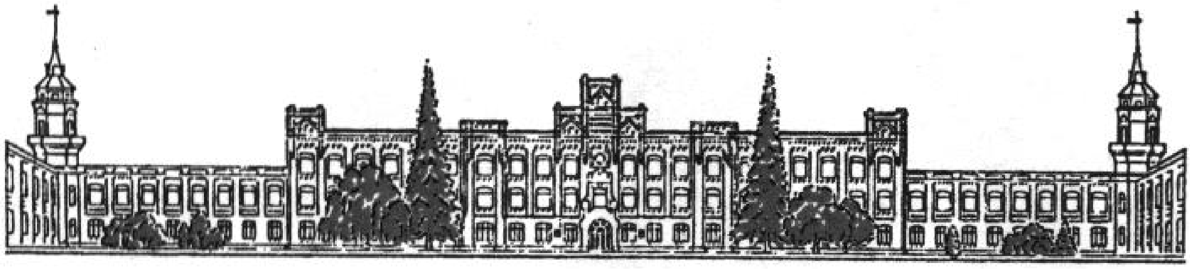

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
УКРАЇНИ«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Звіт

про виконання лабораторної роботи №1 з дисципліни
«Обчислювальна математика»

Виконав:
студент II курсу, групи КА-07
Москаленко Максим Геннадійович
Прийняла:

Київ — 2021

Принцип відокремлення коренів

Необхідно визначити корені рівняння:

$$4x^5 - 3x^4 + x^3 + 2x^2 - 4x + 7 = 0$$

Для початку необхідно визначити чи є корені даного рівняння дійсними. Для цього скористаємось теоремою Гюа. Дійсно кожен з коефіцієнтів задовольняє умову

$$a_k^2 > a_{k-1}a_{k+1} \quad (k = 1, 2 \ldots, n-1)$$

Далі визначимо кільце, в якому знаходяться дійсні корені рівняння за теоремою:

$$\frac{|a_n|}{a' + |a_n|} \leqslant |z| \leqslant 1 + \frac{a}{|a_0|}, \text{ де}$$

$$a = \max\{|a_1|,\ |a_2|,\ \ldots\}$$
$$a' = \max\{|a_0|, |a_1| \ldots |a_{n-1}|\}$$

Отримуємо розв'язок:

$$x \in [-2.75;\ -0.64] \cup [0.64;\ 2.75]$$

Тепер скористаємось теоремою Штурмана для визначення к-ть коренів на проміжку, і виходячи з цього проміжки, на яких наявний лише один корінь. Визначимо множину функцій:

$$f(x) = 4x^5 - 3x^4 + x^3 + 2x^2 - 4x + 7$$

$$f_1(x) = f'(x) = 20x^4 - 12x^3 + 3x^2 + 4x - 4$$

$$f_2(x) = -\frac{f(x)}{f_1(x)} = -0.04x^3 - 1.29x^2 + 3.08x - 6.88$$

$$f_3(x) = -\frac{f_1(x)}{f_2(x)} = -22\,731.25x^2 - 54\,025x + 113\,000$$

$$f_4(x) = -\frac{f_2(x)}{f_3(x)} = 0.013x - 0.0054$$

$$f_5(x) = -\frac{f_3(x)}{f_4(x)} = -94\,641$$

Аналізуємо додатній проміжок та визначаємо знаки значень кожної з функцій.

Точка 0.64:

$f(0.64) > 0$	$f_1(0.64) < 0$	$f_2(0.64) < 0$
$f_3(0.64) > 0$	$f_4(0.64) > 0$	$f_5(0.64) < 0$

Кількість знакозмін: 3

Точка 2.75:

$f(2.75) > 0$	$f_1(2.75) > 0$	$f_2(2.75) < 0$
$f_3(2.75) < 0$	$f_4(2.75) > 0$	$f_5(2.75) < 0$

Кількість знакозмін: 3

Отже, кількість коренів на проміжку $[0.64;\ 2.75]$ дорівнює 0

Аналізуємо від'ємний проміжок та визначаємо знаки значень кожної з функцій.

