**МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

**Кафедра программного обеспечения вычислительной техники и автоматизированных систем**

Лабораторная работа №6

Дисциплина: Вычислительная математика

по теме «Одномерная минимизация функции»

Выполнил: ст. группы ВТ-22  
Макаров Даниил Сергеевич

Проверила: Бондаренко Т.В.

**Белгород 2018**

**Лабораторная работа №6**

**Одномерная минимизация функции**

**Цель работы:** изучить методы нахождения приближенного решения задачи одномерной минимизации функции одной переменной, и получить практические навыки их применения.

**Задания к работе**

1. Найти область определения заданной функции у = f(x) и построить её график, используя равномерную сетку значений хi (шаг сетки выбрать самостоятельно).

2. Найти промежутки унимодальности функции у = f(x), используя построенный график.

3. Найти первую y´=f´(x) и вторую y´´= f´´ (x) производные заданной функции у = f(x).

4. Найти точное решение задачи одномерной минимизации ― минимум функции у = f(x), точку х Т, и минимальное значение функции 𝑚𝑖𝑛(𝑓(𝑥 Т )).

5. Найти приближенное решение задачи одномерной минимизации, точку х̃ такую, что 𝑥 Т ≈ х̃ вручную, используя численные методы одномерной минимизации:

− метод оптимального поиска;

− метод, основанный на использовании чисел Фибоначчи;

− метод деления отрезка пополам; с точностью ε =0,01.

Необходимые параметры методов выбрать самостоятельно. Подробно «вручную» достаточно выполнить только первый шаг численного метода решения. Окончательный результат вычислений может быть получен с помощью приложения MS Excel.

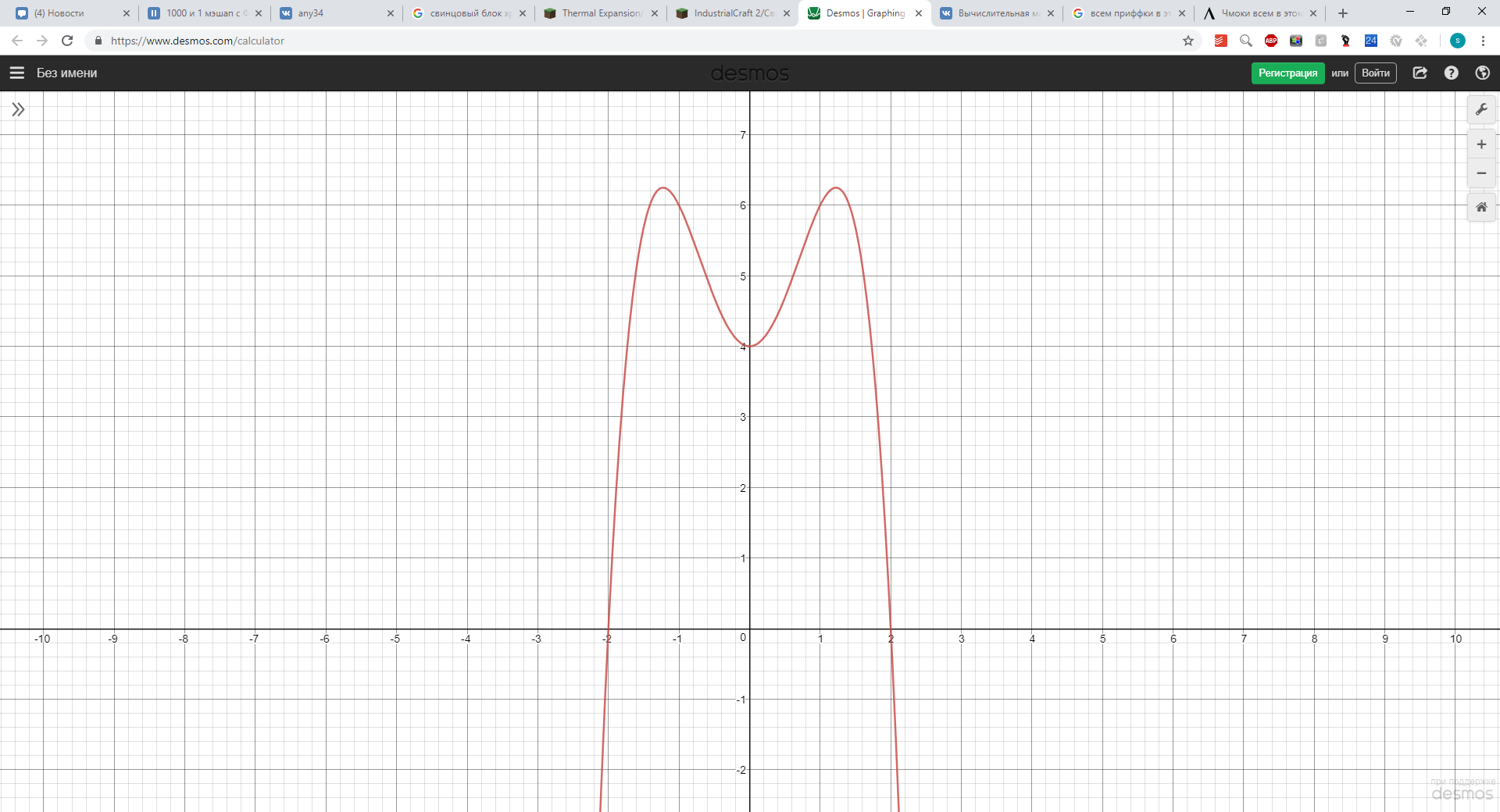
6. Определить абсолютную Δ и относительную δ погрешность решения задачи одномерной минимизации для каждого из используемых численных методов. Представить полученные результаты в виде таблицы (см. табл. 6.1).

