МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Воронежский государственный технический университет»

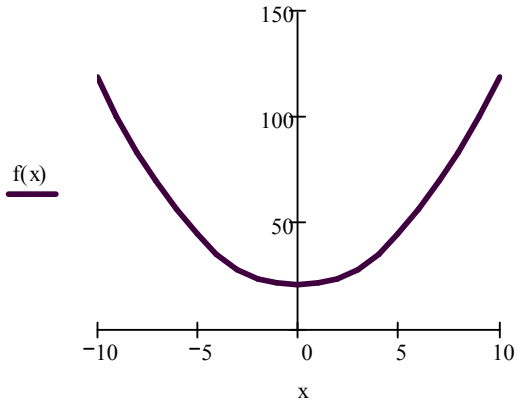
Кафедра автоматизированных и вычислительных систем

331 - 2021

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

к выполнению лабораторных работ
по дисциплине «Методы оптимизации»
для студентов направления подготовки 09.03.01
«Информатика и вычислительная техника» (профиль
«Вычислительные машины, комплексы, системы и сети»)
заочной формы обучения



Воронеж 2021

УДК 681.3.06(07)
ББК 32.973

Составители: канд. техн. наук Т. И. Сергеева,
канд. техн. наук М. Ю. Сергеев,
канд. техн. наук Т. Н. Недикова

Программная реализация оптимизационных задач: методические рекомендации к выполнению лабораторных работ по дисциплине «Методы оптимизации» для студентов направления подготовки 09.03.01 «Информатика и вычислительная техника» (профиль «Вычислительные машины, комплексы, системы и сети») заочной формы обучения / ФГБОУ ВО «Воронежский государственный технический университет; сост.: Т. И. Сергеева, М. Ю. Сергеев, Т. Н. Недикова. Воронеж: Изд-во ВГТУ, 2021. 30 с.

Цель методических рекомендаций – получение умений и навыков разработки программ, реализующих оптимизационные вычислительные задачи.

Методические рекомендации содержат теоретические сведения и практические задания для решения нелинейных оптимизационных задач.

Предназначены для проведения лабораторных работ по дисциплине «Методы оптимизации» для студентов 4 курса заочной формы обучения.

Методические рекомендации подготовлены в электронном виде и содержатся в файле MO_ZO_LR.pdf.

Ил. 9. Табл. 3. Библиогр.: 7 назв.

УДК 681.3.06(07)
ББК 32.973

Рецензент - В. В. Сафронов, канд. техн. наук, доцент кафедры автоматизированных и вычислительных систем ВГТУ

Издается по решению редакционно-издательского совета
Воронежского государственного технического университета

ВВЕДЕНИЕ

При выполнении лабораторных работ студенты должны получить умения и навыки разработки и отладки программ, реализующих решение вычислительных оптимизационных задач.

Методические указания содержат три лабораторных работы по решению нелинейных оптимизационных задач с функцией цели, зависящей от одной или нескольких переменных без ограничений.

Выбор номера варианта лабораторного задания совпадает с порядковым номером студента в списке группы.

1. МЕТОДЫ ПОИСКА ЭКСТРЕМУМА ДЛЯ ФУНКЦИЙ ОДНОЙ ПЕРЕМЕННОЙ БЕЗ ОГРАНИЧЕНИЙ

1.1. Методы одномерной оптимизации

Постановка задачи. Требуется найти безусловный минимум функции f(x) одной переменной, т.е. такую точку $x^* \in R$, что

$$f(x^*) = \min f(x),\ x^* \in R \qquad (1)$$

Для методов одномерной минимизации типично задание начального интервала неопределенности $L_0 = [a_0,b_0]$, в котором предположительно находится точка минимума x^*, но ее точное значение неизвестно.

Большинство известных методов одномерной минимизации применяется для класса унимодальной функции.

Определение. Функция f(x) называется унимодальной на интервале $L_0 = [a_0,b_0]$, если она достигает глобального минимума на $[a_0,b_0]$ в единственной точке x^*, причем слева от x^* эта функция строго убывает, а справа от x^* строго возрастает. Если $a_0 \le y < z < x^*$, то $f(y) > f(z)$, а если $x^* < y < z \le b_0$, то $f(y) < f(z)$.

В большинстве методов решения строят точки, которые выбираются последовательно в процессе поиска с учетом результатов предыдущих вычислений. Эта стратегия решения называется последовательной.

Последовательная стратегия поиска включает в себя три этапа:

- выбор начального интервала неопределенности; границы a_0, b_0 интервала должны быть такими, чтобы функция f(x) была унимодальной;
- уменьшение интервала неопределенности;
- проверка условия окончания поиска; поиск заканчивается, когда длина текущего интервала неопределенности $[a_k,b_k]$ оказывается меньше установленной величины.

Ответом является множество точек, принадлежащих последнему интервалу неопределенности, среди которых каким-либо образом выбирается решение задачи x^*.

Существует достаточно большое количество методов поиска экстремума для функций одной переменной без ограничений. Это следующие методы: ме-

тод равномерного поиска, метод деления интервала пополам, метод дихотомии, метод золотого сечения, метод Фибоначчи, метод квадратичной интерполяции.

1.2. Метод «золотого сечения»

Для построения конкретного метода одномерной оптимизации, работающего по принципу последовательного сокращения интервала неопределенности, следует задать правило выбора на каждом шаге двух внутренних точек. В методе «золотого сечения» в качестве двух внутренних точек выбираются точки золотого сечения.

Точка производит «золотое сечение» отрезка, если отношение длины всего отрезка к большей части равно отношению большей части к меньшей.

Пусть задана функция f(x) на отрезке [a,b], функция f(x) непрерывна на данном отрезке. Тогда для того, чтобы найти значение этой функции на заданном отрезке, отвечающее критерию поиска минимума, рассматриваемый отрезок делится в пропорции золотого сечения в обоих направлениях (рис. 1), то есть выбираются две точки x_1 и x_2 такие, что:

$$\frac{\mathrm{b}-\mathrm{a}}{\mathrm{b}-\mathrm{x}_1}=\frac{\mathrm{b}-\mathrm{a}}{\mathrm{x}_2-\mathrm{a}}=\phi=\frac{1+\sqrt{5}}{2}=1{,}618... \qquad (2)$$

где ϕ — пропорция золотого сечения.

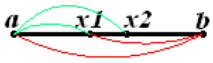


Рис. 1. Выбор промежуточных точек в методе «золотого сечения»

Из формул (2) получаем:

$$x_1=b-\frac{b-a}{\phi}\quad \mathrm{x}_2=\mathrm{a}+\frac{\mathrm{b}-\mathrm{a}}{\phi} \qquad (3)$$

Алгоритм решения задачи следующий.

1. **Шаг 1.** Задают начальные границы отрезка a,b и точность ε, рассчитывают начальные точки деления: $x_1=b-\frac{b-a}{\phi}$, $x_2=a+\frac{b-a}{\phi}$ и значения в них целевой функции: $y_1=f(x_1), y_2=f(x_2)$.

2. **Шаг 2.** Если $y_1\le y_2$, то

$$b=x_2, x_2=x_1, x_1=b-\frac{b-a}{\phi}, y_2=y_1, y_1=f(x_1)$$

Иначе

$$a=x_1, x_1=x_2, x_2=a+\frac{b-a}{\phi}, y_1=y_2, y_2=f(x_2)$$

3. **Шаг 3.**

Если $|b - a| < \varepsilon$, то середина последнего интервала $((x_1 + x_2)/2)$ является искомым значением переменной x, обеспечивающим минимум функции, и поиск решения заканчивается. Иначе необходимо осуществить возврат к Шагу 2.

Структурная схема метода представлена на рис. 2.

