

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**
(БГТУ им. В.Г. Шухова)



ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ

Лабораторная работа №0
по дисциплине: Вычислительная математика
тема: «Погрешности. Приближенные вычисления. Вычислительная
устойчивость.»

Выполнил: ст. группы ВТ-231
Масленников Даниил
Александрович

Проверили:
Островский Алексей Мичеславович
Кабалянц Пётр Степанович

Белгород 2025 г.

Лабораторная работа № 0

Цель работы: Изучить особенности организации вычислительных процессов, связанные с погрешностями, приближенным характером вычислений на компьютерах современного типа, вычислительной устойчивостью.

Вариант 14

- 1) Рассмотреть источники погрешности в решении численных задач и способы их оценки.
- 2) Изучить особенности работы с машинными числами как с результатом дискретной проекции вещественных чисел на конкретную архитектуру компьютера.
- 3) Выяснить условия обеспечения вычислительной устойчивости решения основных численных задач.
- 4) Выполнить индивидуальное задание, закрепляющее на практике полученные знания (номер задания соответствует номеру студента по журналу; если этот номер больше, чем максимальное число заданий, тогда вариант задания вычисляется по формуле: номер по журналу % максимальный номер задания, где % - остаток от деления).

<pre>from math import sqrt s = (a + b + c) / 2</pre>	<pre>from math import sqrt max_side = max(a, b, c) res = sqrt(s * (s - a) * (s - b) * (s - c)) a_scaled, b_scaled, c_scaled = a / max_side, b / max_side, c / max_side s_scaled = (a_scaled + b_scaled + c_scaled) / 2 area_scaled = sqrt(s_scaled * (s_scaled - a_scaled) * (s_scaled - b_scaled) * (s_scaled - c_scaled)) res = area_scaled * max_side**2</pre>
--	--

```
#include <stdio.h>
#include <math.h>
#include <float.h>
```

```
float calculate_area(float a, float b, float c) {
    float s = (a + b + c) / 2.0f;
    return sqrtf(s * (s - a) * (s - b) * (s - c));
}
```

```
float calculate_scaled_area(float a, float b, float c) {
    float max_side = fmaxf(a, fmaxf(b, c));
    float a_scaled = a / max_side;
    float b_scaled = b / max_side;
    float c_scaled = c / max_side;
    float s_scaled = (a_scaled + b_scaled + c_scaled) / 2.0f;
    float area_scaled = sqrtf(s_scaled * (s_scaled - a_scaled) * (s_scaled - b_scaled) * (s_scaled -
c_scaled));
}
```

```

    return area_scaled * powf(max_side, 2.0f);
}

int main() {
    float triangles[][3] = {
        {3.0f, 4.0f, 5.0f},
        {300.0f, 400.0f, 500.0f},
        {0.1f, 0.1f, 0.1f},
        {0.00001f, 0.00001f, 0.00001f},
        {900, 900, 900}
    };
    int size = sizeof(triangles) / sizeof(triangles[0]);

    for (int i = 0; i < size; i++) {
        float a = triangles[i][0];
        float b = triangles[i][1];
        float c = triangles[i][2];

        // Вычисляем обычным способом
        float area = calculate_area(a, b, c);

        // Вычисляем с масштабированием
        float scaled_area = calculate_scaled_area(a, b, c);

        // Вычисляем абсолютную и относительную погрешность
        float error = fabsf(area - scaled_area);
        float relative_error = (error * 100.0f) / area;

        // Выводим результаты
        printf("Стороны: a = %f, b = %f, c = %f\n", a, b, c);
        printf(" Обычный способ: %f\n", area);
        printf(" Масштабированный способ: %f\n", scaled_area);
        printf(" Абсолютная погрешность: %2f\n", error);
        printf(" Относительная погрешность: %f%%\n\n", relative_error);
    }

    return 0;
}

```

```

> gcc -o res lab0.c -lm
> ./res
Стороны: a = 3.000000, b = 4.000000, c = 5.000000
Обычный способ: 6.000000
Масштабированный способ: 6.000001
Абсолютная погрешность: 0.000001
Относительная погрешность: 0.000016 %

Стороны: a = 300.000000, b = 400.000000, c = 500.000000
Обычный способ: 60000.000000
Масштабированный способ: 60000.011719
Абсолютная погрешность: 0.011719
Относительная погрешность: 0.000020 %

Стороны: a = 0.100000, b = 0.100000, c = 0.100000
Обычный способ: 0.004330
Масштабированный способ: 0.004330
Абсолютная погрешность: 0.000000
Относительная погрешность: 0.000011 %

Стороны: a = 0.000010, b = 0.000010, c = 0.000010
Обычный способ: 0.000000
Масштабированный способ: 0.000000
Абсолютная погрешность: 0.000000
Относительная погрешность: 0.000008 %

Стороны: a = 900.000000, b = 900.000000, c = 900.000000
Обычный способ: 350740.312500
Масштабированный способ: 350740.281250
Абсолютная погрешность: 0.031250
Относительная погрешность: 0.000009 %

```

Мой вариант задания показывает чувствительность результата к последовательности операций, так как с математической точки зрения, должен был быть одинаковый ответ во всех случаях, но компьютер из-за особенностей представления чисел в компьютере. Вывод: лабораторная работа позволила получить практические навыки работы с погрешностями, машинными числами и вычислительной устойчивостью, что является важным аспектом при разработке численных методов и алгоритмов.