Точка -0.64:

$f(-0.64) > 0$	$f_1(-0.64) > 0$	$f_2(-0.64) < 0$
$f_3(-0.64) > 0$	$f_4(-0.64) < 0$	$f_5(-0.64) < 0$

Кількість знакозмін: 3

Точка -2.75:

$f(-\ 2.75)\ <\ 0$ $f_1(-\ 2.75)\ >0$ $f_2(-\ 2.75)\ <\ 0$

$f_3(-\ 2.75)\ >0$ $f_4(-\ 2.75)\ <0$ $f_5(-\ 2.75)\ <0$

Кількість знакозмін: 4

Отже, кількість коренів на проміжку $[-\ 2.75;\ 0.64]$ дорівнює 1

Таким чином дане рівняння має 1 корінь, що належить проміжку $[-\ 2.75;\ \ -\ 0.64]$

Текст програми (C++)

```
#include <iostream>
#include <cmath>

using namespace std;

struct Response
{
   double result;
   string method;
   int number_of_iterations;
   Response(double result, string method, int number_of_iterations)
   {
       this->result = result;
       this->method = method;
       this->number_of_iterations = number_of_iterations;
   }
   void print()
   {
       printf("Result %lf was gained by %s method in %i iterations\n", result, method.c_str(),
number_of_iterations);
   }
};

struct Polinom
{
   double lb, rb, eps;
   Polinom(double l_border, double r_border, double epsilon)
   {
       this->lb = l_border;
       this->rb = r_border;
       this->eps = epsilon;
   }
```

```
  double f(double x)
  {
    return 4 * pow(x, 5) - 3 * pow(x, 4) + pow(x, 3) + 2 * pow(x, 2) - 4 * x + 7;
  }
  double df(double x)
  {
    return 20 * pow(x, 4) - 12 * pow(x, 3) + 3 * pow(x, 2) + 4 * x - 4;
  }
  double ddf(double x)
  {
    return 80 * pow(x, 4) - 36 * pow(x, 2) + 6 * x + 4;
  }
  Response *bisection_method()
  {
    cout << "\nBISECTION METHOD\n\n";
    double left = lb;
    double right = rb;
    double center = (lb + rb) / 2;
    int iter_num = 0;
    printf("B Iter: %d (a=%lf f(a)=%lf); (b=%lf f(b)=%lf)\n", iter_num, left, f(left), right,
f(right));

    while (fabs(right - left) >= eps && fabs(f(center)) >= eps)
    {
      iter_num++;
      center = (left + right) / 2;
      double y = f(center);
      if (y > 0)
        right = center;
      if (y < 0)
        left = center;
      if (y == 0)
        break;

      printf("B Iter: %d (a=%lf f(a)=%lf); (b=%lf f(b)=%lf)\n", iter_num, left, f(left), right,
f(right));
    }
    return new Response(center, "bisection", iter_num);
  }

  Response *chord_method()
  {
    cout << "\nCHORD METHOD\n\n";

    double left = lb;
    double right = rb;
    double cur_x = lb;
    int iter_num = 0;
    double prev_x;
    printf("C Iter: %d (a=%lf f(a)=%lf); (b=%lf f(b)=%lf)\n", iter_num, left, f(left), right,
f(right));

    do
    {
```

```
        iter_num++;
        prev_x = cur_x;
        cur_x = (left * f(right) - right * f(left)) / (f(right) - f(left));
        double y = f(cur_x);
        if (y > 0)
          right = cur_x;
        if (y < 0)
          left = cur_x;
        if (y == 0)
          break;

        printf("C Iter: %d (a=%lf f(a)=%lf); (b=%lf f(b)=%lf)\n", iter_num, left, f(left), right,
f(right));

      } while (fabs(cur_x - prev_x) >= eps && fabs(f(cur_x)) >= eps);

      return new Response(cur_x, "chord", iter_num);
    }

    Response *newton_method()
    {
      cout << "\nNEWTON METHOD\n\n";

      double prev_x = f(lb) * ddf(lb) > 0 ? lb : rb;
      double cur_x = prev_x - (f(prev_x) / df(prev_x));
      int iter_num = 0;
      printf("N Iter: x%d=%lf f(x%d)=%lf\n", iter_num, cur_x, iter_num, f(cur_x));

      while (abs(cur_x - prev_x) >= eps && abs(f(cur_x)) >= eps)
      {
        iter_num++;
        prev_x = cur_x;
        cur_x = prev_x - (f(prev_x) / df(prev_x));
        printf("N Iter: x%d=%lf f(x%d)=%lf\n", iter_num, cur_x, iter_num, f(cur_x));
      }
      return new Response(cur_x, "newton", iter_num);
    }
};

int main(int argc, char const *argv[])
{
    if (argc < 4)
    {
      cout << "There are no enough arguments (you need two borders and precision)\n";
      return 1;
    }

    Polinom *p;

    if (stod(argv[1]) > stod(argv[2]))
      p = new Polinom(stod(argv[2]), stod(argv[1]), stod(argv[3]));
    else
      p = new Polinom(stod(argv[1]), stod(argv[2]), stod(argv[3]));
```

```
  if (p->f(stod(argv[1])) * p->f(stod(argv[2])) > 0)
  {
     cout << "This borders are invalid (0 or >1 root)\n";
     return 1;
  }

  p->bisection_method()->print();
  p->chord_method()->print();
  p->newton_method()->print();

  return 0;
}
```

