

**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 №2**  
по дисциплине: Исследование операций  
тема: «Симплекс-метод в чистом виде»

Выполнил: ст. группы ПВ-223  
Пахомов Владислав Андреевич

Проверили:  
проф. Вирченко Юрий Петрович

Белгород 2024 г.

## Лабораторная работа №2

### Симплекс-метод в чистом виде

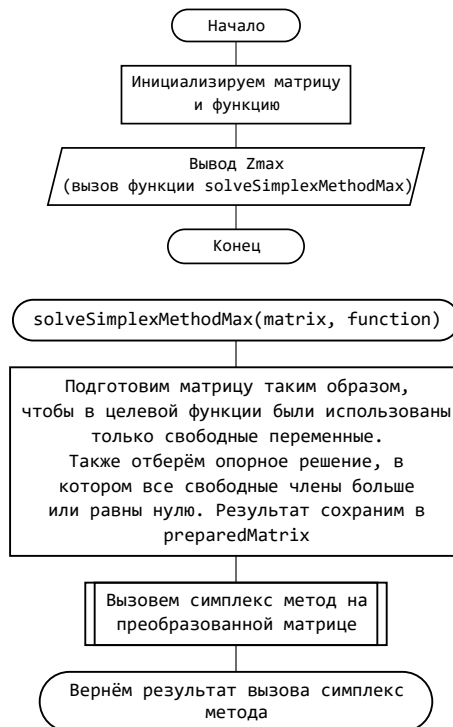
**Цель работы:** изучение симплекс-метода для решения задачи линейного программирования с использованием симплекс-таблиц, получение навыков кодирования изученного алгоритма, отладки и тестирования соответствующих программ.

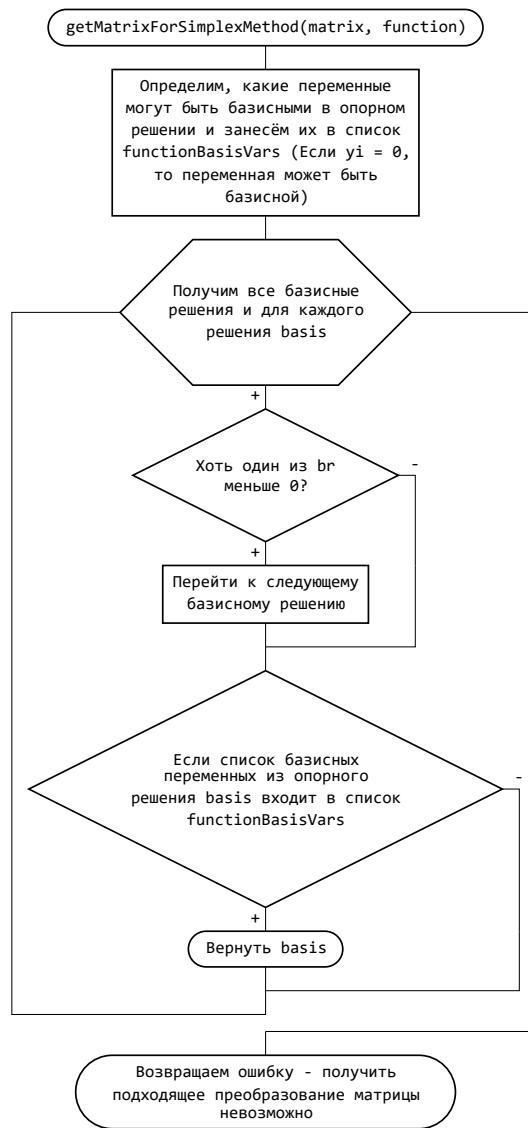
**Задание:** запрограммировать и отладить изученный алгоритм. В рамках подготовки тестовых данных решить задачу вручную.