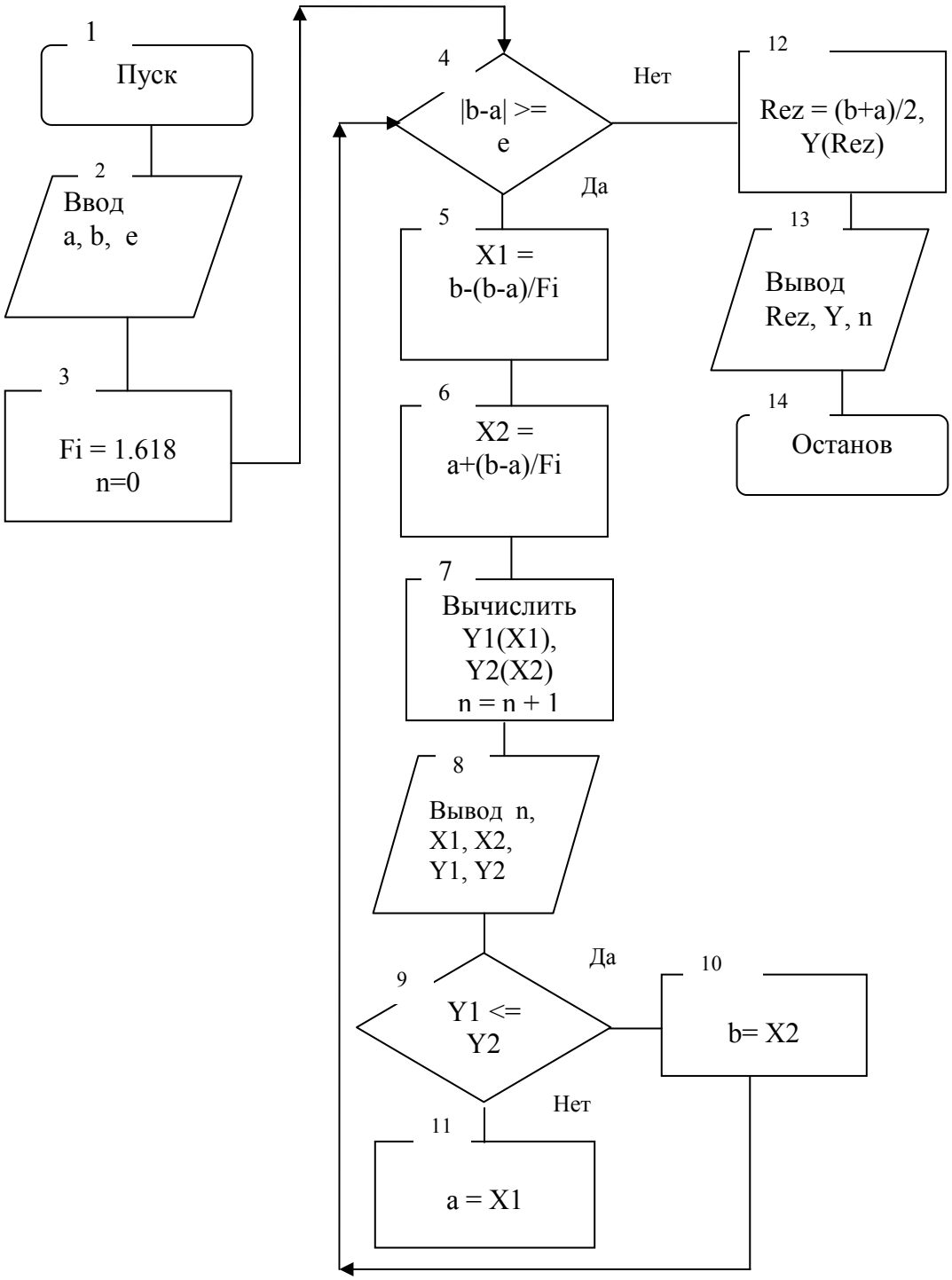


Рис. 2. Структурная схема метода «золотого сечения»

1.3. Методические рекомендации для написания программы в среде программирования Delphi (RAD Studio)

Возможный вид оконного приложения в среде Delphi представлен на рис. 3. Вид оконного приложения может быть другим.

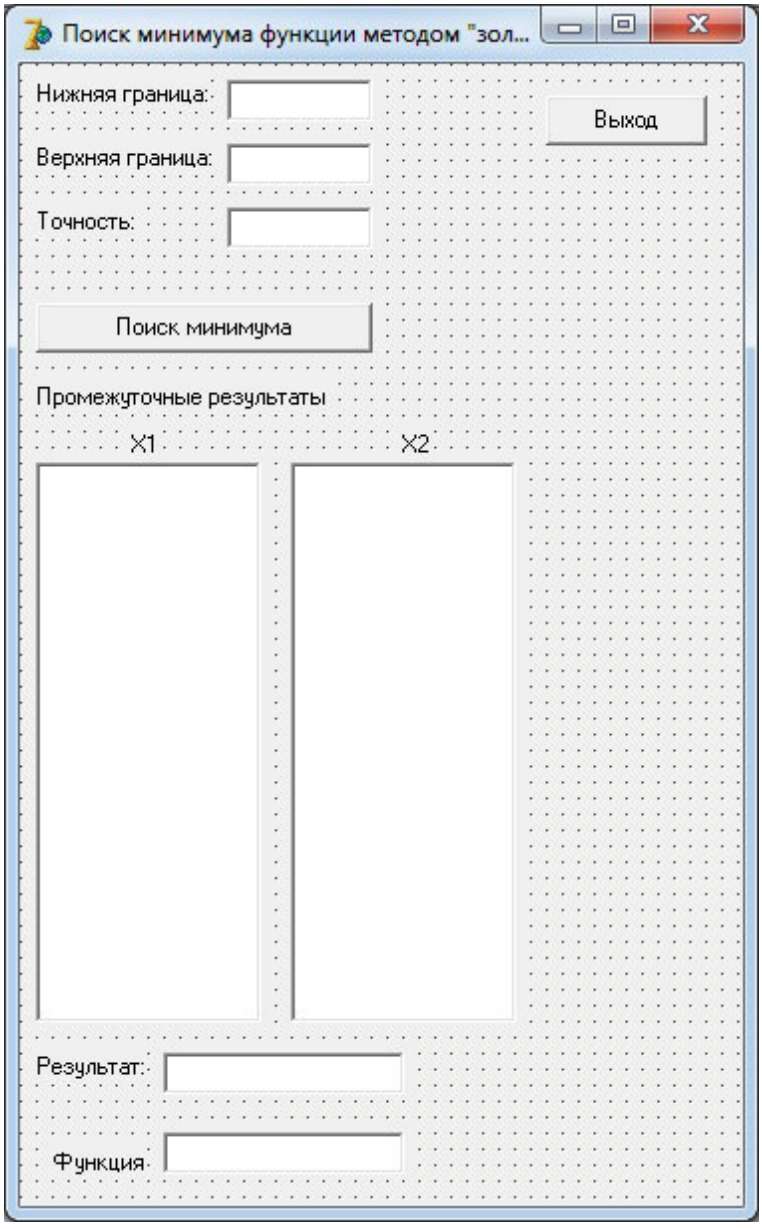


Рис. 3. Возможный вид интерфейса программы

Элементы оконного приложения могут быть созданы с использованием следующих компонентов.

Надпись «Нижняя граница» - компонент Label1, свойство Caption.

Надпись «Верхняя граница» - компонент Label2.

Надпись «Точность» - компонент Label3.

Окно для ввода нижней границы – компонент Edit1.

Окно для ввода верхней границы – компонент Edit2.

Окно для ввода точности – компонент Edit3.

Кнопка «Поиск минимума» - компонент Button1, свойство Caption.

Надпись «Промежуточные результаты» - компонент Label4. Надпись X1 – компонент Label5. Надпись X2 – компонент Label6.

Окна для вывода промежуточных результатов ListBox1, ListBox2.

Надпись «Результат» - компонент Label7.

Окно для вывода результата (x) – компонент Edit4.

Надпись «Функция» - компонент Label8.

Окно для вывода функции (y) – компонент Edit5.

Кнопка «Выход» - компонент Button2.

Данные, **введенные в компоненты Edit**, для использования в вычислениях должны быть **преобразованы из текстового формата в числовой** с использованием следующих функций:

- преобразование строки из Edit1.Text в вещественное число - StrToFloat(Edit1.Text);

- преобразование строки из Edit1.Text в целое число - StrToInt(Edit1.Text).

Преобразование результатов вычислений из числового формата в строковый формат происходит с помощью следующих функций:

- преобразование вещественного числа в текст – функция FloatToStr; например: Edit4.Text := FloatToStr(rez); **форматированный вывод**: FloatToStrF(rez,ffFixed,8,3);

- преобразование целого числа в текст – функция IntToStr.

Добавление в компонент ListBox промежуточных значений и преобразование вещественного числа в строку осуществляется следующим образом:

ListBox1.Items.Add(FloatToStr(X1));

1.4. Методические рекомендации для написания программы в среде Visual Studio на языке C#

Возможный вид оконного приложения в среде Visual Studio для реализации программы на C# представлен на рис. 4.

Элементы оконного приложения могут быть созданы с использованием следующих компонентов.

Надпись «Введите нижнюю граница» - компонент Label1.

Надпись «Введите верхнюю границу» - компонент Label2.

Надпись «Введите точность» - компонент Label3.

Окно для ввода нижней границы – компонент textBox1.

Окно для ввода верхней границы – компонент textBox2.

Окно для ввода точности – компонент textBox3.

Кнопка «Выход» - компонент Button1.

Кнопка «Поиск экстремума» - компонент Button2.

Надпись X1 – компонент Label4. Надпись X2 – компонент Label5.