7. Описать в модуле функции, которые возвращают приближенные значения минимума функции у = f(x) для заданного промежутка унимодальности X ⊂ R с заданной точностью ε каждым из рассмотренных численных методов: метод оптимального поиска; метод, основанный на использовании чисел Фибоначчи; метод деления отрезка пополам.

8. Составить программу для вычисления приближенного решения задачи одномерной минимизации для заданного варианта задания с использованием функций, описанных в модуле.

**Вариант 7**

Область определения функции принадлежит R



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **y** | **-1** | **-0.8** | **-0.6** | **-0.4** | **-0.2** | **0** | **0.2** | **0.4** | **0.6** | **0.8** |
| **x** | **6** | **5.51** | **4.95** | **4.454** | **4.118** | **4** | **4.118** | **4.454** | **4.95** | **5.51** |

Промежуток унимодальности (-1.225;1.225)

**Метод оптимального поиска**

=122

|  |  |  |
| --- | --- | --- |
| i | x | y |
| 0 | -0,3 | 4,2619 |
| 1 | -0,29 | 4,245227 |
| 2 | -0,28 | 4,229053 |
| 3 | -0,27 | 4,213386 |
| 4 | -0,26 | 4,19823 |
| 5 | -0,25 | 4,183594 |
| 6 | -0,24 | 4,169482 |
| 7 | -0,23 | 4,155902 |
| 8 | -0,22 | 4,142857 |
| 9 | -0,21 | 4,130355 |
| 10 | -0,2 | 4,1184 |
| 11 | -0,19 | 4,106997 |
| 12 | -0,18 | 4,09615 |
| 13 | -0,17 | 4,085865 |
| 14 | -0,16 | 4,076145 |
| 15 | -0,15 | 4,066994 |
| 16 | -0,14 | 4,058416 |
| 17 | -0,13 | 4,050414 |
| 18 | -0,12 | 4,042993 |
| 19 | -0,11 | 4,036154 |
| 20 | -0,1 | 4,0299 |
| 21 | -0,09 | 4,024234 |
| 22 | -0,08 | 4,019159 |
| 23 | -0,07 | 4,014676 |
| 24 | -0,06 | 4,010787 |
| 25 | -0,05 | 4,007494 |
| 26 | -0,04 | 4,004797 |
| 27 | -0,03 | 4,002699 |
| 28 | -0,02 | 4,0012 |
| 29 | -0,01 | 4,0003 |
| 30 | 0 | 4 |
| 31 | 0,01 | 4,0003 |
| 32 | 0,02 | 4,0012 |
| 33 | 0,03 | 4,002699 |
| 34 | 0,04 | 4,004797 |
| 35 | 0,05 | 4,007494 |
| 36 | 0,06 | 4,010787 |
| 37 | 0,07 | 4,014676 |
| 38 | 0,08 | 4,019159 |
| 39 | 0,09 | 4,024234 |
| 40 | 0,1 | 4,0299 |
| 41 | 0,11 | 4,036154 |
| 42 | 0,12 | 4,042993 |
| 43 | 0,13 | 4,050414 |
| 44 | 0,14 | 4,058416 |
| 45 | 0,15 | 4,066994 |
| 46 | 0,16 | 4,076145 |
| 47 | 0,17 | 4,085865 |
| 48 | 0,18 | 4,09615 |
| 49 | 0,19 | 4,106997 |
| 50 | 0,2 | 4,1184 |
| 51 | 0,21 | 4,130355 |
| 52 | 0,22 | 4,142857 |
| 53 | 0,23 | 4,155902 |
| 54 | 0,24 | 4,169482 |
| 55 | 0,25 | 4,183594 |
| 56 | 0,26 | 4,19823 |
| 57 | 0,27 | 4,213386 |
| 58 | 0,28 | 4,229053 |
| 59 | 0,29 | 4,245227 |
| 60 | 0,3 | 4,2619 |