Результат роботи програми:

BISECTION METHOD

B Iter: 0 (a=-2.750000 f(a)=-788.351562); (b=-0.640000 f(b)=9.184243)
B Iter: 1 (a=-1.695000 f(a)=-66.070514); (b=-0.640000 f(b)=9.184243)
B Iter: 2 (a=-1.167500 f(a)=-1.445520); (b=-0.640000 f(b)=9.184243)
B Iter: 3 (a=-1.167500 f(a)=-1.445520); (b=-0.903750 f(b)=7.097488)
B Iter: 4 (a=-1.167500 f(a)=-1.445520); (b=-1.035625 f(b)=3.960812)
B Iter: 5 (a=-1.167500 f(a)=-1.445520); (b=-1.101562 f(b)=1.591217)
B Iter: 6 (a=-1.167500 f(a)=-1.445520); (b=-1.134531 f(b)=0.163079)
B Iter: 7 (a=-1.151016 f(a)=-0.617769); (b=-1.134531 f(b)=0.163079)
B Iter: 8 (a=-1.142773 f(a)=-0.221596); (b=-1.134531 f(b)=0.163079)
B Iter: 9 (a=-1.138652 f(a)=-0.027835); (b=-1.134531 f(b)=0.163079)
B Iter: 10 (a=-1.138652 f(a)=-0.027835); (b=-1.136592 f(b)=0.067976)
B Iter: 11 (a=-1.138652 f(a)=-0.027835); (b=-1.137622 f(b)=0.020159)
B Iter: 12 (a=-1.138137 f(a)=-0.003816); (b=-1.137622 f(b)=0.020159)
B Iter: 13 (a=-1.138137 f(a)=-0.003816); (b=-1.137880 f(b)=0.008177)
B Iter: 14 (a=-1.138137 f(a)=-0.003816); (b=-1.138008 f(b)=0.002182)
B Iter: 15 (a=-1.138073 f(a)=-0.000816); (b=-1.138008 f(b)=0.002182)
B Iter: 16 (a=-1.138073 f(a)=-0.000816); (b=-1.138041 f(b)=0.000683)
B Iter: 17 (a=-1.138057 f(a)=-0.000067); (b=-1.138041 f(b)=0.000683)
B Iter: 18 (a=-1.138057 f(a)=-0.000067); (b=-1.138049 f(b)=0.000308)
Result -1.138049 was gained by bisection method in 18 iterations

CHORD METHOD

C Iter: 0 (a=-2.750000 f(a)=-788.351562); (b=-0.640000 f(b)=9.184243)
C Iter: 1 (a=-2.750000 f(a)=-788.351562); (b=-0.664298 f(b)=9.144953)
C Iter: 2 (a=-2.750000 f(a)=-788.351562); (b=-0.688215 f(b)=9.083607)
C Iter: 3 (a=-2.750000 f(a)=-788.351562); (b=-0.711701 f(b)=8.999290)
C Iter: 4 (a=-2.750000 f(a)=-788.351562); (b=-0.734706 f(b)=8.891386)