Окна для вывода промежуточных результатов ListBox1, ListBox2.
Надпись «Результат» - компонент Label6.
Окно для вывода результата (x) – компонент textBox4.
Надпись «Функция» - компонент Label7.
Окно для вывода функции (y) – компонент textBox5.

Для размещения окна формы по центру экрана для свойства формы StartPosition выбирают из списка CenterScreen.

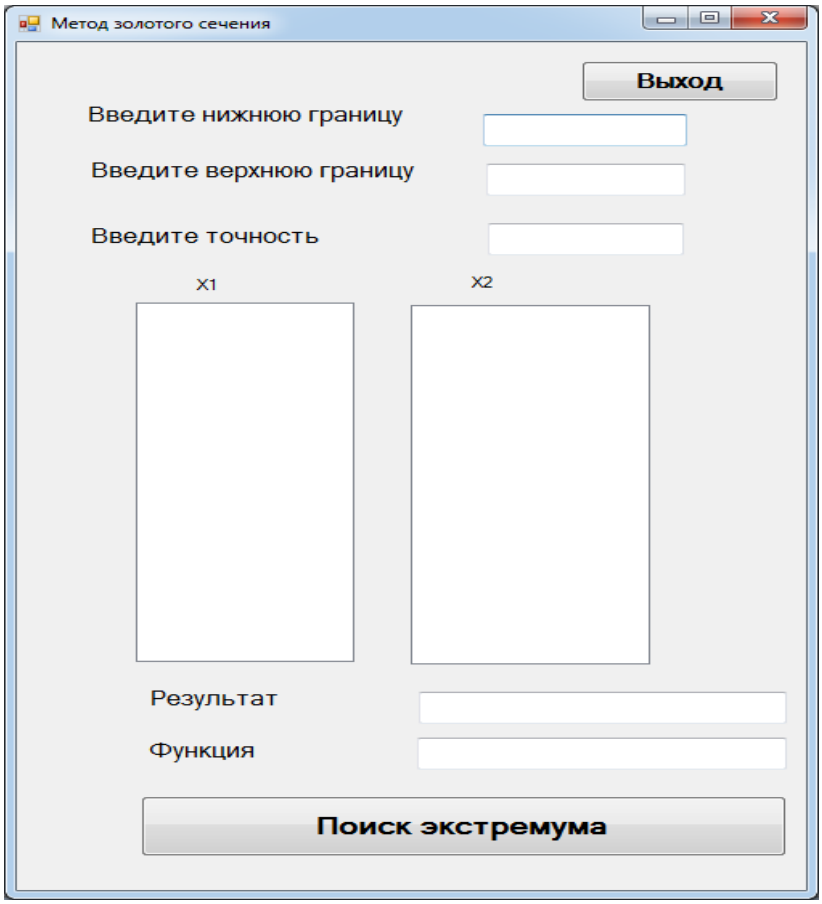


Рис. 4. Оконная форма для ввода данных

Элементы программирования на C#

Вещественные переменные описывают с помощью служебного слова double. Например:

```
double a, b, t, y1, y2;
```

Ввод данных из поля ввода, например из textBox1 в переменную a, реализуется так:

```
a = Convert.ToDouble(textBox1.Text);
```

При этом текстовая переменная преобразуется в вещественное число.

Вычисления с применением математических функций записывают следующим образом:

```
y1 = -Math.Exp(-x1) * Math.Log(x1);
```

Оператор цикла с предусловием записывают так:

```
while (Math.Abs(b - a) >= t)
{
Операторы, повторяющиеся в цикле
}
```

Вывод переменной (например x1) **в компонент listBox1** записывают так:

```
listBox1.Items.Add(Convert.ToString(x1));
```

При этом переменная x1 преобразуется в строковый формат представления.

Проверку условия записывают так:

```
if (y1 <= y2) {операторы, если условие выполняется}
else
{
операторы, если условие не выполняется
}
```

Вывод результата (функции y) **в компонент textBox5** записывают так:

```
textBox5.Text = Convert.ToString(y);
```

При этом реализуется преобразование переменной y в строковый формат.

2. ЛАБОРАТОРНАЯ РАБОТА № 1. РЕАЛИЗАЦИЯ МЕТОДА «ЗОЛОТОГО СЕЧЕНИЯ»

Цель лабораторной работы – разработка, отладка и проверка работы программы, реализующей метод «золотого сечения».

Задание

Выполнить следующие действия:

- создать экранную форму для реализации задания, вариант задания выбрать из табл. 1; номер варианта совпадает с порядковым номером студента в списке группы;
- написать и отладить программу в соответствии с вариантом задания;
- продемонстрировать преподавателю работу программы;
- оформить отчет.

Отчет

Отчет должен содержать:

- титульный лист;
- задание;
- текст программы с комментариями, поясняющими группы операторов;
- скриншот оконной формы с результатами работы программы.

Таблица 1

Варианты заданий для лабораторной работы № 1

Номер варианта	Функция	a	b	E	Ответы X, F
1	$F(x) = -e^{-x} \cdot \ln(x)$	0.1	3	0.1 0.01 0.001	1.763; -0.097
2	$F(x) = -e^{-x} \cdot x$	0.1	3	0.1 0.01 0.001	1.0; -0.368
3	$F(x) = 2 \cdot x^2 + 3 \cdot e^{-x}$	0	1	0.1 0.01 0.001	0.469; 2.317
4	$F(x) = x^2 + x + 5 \cdot e^{-x}$	0	2	0.1 0.01 0.001	0.719; 3.672
5	$F(x)=2 \cdot x^2 - x + e^{-x}$	-1	1	0.1 0.01 0.001	0.415; 0.589
6	$F(x)= -3 \cdot e^{-x} \cdot \ln(2 \cdot x)$	0.5	2.5	0.1 0.01 0.001	1.173; -0.792
7	$F(x)=x^4 - 10 \cdot x^3 + 20 \cdot x^2$	4	8	0.1 0.01 0.001	5.765; -146.726
8	$F(x)=x^4 - 12 \cdot x^3 + 7 \cdot x^2 - 5 \cdot x$	8	9.5	0.1 0.01 0.001	8.61; -1687.886
9	$F(x)=x^4 - 10 \cdot x^2 + 5 \cdot x$	-5	5	0.1 0.01 0.001	-2.352; -36.477
10	$F(x)=x^4 - 10 \cdot x^3 + 20 \cdot x$	6.5	8	0.1 0.01 0.001	7.4; -905.6
11	$F(x)=x^4 - 6 \cdot x^3 - 5 \cdot x^2 + 10 \cdot x$	4	6	0.1 0.01 0.001	4.906; -200.466
12	$F(x) = x4 - 10 \cdot x3 - 30 \cdot x2 - 15 \cdot x$	8	10	0.1 0.01 0.001	9.179; -3300.241

Окончание табл. 1

13	$F(x) = 2x^2 - 12x$	0	5	0.1 0.01 0.001	2.998; -18.00
14	$F(x) = 32x^2 - 15x + 2$	0	0.5	0.1 0.01 0.001	0.233; 0.242
15	$F(x)=10x*\ln(x)-x^2/2$	0,1	1,1	0.1 0.01 0.001	0,38; -3.75
16	$F(x)= e^x-x^3/3+2x$	-4	4	0.1 0.01 0.001	-1,49; -1.65
17	$F(x)=x^2-2x-2\cos(x)$	0	2	0.1 0.01 0.001	0,51; -2.51
18	$F(x)=x^4-14x^3+60x^2-70x$	-1	2	0.1 0.01 0.001	0.78; -24.37
19	$F(x)=x-\ln(\text{abs}(\ln(x)))$	1	3	0.1 0.01 0.001	1,76; 2,33
20	$F(x)=x/(x^2-2x+5)$	-4	2	0.1 0.01 0.001	-2,25; -0,15

3. МЕТОД ПОЛОВИННОГО ДЕЛЕНИЯ

Требуется найти безусловный минимум функции f(x) одной переменной, используя метод половинного деления. Данный метод возвращает приближенное решение оптимизационной задачи с заданной точностью ε. Суть метода заключается в следующем. Отрезок [a,b] разбивается напополам точкой

$$c_0 = \frac{a+b}{2}$$

Далее из двух отрезков $[a,c_0]$ $[c_0,b]$ выбирается тот, который содержит точку минимума. Выбранный отрезок становится исходным отрезком [a,b] для следующего этапа. На следующем этапе новый отрезок [a,b], полученный на предыдущем этапе, делится пополам и вновь находится та его половина, которая содержит точку минимума.

Данная процедура продолжается до тех пор, пока не будет достигнута требуемая точность ε, т.е. пока не будет выполнено условие:

$$|b - a| \le \varepsilon \qquad (4)$$

Условие (4) означает, что длина отрезка стала меньше заданной точности.