**Метод деления отрезка пополам**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i | a | b | α | β | f(α) | f(β) |
| 0 | -1,225 | 1,225 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 1 | -1,225 | 0,001 | -0,613 | -0,611 | 4,986105 | 4,9805944 |
| 2 | -0,613 | 0,001 | -0,307 | -0,305 | 4,273864 | 4,2704213 |
| 3 | -0,307 | 0,001 | -0,154 | -0,152 | 4,070586 | 4,0687782 |
| 4 | -0,154 | 0,001 | -0,078 | -0,076 | 4,017983 | 4,0170683 |
| 5 | -0,0775 | 0,001 | -0,039 | -0,037 | 4,004619 | 4,0041608 |
| 6 | -0,03925 | 0,001 | -0,02 | -0,018 | 4,001215 | 4,0009854 |
| 7 | -0,02013 | 0,001 | -0,011 | -0,009 | 4,000335 | 4,0002199 |
| 8 | -0,01056 | 0,001 | -0,006 | -0,004 | 4,0001 | 4,0000429 |
| 9 | -0,00578 | 0,001 | -0,003 | -0,001 | 4,000034 | 4,0000058 |
| 10 | -0,00339 | 0,001 | -0,002 | -2E-04 | 4,000014 | 4,0000001 |
| 11 | -0,0022 | 0,001 | -0,002 | 0,0004 | 4,000008 | 4,0000005 |
| 12 | -0,0016 | 0,001 | -0,001 | 0,0007 | 4,000005 | 4,0000015 |
| 13 | -0,0013 | 0,001 | -0,001 | 0,0009 | 4,000004 | 4,0000022 |
| 14 | -0,00115 | 0,001 | -0,001 | 0,0009 | 4,000003 | 4,0000026 |
| 15 | -0,00107 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,0000028 |
| 16 | -0,00104 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,0000029 |
| 17 | -0,00102 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,0000029 |
| 18 | -0,00101 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 19 | -0,001 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 20 | -0,001 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 21 | -0,001 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 22 | -0,001 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 23 | -0,001 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 24 | -0,001 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 25 | -0,001 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 26 | -0,001 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 27 | -0,001 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 28 | -0,001 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 29 | -0,001 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 30 | -0,001 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 31 | -0,001 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 32 | -0,001 | 0,001 | -0,001 | 0,001 | 4,000003 | 4,000003 |
| 33 | 0,001 | 0,001 | 0 | 0,002 | 4 | 4,000012 |
|  | x min | 0,001 |  |  |  |  |

**#include <stdio.h>**

**#include <stdlib.h>**

**const double eps = 0.0001;**

**typedef double(\*func)(double x);**

**double search\_method(func f,double a,double b){**

**int step\_count = (b - a)/eps - 1;**

**double x[step\_count+2], y[step\_count+2], min, x\_min;**

**x[0] = a;**

**for (int i = 1; i <= step\_count + 1; i++)**

**x[i] = x[i-1] + eps;**

**min = f(x[0]); x\_min = x[0];**

**for (int i = 0; i <= step\_count + 1; i++){**

**y[i] = f(x[i]);**

**if (y[i] < min){**

**min = y[i];**

**x\_min = x[i];**

**}**

**}**

**return x\_min;**

**}**

**double fibb\_method(func f,double a,double b){**

**int step\_count = (b - a)/eps - 1;**

**double fibb\_row[step\_count+2], min, a\_k = a, b\_k = b, a1, b1;**

**fibb\_row[0] = 1; fibb\_row[1] = 1;**

**for (int i = 2; i <= step\_count + 1; i++)**

**fibb\_row[i] = fibb\_row[i-1] + fibb\_row[i-2];**

**for (int k = 0; k <= step\_count - 2; k++){**

**a1 = a\_k + fibb\_row[step\_count-k-1]/fibb\_row[step\_count-k+1]\*(b\_k - a\_k);**

**b1 = a\_k + fibb\_row[step\_count-k]/fibb\_row[step\_count-k+1]\*(b\_k - a\_k);**

**if (f(a1) <= f(b1)) b\_k = b1;**

**else a\_k = a1;**

**}**

**if (f(a1) <= f(b1)) min = a1;**

**else min = b1;**

**return min;**

**}**

**double div\_method(func f,double a,double b){**

**int step\_count = (b - a)/eps - 1;**

**double min, a\_k = a, b\_k = b, a1, b1;**

**for (int k = 0; k <= step\_count; k++){**

**a1 = (a\_k + b\_k)/2 - eps;**

**b1 = (a\_k + b\_k)/2 + eps;**

**if (f(a1) <= f(b1)) b\_k = b1;**

**else a\_k = a1;**

**}**

**min = (b\_k + a\_k)/2;**

**return min;**

**}**

**double f1(double x){**

**return (x\*x+1)\*(x\*x-4);**

**}**

**int main(){**

**double min,a=-0.3,b=0.3;**

**min=search\_method(f1,a,b);**

**printf("Optimal search method result = %f\n",min);**

**min=fibb\_method(f1,a,b);**

**printf("Fibonacci search method result = %f\n",min);**

**min=div\_method(f1,a,b);**

**printf("Division method result = %f\n",min);**

**return 1;**

**}**