C Iter: 5 (a=-2.750000 f(a)=-788.351562); (b=-0.757182 f(b)=8.759616)
C Iter: 6 (a=-2.750000 f(a)=-788.351562); (b=-0.779082 f(b)=8.604067)
C Iter: 7 (a=-2.750000 f(a)=-788.351562); (b=-0.800360 f(b)=8.425216)
C Iter: 8 (a=-2.750000 f(a)=-788.351562); (b=-0.820976 f(b)=8.223930)
C Iter: 9 (a=-2.750000 f(a)=-788.351562); (b=-0.840891 f(b)=8.001465)
C Iter: 10 (a=-2.750000 f(a)=-788.351562); (b=-0.860073 f(b)=7.759434)
C Iter: 11 (a=-2.750000 f(a)=-788.351562); (b=-0.878494 f(b)=7.499776)
C Iter: 12 (a=-2.750000 f(a)=-788.351562); (b=-0.896130 f(b)=7.224705)
C Iter: 13 (a=-2.750000 f(a)=-788.351562); (b=-0.912965 f(b)=6.936654)
C Iter: 14 (a=-2.750000 f(a)=-788.351562); (b=-0.928988 f(b)=6.638202)
C Iter: 15 (a=-2.750000 f(a)=-788.351562); (b=-0.944194 f(b)=6.332014)
C Iter: 16 (a=-2.750000 f(a)=-788.351562); (b=-0.958582 f(b)=6.020766)
C Iter: 17 (a=-2.750000 f(a)=-788.351562); (b=-0.972160 f(b)=5.707082)
C Iter: 18 (a=-2.750000 f(a)=-788.351562); (b=-0.984938 f(b)=5.393476)
C Iter: 19 (a=-2.750000 f(a)=-788.351562); (b=-0.996931 f(b)=5.082302)
C Iter: 20 (a=-2.750000 f(a)=-788.351562); (b=-1.008160 f(b)=4.775710)
C Iter: 21 (a=-2.750000 f(a)=-788.351562); (b=-1.018649 f(b)=4.475620)
C Iter: 22 (a=-2.750000 f(a)=-788.351562); (b=-1.028422 f(b)=4.183700)
C Iter: 23 (a=-2.750000 f(a)=-788.351562); (b=-1.037510 f(b)=3.901360)
C Iter: 24 (a=-2.750000 f(a)=-788.351562); (b=-1.045943 f(b)=3.629748)
C Iter: 25 (a=-2.750000 f(a)=-788.351562); (b=-1.053753 f(b)=3.369760)
C Iter: 26 (a=-2.750000 f(a)=-788.351562); (b=-1.060973 f(b)=3.122051)
C Iter: 27 (a=-2.750000 f(a)=-788.351562); (b=-1.067635 f(b)=2.887058)
C Iter: 28 (a=-2.750000 f(a)=-788.351562); (b=-1.073774 f(b)=2.665015)
C Iter: 29 (a=-2.750000 f(a)=-788.351562); (b=-1.079421 f(b)=2.455984)
C Iter: 30 (a=-2.750000 f(a)=-788.351562); (b=-1.084610 f(b)=2.259870)
C Iter: 31 (a=-2.750000 f(a)=-788.351562); (b=-1.089370 f(b)=2.076455)
C Iter: 32 (a=-2.750000 f(a)=-788.351562); (b=-1.093732 f(b)=1.905411)
C Iter: 33 (a=-2.750000 f(a)=-788.351562); (b=-1.097726 f(b)=1.746328)
C Iter: 34 (a=-2.750000 f(a)=-788.351562); (b=-1.101378 f(b)=1.598730)
C Iter: 35 (a=-2.750000 f(a)=-788.351562); (b=-1.104714 f(b)=1.462096)
C Iter: 36 (a=-2.750000 f(a)=-788.351562); (b=-1.107760 f(b)=1.335870)
C Iter: 37 (a=-2.750000 f(a)=-788.351562); (b=-1.110538 f(b)=1.219478)
C Iter: 38 (a=-2.750000 f(a)=-788.351562); (b=-1.113070 f(b)=1.112338)
C Iter: 39 (a=-2.750000 f(a)=-788.351562); (b=-1.115377 f(b)=1.013870)
C Iter: 40 (a=-2.750000 f(a)=-788.351562); (b=-1.117476 f(b)=0.923501)
C Iter: 41 (a=-2.750000 f(a)=-788.351562); (b=-1.119387 f(b)=0.840673)
C Iter: 42 (a=-2.750000 f(a)=-788.351562); (b=-1.121123 f(b)=0.764848)
C Iter: 43 (a=-2.750000 f(a)=-788.351562); (b=-1.122702 f(b)=0.695507)
C Iter: 44 (a=-2.750000 f(a)=-788.351562); (b=-1.124137 f(b)=0.632161)
C Iter: 45 (a=-2.750000 f(a)=-788.351562); (b=-1.125439 f(b)=0.574341)
C Iter: 46 (a=-2.750000 f(a)=-788.351562); (b=-1.126622 f(b)=0.521609)