Для поиска отрезка, который содержит точку максимума или минимума, к середине отрезка (точка C) можно добавить и вычесть малое значение δ, зависящее от длины отрезка, но много меньшее этой длины. Можно в качестве δ взять заданную точность ε.

Пусть $c1 = c - \varepsilon$, $c2 = c + \varepsilon$.

Тогда если $f(c1) < f(c2)$, то можно предположить, что минимум находится в интервале [a,c], в противном случае – в интервале [c,b].

Поэтому если $f(c1) < f(c2)$, то вместо точки b на следующем этапе будет рассматриваться точка c, в противном случае – вместо точки a – точка c.

Пример расположения точек на графике приведен на рис. 5.

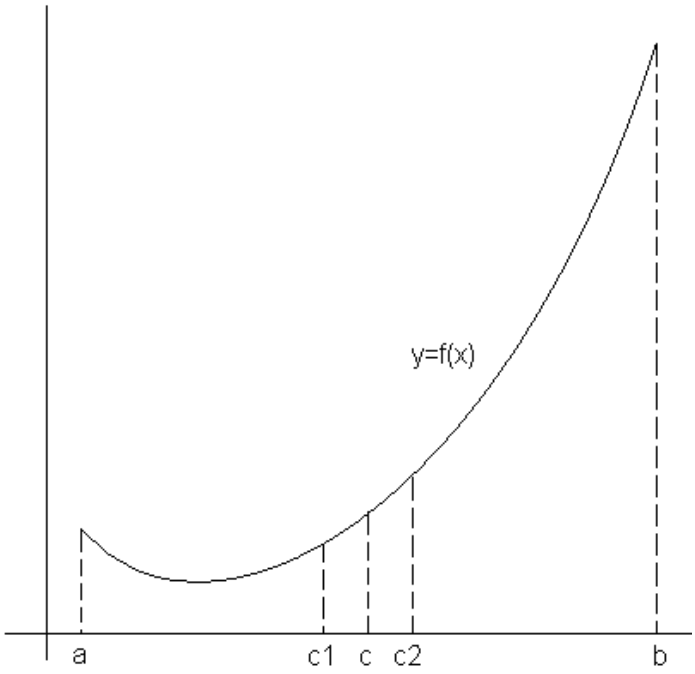


Рис. 5. Выбор интервала в методе половинного деления

4. ЛАБОРАТОРНАЯ РАБОТА № 2. РЕАЛИЗАЦИЯ МЕТОДА ПОЛОВИННОГО ДЕЛЕНИЯ

Цель лабораторной работы – разработка, отладка и проверка работы программы, реализующей метод половинного деления.

Задание

Выполнить следующие действия:

- выбрать вариант задания из табл. 2; номер варианта совпадает с порядковым номером студента в списке группы;
- создать экранную форму для реализации задания, возможный вид экранной формы приведен на рис. 6.

- написать и отладить программу в соответствии с вариантом задания;
- продемонстрировать преподавателю работу программы;
- оформить отчет.

Отчет

Отчет должен содержать:
- титульный лист;
- задание;
- текст программы с комментариями, поясняющими группы операторов;
- скриншот оконной формы с результатами работы программы.

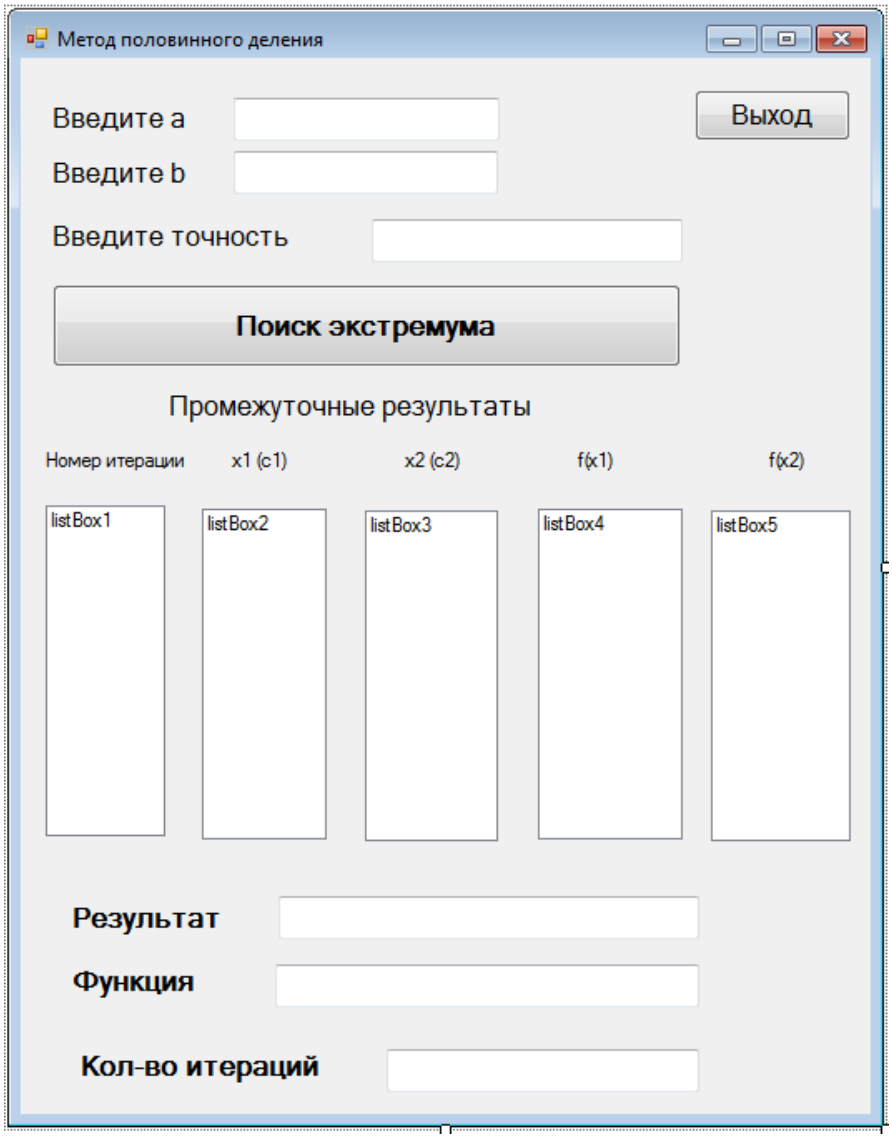


Рис. 6. Экранная форма для ввода данных и вывода результатов

Таблица 2

Варианты заданий для лабораторной работы № 2

Номер варианта	Функция	a	b	E	Ответы X, F
1	$F(x) = -e^{-x} \cdot x$	0.1	3	0.1 0.01 0.001	1.0; -0.368
2	$F(x) = -e^{-x} \cdot \ln(x)$	0.1	3	0.1 0.01 0.001	1.763; -0.097
3	$F(x) = x^2 + x + 5 \cdot e^{-x}$	0	2	0.1 0.01 0.001	0.719; 3.672
4	$F(x) = 2 \cdot x^2 + 3 \cdot e^{-x}$	0	1	0.1 0.01 0.001	0.469; 2.317
5	$F(x) = -3 \cdot e^{-x} \cdot \ln(2 \cdot x)$	0.5	2.5	0.1 0.01 0.001	1.173; -0.792
6	$F(x) = 2 \cdot x^2 - x + e^{-x}$	-1	1	0.1 0.01 0.001	0.415; 0.59
7	$F(x) = x^4 - 12 \cdot x^3 + 7 \cdot x^2 - 5 \cdot x$	8	9.5	0.1 0.01 0.001	8.61; -1687.886
8	$F(x) = x^4 - 10 \cdot x^3 + 20 \cdot x^2$	4	8	0.1 0.01 0.001	5.765; -146.726
9	$F(x) = x^4 - 10 \cdot x^3 + 20 \cdot x$	6.5	8	0.1 0.01 0.001	7.4; -905.6
10	$F(x) = x^4 - 10 \cdot x^2 + 5 \cdot x$	-5	5	0.1 0.01 0.001	-2.352; -36.477
11	$F(x) = x4 - 10 \cdot x3 - 30 \cdot x2 - 15 \cdot x$	8	10	0.1 0.01 0.001	9.179; -3300.241
12	$F(x) = x^4 - 6 \cdot x^3 - 5 \cdot x^2 + 10 \cdot x$	4	6	0.1 0.01 0.001	4.906; -200.466

