  else  res = NewtonMethod(f0,newa,b,iter,e)  return  end  end |

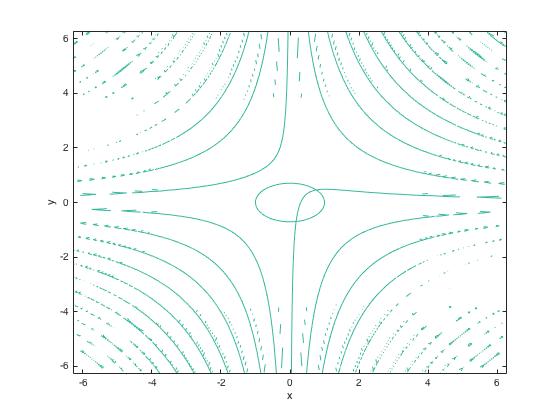
1. Решить систему нелинейных уравнений:

где x>0, y>0, с точностью до 0,0001 методами простых итераций и Ньютона, m = 0.3 a = 1,0

Начальные приближения найти графически. Сравнить скорость сходимости методов.

Функции для вычисления расположены в модуле SystemEQSolver.m

Так выглядит графическое решение системы

****

В качестве начальных приблежений выбрал x=0.4 , 0.7

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Число итераций | X1 | X2 |
| Простых итераций | 7 | 0.4602 | 0.7593 |
| Ньютона | 3 | 0.4602 | 0.7593 |

|  |
| --- |
| function res = SimpleIteration(x01,x02,gx1,gx2,e)  syms x1 x2 real    %%%% checking divergence  g1x1x1 = diff(gx1,x1)  g1x1x2 = diff(gx1,x2)  g1x2x1 = diff(gx2,x1)  g1x2x2 = diff(gx2,x2)    disp(double(subs(subs(g1x1x1,x1,x01),x2,x02)) + (double(subs(subs(g1x2x1,x1,x01),x2,x02))))  disp(double(subs(subs(g1x1x2,x1,x01),x2,x02)) + (double(subs(subs(g1x2x2,x1,x01),x2,x02))))  %%%%    x1c = x01+e+e  x2c = x02+e+e  x1n = x01;  x2n = x02;    iter = 0  while(abs(x1n-x1c)>e || abs(x2n-x2c)>e)  iter = iter + 1  x1c = x1n  x2c = x2n  x2n = double(subs(subs(gx2,x1,x1c),x2,x2c));  x1n = double(subs(subs(gx1,x1,x1c),x2,x2n));  end  res = [x1n,x2n,iter]  end |
| function res = NewtonMethod(x01,x02,f1,f2,e)  syms x1 x2 real    df1x1 = diff(f1,x1)  df1x2 = diff(f1,x2)  df2x1 = diff(f2,x1)  df2x2 = diff(f2,x2)    x1c = x01+e+e  x2c = x02+e+e  x1n = x01;  x2n = x02;    iter = 0  while(abs(x1n-x1c)>e || abs(x2n-x2c)>e)  iter=iter+1;  x1c = x1n  x2c = x2n  J=[double(subs(subs(df1x1,x1,x1c),x2,x2c)), double(subs(subs(df1x2,x1,x1c),x2,x2c));  double(subs(subs(df2x1,x1,x1c),x2,x2c)), double(subs(subs(df2x2,x1,x1c),x2,x2c))]^-1  f=[double(subs(subs(f1,x1,x1c),x2,x2c)), double(subs(subs(f2,x1,x1c),x2,x2c))]  fJ=J\*f';  x1n=x1c-fJ(1,1);  x2n=x2c-fJ(2,1);  end  res = [x1n,x2n,iter]  end |