C Iter: 47 (a=-2.750000 f(a)=-788.351562); (b=-1.127695 f(b)=0.473553)
C Iter: 48 (a=-2.750000 f(a)=-788.351562); (b=-1.128669 f(b)=0.429788)
C Iter: 49 (a=-2.750000 f(a)=-788.351562); (b=-1.129553 f(b)=0.389956)
C Iter: 50 (a=-2.750000 f(a)=-788.351562); (b=-1.130354 f(b)=0.353722)
C Iter: 51 (a=-2.750000 f(a)=-788.351562); (b=-1.131080 f(b)=0.320778)
C Iter: 52 (a=-2.750000 f(a)=-788.351562); (b=-1.131739 f(b)=0.290840)
C Iter: 53 (a=-2.750000 f(a)=-788.351562); (b=-1.132336 f(b)=0.263644)
C Iter: 54 (a=-2.750000 f(a)=-788.351562); (b=-1.132876 f(b)=0.238948)
C Iter: 55 (a=-2.750000 f(a)=-788.351562); (b=-1.133366 f(b)=0.216531)
C Iter: 56 (a=-2.750000 f(a)=-788.351562); (b=-1.133810 f(b)=0.196188)
C Iter: 57 (a=-2.750000 f(a)=-788.351562); (b=-1.134212 f(b)=0.177733)
C Iter: 58 (a=-2.750000 f(a)=-788.351562); (b=-1.134577 f(b)=0.160995)
C Iter: 59 (a=-2.750000 f(a)=-788.351562); (b=-1.134906 f(b)=0.145817)
C Iter: 60 (a=-2.750000 f(a)=-788.351562); (b=-1.135205 f(b)=0.132057)
C Iter: 61 (a=-2.750000 f(a)=-788.351562); (b=-1.135476 f(b)=0.119584)
C Iter: 62 (a=-2.750000 f(a)=-788.351562); (b=-1.135720 f(b)=0.108281)
C Iter: 63 (a=-2.750000 f(a)=-788.351562); (b=-1.135942 f(b)=0.098039)
C Iter: 64 (a=-2.750000 f(a)=-788.351562); (b=-1.136143 f(b)=0.088760)
C Iter: 65 (a=-2.750000 f(a)=-788.351562); (b=-1.136324 f(b)=0.080354)
C Iter: 66 (a=-2.750000 f(a)=-788.351562); (b=-1.136489 f(b)=0.072740)
C Iter: 67 (a=-2.750000 f(a)=-788.351562); (b=-1.136638 f(b)=0.065845)
C Iter: 68 (a=-2.750000 f(a)=-788.351562); (b=-1.136773 f(b)=0.059600)
C Iter: 69 (a=-2.750000 f(a)=-788.351562); (b=-1.136894 f(b)=0.053946)
C Iter: 70 (a=-2.750000 f(a)=-788.351562); (b=-1.137005 f(b)=0.048826)
C Iter: 71 (a=-2.750000 f(a)=-788.351562); (b=-1.137105 f(b)=0.044191)
C Iter: 72 (a=-2.750000 f(a)=-788.351562); (b=-1.137195 f(b)=0.039994)
C Iter: 73 (a=-2.750000 f(a)=-788.351562); (b=-1.137277 f(b)=0.036195)
C Iter: 74 (a=-2.750000 f(a)=-788.351562); (b=-1.137351 f(b)=0.032756)
C Iter: 75 (a=-2.750000 f(a)=-788.351562); (b=-1.137418 f(b)=0.029644)
C Iter: 76 (a=-2.750000 f(a)=-788.351562); (b=-1.137479 f(b)=0.026826)
C Iter: 77 (a=-2.750000 f(a)=-788.351562); (b=-1.137534 f(b)=0.024276)
C Iter: 78 (a=-2.750000 f(a)=-788.351562); (b=-1.137583 f(b)=0.021968)
C Iter: 79 (a=-2.750000 f(a)=-788.351562); (b=-1.137628 f(b)=0.019879)
C Iter: 80 (a=-2.750000 f(a)=-788.351562); (b=-1.137669 f(b)=0.017988)
C Iter: 81 (a=-2.750000 f(a)=-788.351562); (b=-1.137706 f(b)=0.016277)
C Iter: 82 (a=-2.750000 f(a)=-788.351562); (b=-1.137739 f(b)=0.014729)
C Iter: 83 (a=-2.750000 f(a)=-788.351562); (b=-1.137769 f(b)=0.013327)
C Iter: 84 (a=-2.750000 f(a)=-788.351562); (b=-1.137796 f(b)=0.012059)
C Iter: 85 (a=-2.750000 f(a)=-788.351562); (b=-1.137821 f(b)=0.010912)
C Iter: 86 (a=-2.750000 f(a)=-788.351562); (b=-1.137843 f(b)=0.009874)
C Iter: 87 (a=-2.750000 f(a)=-788.351562); (b=-1.137863 f(b)=0.008934)
C Iter: 88 (a=-2.750000 f(a)=-788.351562); (b=-1.137882 f(b)=0.008084)