Окончание табл. 2

13	$F(x) = 32x^2 – 15x + 2$	0	0.5	0.1 0.01 0.001	0.233; 0.242
14	$F(x) = 2x^2 – 12x$	0	5	0.1 0.01 0.001	2.998; -18.00
15	$F(x)= e^x-x^3/3+2x$	-4	4	0.1 0.01 0.001	-1,49; -1.65
16	$F(x)=10x*\ln(x)-x^2/2$	0,1	1,1	0.1 0.01 0.001	0,38; -3.75
17	$F(x)=x^4-14x^3+60x^2-70x$	-1	2	0.1 0.01 0.001	0.78; -24.37
18	$F(x)=x^2-2x-2\cos(x)$	0	2	0.1 0.01 0.001	0,51; -2.51
19	$F(x)=x/(x^2-2x+5)$	-4	2	0.1 0.01 0.001	-2,25; -0,15
20	$F(x)=x-\ln(\text{abs}(\ln(x)))$	1	3	0.1 0.01 0.001	1,76; 2,33

5. РЕАЛИЗАЦИЯ МЕТОДА ПОИСКА МИНИМУМА ФУНКЦИИ НЕСКОЛЬКИХ ПЕРЕМЕННЫХ С ИСПОЛЬЗОВАНИЕМ МЕТОДА ГРАДИЕНТНОГО СПУСКА С ПОСТОЯННЫМ ШАГОМ

5.1. Теоретические сведения

Если в какой-либо точке функция непрерывна и дифференцируема, то в этой точке существует *градиент* целевой функции. Градиент указывает направление наискорейшего возрастания (подъема) функции, а антиградиент - противоположное направление. Градиент определяется как вектор-столбец частных производных L(x) по x.

$$L'(x') = \begin{vmatrix} \frac{\partial L(x')}{\partial x_1} \\ \frac{\partial L(x')}{\partial x_2} \\ \dots \\ \frac{\partial L(x')}{\partial x_n} \end{vmatrix}. \qquad (5)$$

В специальной литературе доказывается, что градиент ортогонален линии уровня, проходящей через точку дифференцирования, и направлен в сторону максимума функции (направление наискорейшего подъема). Линия уровня – это множество точек, для которой целевая функция имеет постоянное значение Противоположное направление градиента (антиградиент) указывает направление минимума (направление наискорейшего спуска).

Постановка задачи. Пусть дана функция f(x), ограниченная снизу на множестве R^n и имеющая непрерывные частные производные во всех его точках.

Требуется найти локальный минимум функции f(x) на множестве допустимых решений $X = R^n$, т.е. найти такую точку $x^* \in R^n$, что

$$f(x^*) = \min_{x \in R^n} f(x).$$

Стратегия поиска. Стратегия решения задачи состоит в построении последовательности точек $\{x^k\}$, k = 0,1,… , таких, что $f(x^{k+1}) < f(x^k)$, k = 0, 1, … . Точки последовательности $\{x^k\}$ вычисляются по правилу

$$x^{k+1} = x^k - t_k \nabla f(x^k), \; k = 0,1,\dots, \qquad (6)$$

где точка x^0 задается пользователем; $\nabla f(x^k)$ – градиент функции f(x), вычисленный в точке x^k; величина шага t_k задается пользователем и остается постоянной до тех пор, пока функция убывает в точках последовательности, что контролируется путем проверки выполнения условия $f(x^{k+1}) - f(x^k) < 0$ или $f(x^{k+1}) - f(x^k) < -\varepsilon \|\nabla f(x^k)\|^2$, $0 < \varepsilon < 1$.

Построение последовательности $\{x^k\}$ заканчивается:

- в точке x^k, для которой $\|\nabla f(x^k)\| < \varepsilon_1$, где ε_1 - заданное малое положительное число (модуль вектора x вычисляется по формуле $\|x\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$),

- или при $k \geq M$, где M – предельное число итераций,

- или при двукратном одновременном выполнении двух неравенств $\|x^{k+1} - x^k\| < \varepsilon_2$, $|f(x^{k+1}) - f(x^k)| < \varepsilon_2$, где ε_2 – малое положительное число.

Вопрос о том, может ли точка x^k рассматриваться как найденное приближение искомой точки минимума, требует проведения дополнительных исследований.

Алгоритм. Алгоритм решения задачи состоит из последовательности шагов.

Шаг 1. Задать x^0, $0 < \varepsilon < 1$, $\varepsilon_1 > 0$, $\varepsilon_2 > 0$. Например: $\varepsilon=0.2$, $\varepsilon_1=0.1$, $\varepsilon_2=0.15$. Задать M – предельное число итераций (например, M=30 или взять значение больше). Найти градиент функции в произвольной точке

$\nabla f(x^k) = (\frac{df(x)}{dx_1},...,\frac{df(x)}{dx_n})^T$.

Шаг 2. Положить k=0.

Шаг 3. Вычислить градиент функции в текущем векторе $\nabla f(x^k)$.

Шаг 4. Проверить выполнение критерия окончания (модуль градиента функции меньше заданной точности) $\|\nabla f(x^k)\| < \varepsilon_1$:

- если критерий выполнен, то поиск закончен, $x^* = x^k$;
- если критерий не выполнен, то перейти к шагу 5.

Шаг 5. Проверить выполнение неравенства $k \geq M$:

- если неравенство выполнено, то расчет окончен и $x^* = x^k$;
- если нет, то перейти к шагу 6.

Шаг 6. Задать величину шага t_k. Например: $t_k=0.5$.

Шаг 7. Вычислить $x^{k+1} = x^k - t_k \nabla f(x^k)$.

Шаг 8. Проверить выполнение условия

$f(x^{k+1}) - f(x^k) < 0$ (или $f(x^{k+1}) - f(x^k) < -\varepsilon \|\nabla f(x^k)\|^2$):

- если условие выполнено, то перейти к шагу 9:
- если условие не выполнено, положить $t_k = t_k / 2$ и перейти к шагу 7.

Шаг 9. Проверить выполнение условий

$\|x^{k+1} - x^k\| < \varepsilon_2$, $|f(x^{k+1}) - f(x^k)| < \varepsilon_2$:

- если оба условия выполнены при текущем значении k и при предыдущем значении k (k=k-1), то расчет окончен и $x^* = x^{k+1}$;
- если хотя бы одно из условий не выполнено, то положить k = k + 1 и перейти к шагу 3.

Геометрическая интерпретация метода для n=2 приведена на рис. 7.

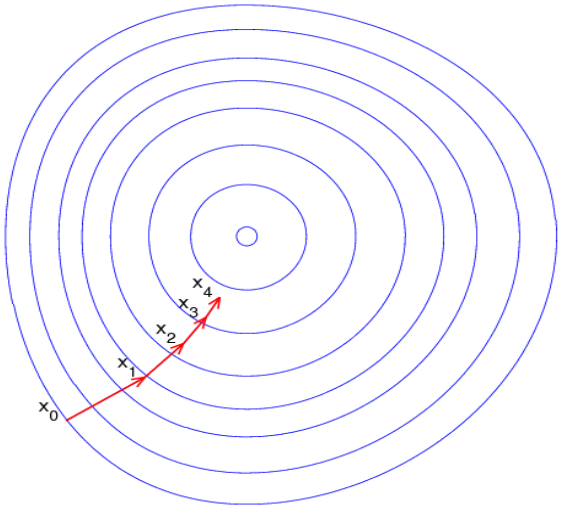


Рис. 7. Геометрическая интерпретация метода

5.2. Структурная схема алгоритма

Структурная схема данного метода представлена на рис. 8.

Основные обозначения на схеме следующие:

- x – исходный вещественный массив;
- x1 – промежуточный вещественный массив x;
- xr – массив - решение (вещественный массив);
- rx – поэлементная разность массивов x1 и x (вещественный массив);
- g - массив-градиент (вещественный массив);
- k – номер (количество) итерации (переменная целого типа);
- m – максимальное количество итераций (переменная целого типа);
- i - параметр цикла (переменная целого типа);
- t – шаг (вещественная переменная);
- e1, e2 – точности (вещественная переменная);
- mm, mm1 – модуль (нормаль) вектора-градиента и разности массивов x1 и x (вещественные переменные);
- u1, u2 – значения функции для массивов x1 и x (вещественные переменные);
- u – разность u1 и u2 (вещественная переменная);
- flag, flag1 - логические переменные, используемые для организации итерационного цикла.

Если функция цели имеет вид

$F(x1, x2, x3) = 2*(x_1+2)^2 + 2*(x_2+3)^2 + 3*(x_3-5)^2$,

то градиент функции цели (вектор частных производных) для неё вычисляют следующим образом:

- для x1 частная производная равна $2*2*(x_1+2) = 4x_1+8$;
- для x2 частная производная равна $2*2*(x_2+3) = 4x_2+12$;

- для x3 частная производная равна $2*3*(x_3-5) = 6x_3-30$.

Таким образом, имеем следующий вектор градиента:

```
g[1] = 4*x[1]+8;
g[2] = 4*x[2]+12;
g[3] = 6*x[3]-30;
```

Модуль градиента функции цели вычисляют по формуле:

```
mm = sqrt(g[1]*g[1]+g[2]*g[2]+g[3]*g[3])
```

В блоках 2-5 осуществляют ввод и подготовку исходных данных.

Блок 6 – это начало цикла с предусловием. Если условие истинно, то выполняют итерацию. Если условие ложно, то происходит выход из цикла и вывод найденного решения XR, функции и количества итераций.

Первоначально в цикле осуществляют вывод номера итерации, массива X и функции (блок 7).

Затем вычисляют градиент и модуль градиента (блоки 8-9).

В блоке 10 проверяют первое условие выхода из цикла по переменной Flag. Если модуль градиента меньше заданной точности, то поиск решения прекращают, переменной Flag = true, последний вектор X запоминают как решение.

Если в блоке 10 условие не выполнено, то в блоке 12 осуществляют проверку второго условия выхода из цикла по переменной Flag. Проверяют, что количество итераций стало больше заданной переменной M. Если это так, то поиск решения прекращают, переменной Flag = true, последний вектор X запоминают как вектор решения XR.

Если оба условия выхода из цикла не выполнились, то продолжают процесс поиска решения.

При этом выполняют следующие действия:

- в блоках 16-26 вычисляют новый вектор X1, функцию в новом векторе X1, проверяют, что функция уменьшилась; если функция не уменьшается, то шаг уменьшают в 10 раз и расчет нового X1 повторяют;
- в блоках 27-32 проводят ряд вычислений и проверяют третье условие выхода из цикла; если оно выполнено, то в 33 блоке переменной Flag= true и в вектор решения записывают вектор X1; если условие не выполнено,, то в блоке 34 количество итераций увеличивают на 1 и в массиве X запоминают X1, осуществляют переход на начало цикла с предусловием по переменной Flag.