C Iter: 89 (a=-2.750000 f(a)=-788.351562); (b=-1.137898 f(b)=0.007314)
C Iter: 90 (a=-2.750000 f(a)=-788.351562); (b=-1.137913 f(b)=0.006618)
C Iter: 91 (a=-2.750000 f(a)=-788.351562); (b=-1.137927 f(b)=0.005988)
C Iter: 92 (a=-2.750000 f(a)=-788.351562); (b=-1.137939 f(b)=0.005418)
C Iter: 93 (a=-2.750000 f(a)=-788.351562); (b=-1.137950 f(b)=0.004902)
C Iter: 94 (a=-2.750000 f(a)=-788.351562); (b=-1.137960 f(b)=0.004436)
C Iter: 95 (a=-2.750000 f(a)=-788.351562); (b=-1.137969 f(b)=0.004014)
Result -1.137969 was gained by chord method in 95 iterations

NEWTON METHOD

N Iter: x0=-8.489973 f(x0)=-192451.063523
N Iter: x1=-6.762890 f(x1)=-63046.957862
N Iter: x2=-5.381940 f(x2)=-20647.914289
N Iter: x3=-4.278469 f(x3)=-6757.432693
N Iter: x4=-3.398057 f(x4)=-2207.766477
N Iter: x5=-2.698169 f(x5)=-718.303255
N Iter: x6=-2.146801 f(x6)=-231.208763
N Iter: x7=-1.722394 f(x7)=-72.324569
N Iter: x8=-1.415082 f(x8)=-20.894908
N Iter: x9=-1.226069 f(x9)=-4.793993
N Iter: x10=-1.149874 f(x10)=-0.562217
N Iter: x11=-1.138300 f(x11)=-0.011424
N Iter: x12=-1.138055 f(x12)=-0.000005
Result -1.138055 was gained by newton method in 12 iterations

Висновок

Найкраще себе показав метод Ньютона (та все ж не так добре, як хотілося б), а найгірше метод хорд. Це можна пояснити тим, що на досліджуваному проміжку функція дуже різко зростає і тому її дотичні у точках цього проміжку мають вигляд вертикальних прямих, а отже зсув по вісі X не значний. Тому необхідно багато ітерацій аби цей зсув став значний та наблизив проміжок до шуканої точки (розв’язку рівняння). На метод бісекцій дана особливість не вплинула, так як він не залежить від дотичних, а лише від ширини початкового проміжку.