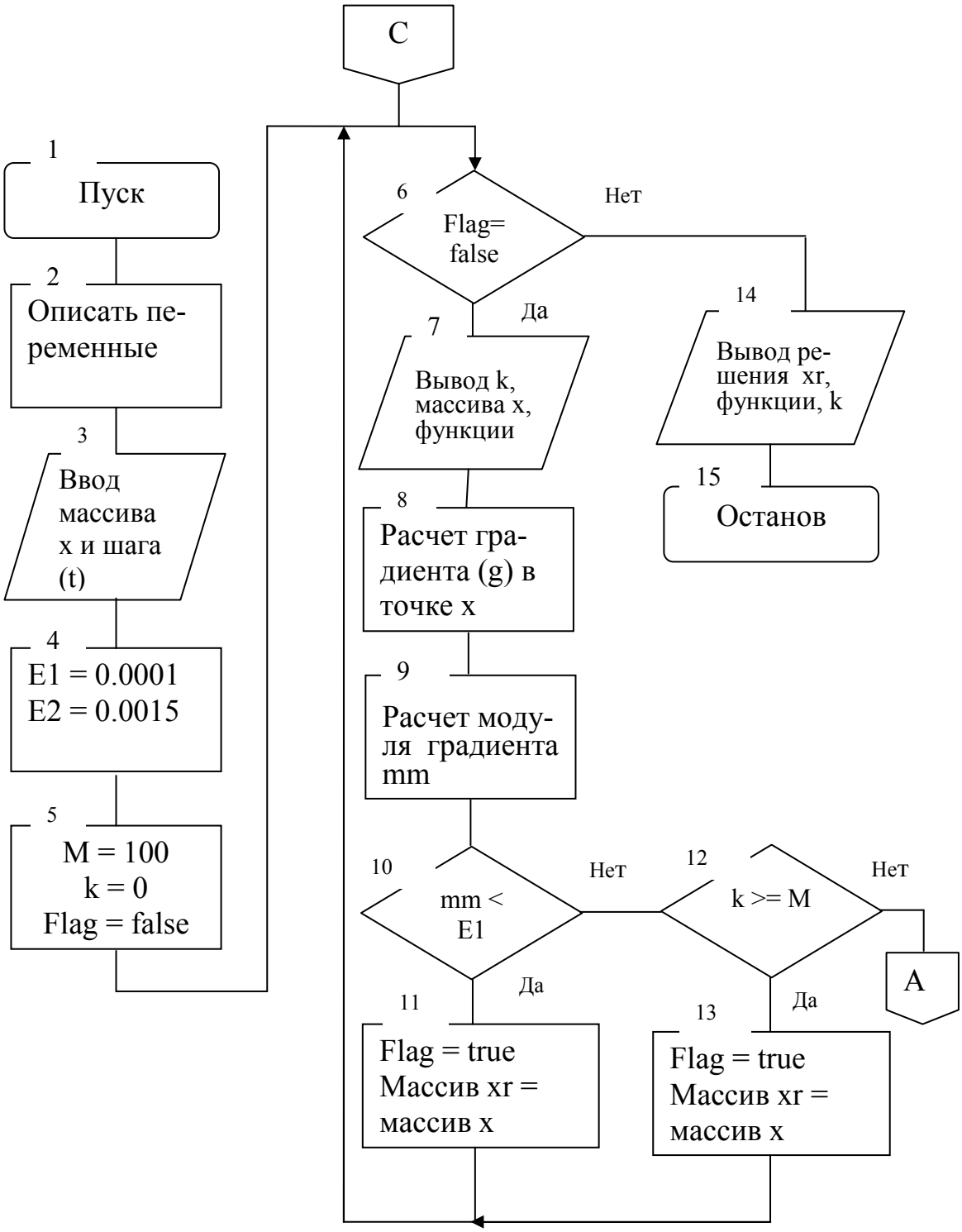


Рис. 8 (начало). Структурная схема градиентного метода

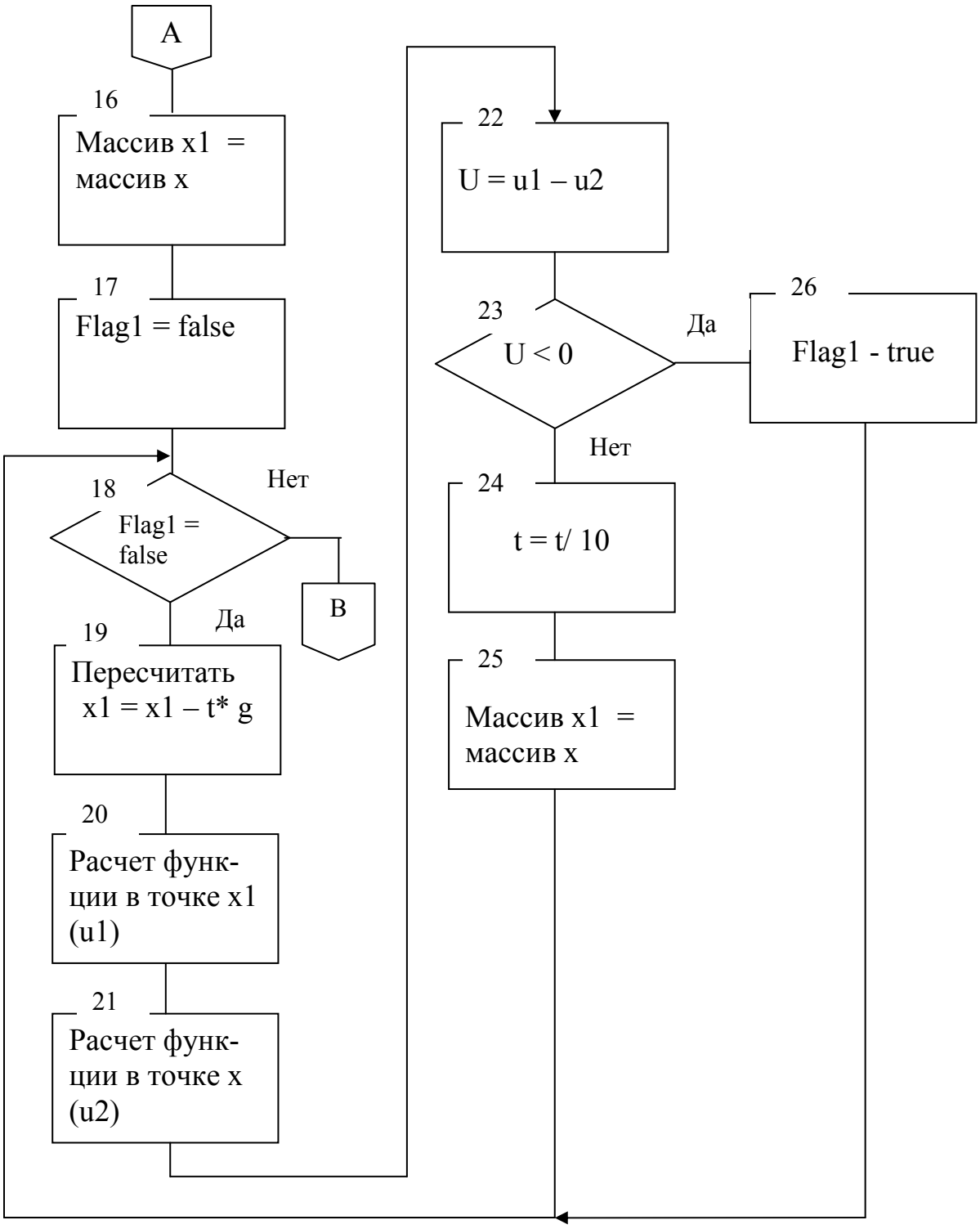


Рис. 8 (продолжение). Структурная схема градиентного метода

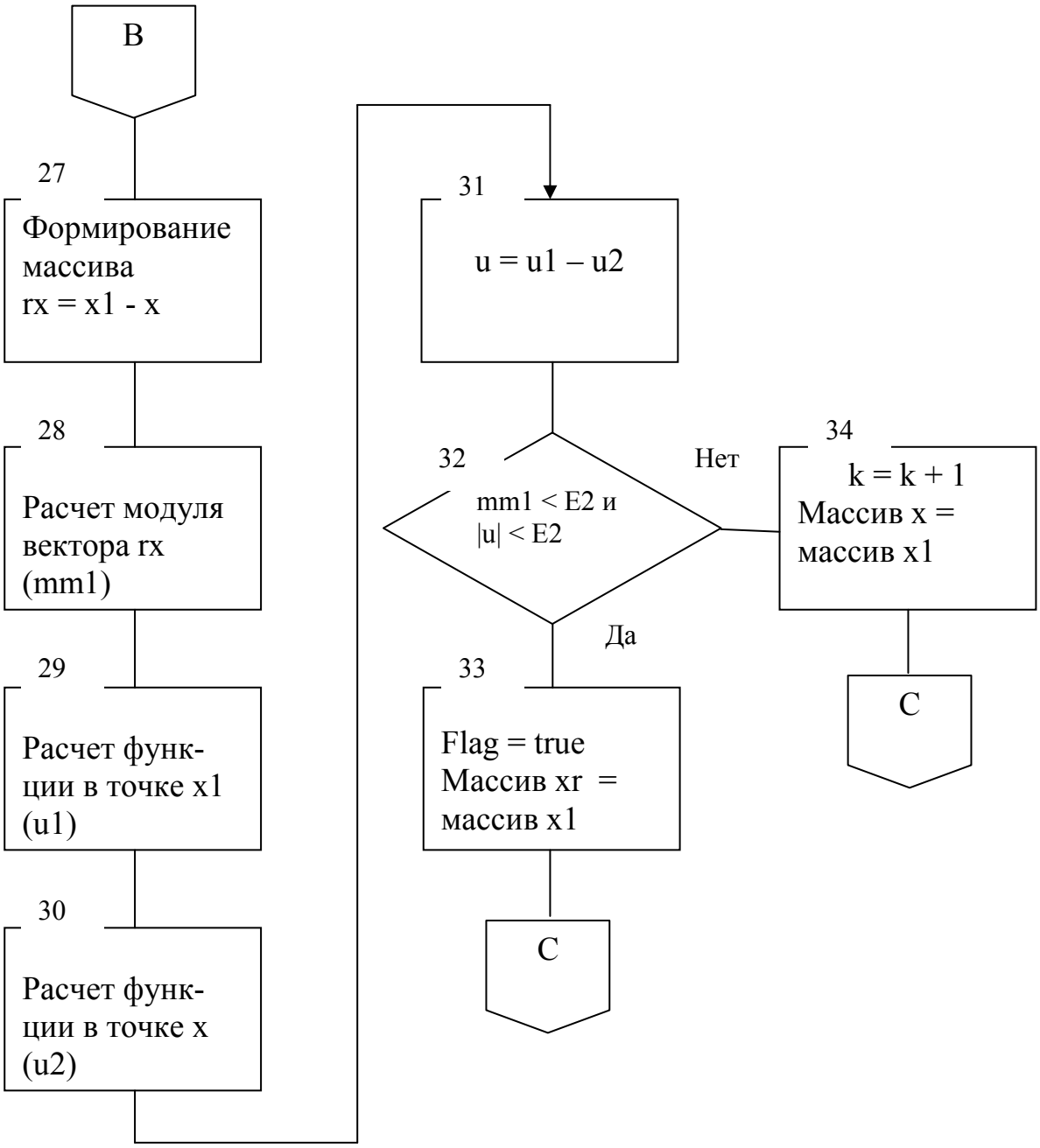


Рис. 8 (окончание). Структурная схема градиентного метода

5.3. Методические рекомендации для написания программы в среде программирования Visual Studio (C#)

Возможный вид пользовательского интерфейса представлен на рис. 9.

Элементы оконного приложения могут быть созданы с использованием следующих компонентов.

- надписи с применением компонента label;
- поля для ввода с применением компонента textbox;

- поля для вывода результата с применением компонента listBox;

- кнопки для вызова процедур обработки событий с применением компонента button.

Для размещения окна формы по центру экрана для свойства формы StartPosition выбирают из списка CenterScreen.

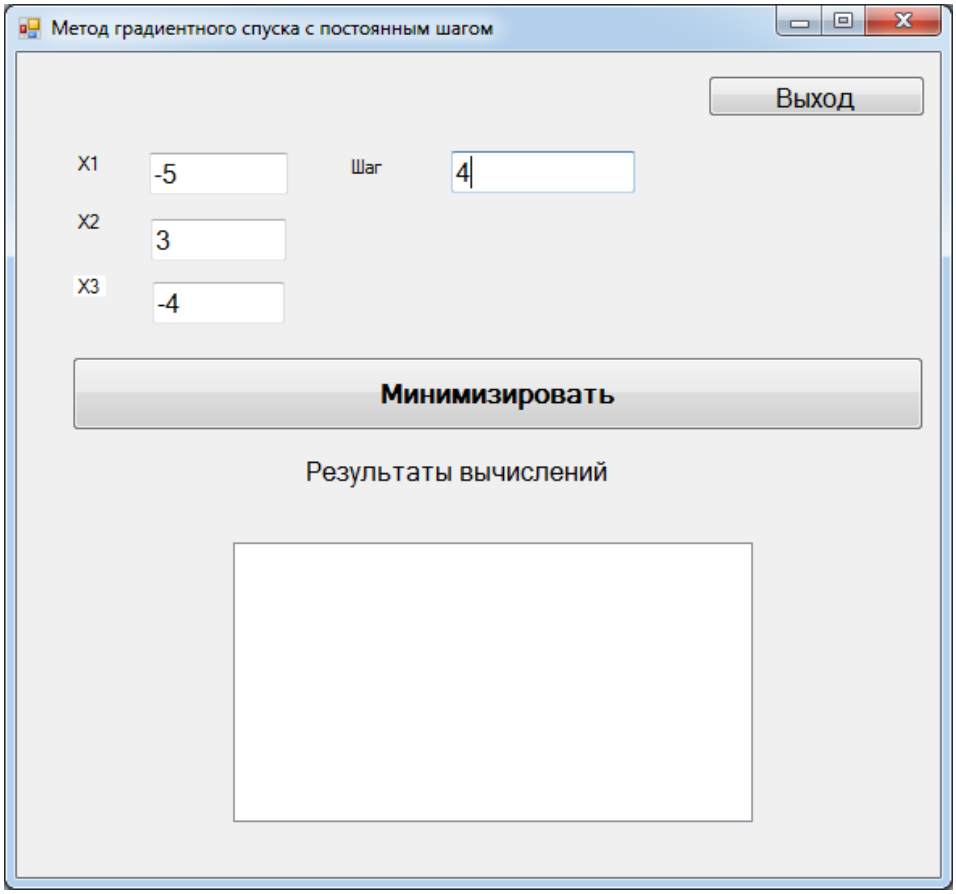


Рис. 9. Вид окна пользовательского интерфейса

Переменные, используемые в программе, описывают до их использования. Например, описание переменных программы приведено ниже:

```
const int n = 3;
double[] g = new double[n];
double[] x = new double[n];
double[] x1 = new double[n];
double[] xr = new double[n], rx = new double[n];
int i, m, k;
double t, e1, e2, mm, mm1, u, u1, u2,ff;
Boolean flag, flag1;
```

Во фрагменте описаны целые переменные, массивы, вещественные и логические переменные.

Операторы присваивания приведены ниже:

```
e1 = 0.001;
e2 = 0.015;
m = 40;
flag = false;
k = 0;
```

Ввод переменных из полей ввода и присвоение их переменным программы:

```
x[0] = Convert.ToDouble(textBox1.Text);
 x[1] = Convert.ToDouble(textBox2.Text);
 x[2] = Convert.ToDouble(textBox3.Text);
t = Convert.ToDouble(textBox4.Text);
```

Вывод в переменную listBox

```
listBox1.Items.Add("k= ");
listBox1.Items.Add(Convert.ToString(k));
istBox1.Items.Add("Точка");
for (int j=0; j< 3; ++j)
   listBox1.Items.Add(Convert.ToString(x[j]));
listBox1.Items.Add(" ");
listBox1.Items.Add("Функция");
ff = 2 * (x[0] + 2) * (x[0] + 2) + 2 * (x[1] + 3) * (x[1] + 3) + 3 * (x[2] - 5) *
(x[2] - 5);
listBox1.Items.Add(Convert.ToString(ff));
listBox1.Items.Add(" ");
```

Оператор проверки одного условия

```
if (mm < e1)
   {
Операторы, если условие выполнено
   }
   else
         {
Операторы, если условие не выполнено
         }
```

Конструкция else может отсутствовать.

Оператор проверки двух условий

```
if  (mm1 < e2 && Math.Abs(u) < e2)
      {
Операторы, если условие выполнено
      }
```

```
        else
        {
Операторы, если условие не выполнено
            }
```

Оператор цикла с заданным числом повторений

```
  for (int j=0; j< 3; ++j)
      {
      Операторы цикла
      }
```

Оператор цикла с предусловием

```
  while (flag == false)
  {
```

Операторы цикла

В цикле обязательно должен быть оператор, меняющий условие, записанное в while. Иначе выход из цикла не состоится, произойдет зацикливание

```
   }
```

6. ЛАБОРАТОРНАЯ РАБОТА № 3. РЕАЛИЗАЦИЯ МЕТОДА ГРАДИЕНТНОГО СПУСКА С ПОСТОЯННЫМ ШАГОМ

Цель лабораторной работы – разработка, отладка и проверка работы программы, реализующей метод градиентного спуска с постоянным шагом.

Задание

Выполнить следующие действия:

- выбрать вариант задания из табл. 3; номер варианта совпадает с порядковым номером студента в списке группы;
- создать экранную форму для реализации задания, возможный вид экранной формы приведен на рис. 10; количество полей ввода зависит от размерности вектора X;
- написать и отладить программу в соответствии с вариантом задания;
- продемонстрировать преподавателю работу программы;
- оформить отчет.

Отчет

Отчет должен содержать:

- титульный лист;
- задание;
- текст программы с комментариями, поясняющими группы операторов;
- скриншот оконной формы с результатами работы программы (вектор X, минимизирующий функцию; минимальное значение функции; количество итераций, при котором было достигнуто минимальное значение функции).

Таблица 3

Варианты заданий для лабораторной работы № 3

№ вар.	Функция	X	H	Xmin
1	$F(x1..x3) = 2*(x1+2)^2 + 2*(x2+3)^2 + 3*(x3-5)^2$	(-5, 3, -4)	4	(-2, -3, 5)
2	$F(x1..x2) = 2*(x1+7)^2 + (x2-11)^2$	(5, -5)	4	(-7, 11)
3	$F(x1..x4) = 2*(x1+1)^2 + 4*(x2-2)^2 + 6*(x3+3)^2 + 8*(x4-4)^2$	(4,-4,7, -7)	4	(-1,2, -3,4)
4	$F(x1..x3) =(x1-5)^2 + 3*(x2+7)^2 + (x3-9)^2$	(-1,3,7)	4	(5,-7,9)
5	$F(x1..x2) =3*(x1+5)^2 + 2*(x2-5)^2$	(4,-4)	4	(-5,5)
6	$F(x1..x4) = (x1+7)^2 + 2*(x2-7)^2 + 3*(x3+7)^2 + 4*(x4-7)^2$	(4,-1,4, -1)	4	(-7,7, -7,7)
7	$F(x1..x3) =(x1-25)^2 + 2*(x2+27)^2 +2*(x3-29)^2$	(-1, 2, -1)	4	(25, -27, 29)
8	$F(x1..x2) =10*(x1+5)^2 + 12*(x2-15)^2$	(10, -10)	4	(-5, 15)
9	$F(x1..x4) = 4*(x1-5)^2 + 3*(x2-10)^2 + 2*(x3+15)^2 + (x4+20)^2$	(2, -2, 2, -2)	4	(5, 10, -15, -20)
10	$F(x1..x3) = (x1+2)^2 + 2*(x2+4)^2 + 4*(x3+6)^2$	(1, 1, 1)	4	(-2, -4, -6)
11	$F(x1..x2) =3*(x1-21)^2 + 2*(x2+12)^2$	(-5, -5)	4	(21, -12)
12	$F(x1..x4) = 5*(x1+11)^2 + 7*(x2+9)^2 + 9*(x3+7)^2 + 11*(x4+5)_2$	(2, 2, 2, 2)	4	(-11, -9, -7, -5)
13	$F(x1..x2) = 2*(x1+5)^2 + (x2- 7)^2$	(2, -2)	4	(-5, 7)
14	$F(x1..x3) =(x1-4)^2 + 3*(x2+5)^2 + (x3-6)^2$	(-1, 3, 5)	4	(4, -5, 6)
15	$F(x1,x2) = 5(x_1 - x_2)^2 + 10x_1x_2 + 6x_2^2$ Точность 0,002 Коэффициент для пересчета шага 0,1	5;2.5	1.2	(0, 0)

Окончание табл. 3

16	$F(x1,x2) = 2x_1^2 + 2x_1x_2 + 3(x_2 - x_1)^2$ Точность 0,005 Коэффициент для пересчета шага 0,01	2; 1	2.5	(0,0)
17	$F(x1,x2) = 10x_1^2 + 4x_1x_2 + x_2^2$ Точность 0,003 Коэффициент для пересчета шага 0,05	4; 5	1.8	(0, 0)
18	$F(x1,x2) = 6x_1^2 + 0.5x_1x_2 + 5x_2^2$ Точность 0,002 Коэффициент для пересчета шага 0,1	6; 5	1.2	(0, 0)
19	$F(x1,x2) = (x_1 - x_2)^2 + 2x_1x_2 + 6x_2^2$ Точность 0,001 Коэффициент для пересчета шага 0,02	2; 2	1.3	(0, 0)
20	$F(x1..x2) = 10*(x1+3)^2 + 12*(x2-10)^2$	(8, -8)	4	(-3, 10)

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Акулич И.Л. Математическое программирование в примерах и задачах: учеб. пособие / И.Л. Акулич. СПб.: Издательство «Лань», 2009. – 352 с.

2. Агальцов В.П. Математические методы в программировании / В.П. Агальцов. М.: ИД «Форум», 2010. – 240 с.

3. Струченков В.И. Методы оптимизации в прикладных задачах [Электронный ресурс] / В. И. Струченков— Электрон. текстовые данные.— М.: СОЛОН-ПРЕСС, 2009.— 315 с.— Режим доступа: http://www.iprbookshop.ru/8722.html.— ЭБС «IPRbooks»

4. Пантелеев А.В. Методы оптимизации [Электронный ресурс]: учебное пособие / Пантелеев А.В., Летова Т.А.— Электрон. текстовые данные.— М.: Логос, 2011.— 424 с.— Режим доступа: http://www.iprbookshop.ru/9093.html.— ЭБС «IPRbooks»

5. Розова В.Н. Методы оптимизации [Электронный ресурс]: учебное пособие / Розова В.Н., Максимова И.С.— Электрон. текстовые данные.— М.: Российский университет дружбы народов, 2010.— 112 с.— Режим доступа: http://www.iprbookshop.ru/11536.html.— ЭБС «IPRbooks»

6. Холопкина Л.В., Кремер О.Б. Методы оптимизации, компьютерные технологии. – Воронеж, ВГТУ, 2016. -126 с.

7. Олейникова С.А. Численные методы оптимизации: практикум / С.А. Олейникова, Т.И. Сергеева, М.Ю. Сергеев. Воронеж: Изд-во ВГТУ, 2021. – 90 с.

ОГЛАВЛЕНИЕ

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

к выполнению лабораторных работ
по дисциплине «Методы оптимизации»
для студентов направления подготовки 09.03.01
«Информатика и вычислительная техника»
(профиль «Вычислительные машины, комплексы, системы и сети»)
заочной формы обучения

Составители:
Сергеева Татьяна Ивановна
Сергеев Михаил Юрьевич
Недикова Татьяна Николаевна

В авторской редакции

Компьютерный набор Т. И. Сергеевой

Подписано к изданию 23.09.2021.
Уч.-изд. л. 1,7.

ФГБОУ ВО «Воронежский государственный технический университет»
394026 Воронеж, Московский просп., 14