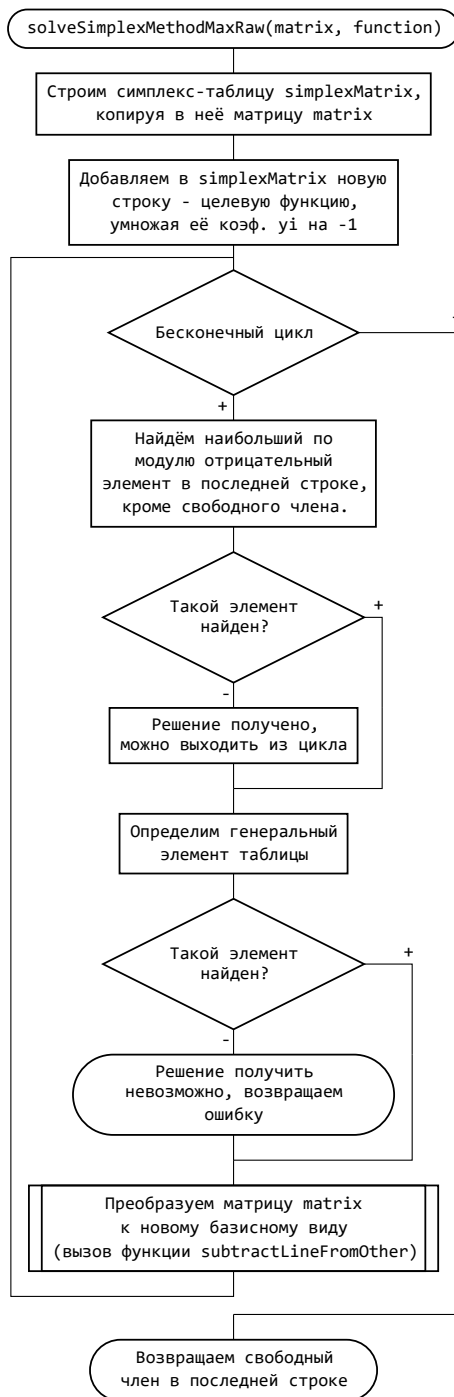
### Вариант 10

$$\begin{cases} z = 6x_1 + 2x_2 - 5x_4 \rightarrow \max; \\ -x_1 + 3x_2 + 6x_4 + x_6 = 26, \\ 5x_1 + x_2 + 7x_3 - 3x_4 = 14, \\ 2x_1 - x_2 - 2x_4 + x_5 = 12, \\ x_i \geq 0 (i = \overline{1, 6}). \end{cases}$$

### Блок-схемы:







## Листинг программы:

```

#include <iostream>
#include <iomanip>
#include <array>

#include "../Libs/alg/alg.h"

int main() {
    // Инициализируем матрицу и функцию
    std::vector<std::array<double, 7>> matrix = {
        {-1, 3, 0, 6, 0, 1, 26},
        {5, 1, 7, -3, 0, 0, 14},
        {2, -1, 0, -2, 1, 0, 12}};
    std::array<double, 7> function{{6.0, 2.0, 0.0, -5.0, 0.0, 0.0, 0.0}};

    // Вывод ответа

```

```
std::cout << "Zmax: " << solveSimplexMethodMax(matrix, function);  
}
```

## Ссылка на репозиторий

```
#pragma once  
  
#include <optional>  
#include <vector>  
#include <algorithm>  
  
#include "../lab1/task1.tpp"  
  
template <std::size_t T>  
auto getMatrixForSimplexMethod(std::vector<std::array<double, T>>& matrix, std::array<double, T>& function) {  
    // Определим, какие переменные могут быть базисными в опорном решении и занесём их в список functionBasisVars  
    // (Если  $y_i = 0$ , то переменная может быть базисной)  
    std::vector<int> functionBasisVars;  
    for (int i = 0; i < function.size() - 1; i++)  
        if (std::abs(function[i]) < EPS)  
            functionBasisVars.push_back(i);  
  
    // Получим все базисные решения и для каждого решения basis  
    for (auto& basis : getAllBasises(matrix)) {  
        bool isAllBsMoreOrEqualToZero = true;  
        for (int i = 0; i < basis.matrix.size() && isAllBsMoreOrEqualToZero; i++) {  
            if (basis.matrix[i].back() < EPS)  
                isAllBsMoreOrEqualToZero = false;  
        }  
  
        // Хотя один из  $b_i$  меньше 0?  
        if (!isAllBsMoreOrEqualToZero) {  
            // Перейти к следующему базисному решению  
            continue;  
        }  
  
        // Если список базисных переменных из опорного решения basis входит в список  
        // functionBasisVars  
        std::sort(basis.indices.begin(), basis.indices.end());  
        if (std::includes(basis.indices.begin(), basis.indices.end(), functionBasisVars.begin(),  
            ↪ functionBasisVars.end()))  
            // Возвращаем искомый базис  
            return basis;  
    }  
  
    // Возвращаем ошибку - получить подходящее преобразование матрицы невозможно  
    throw std::invalid_argument("No basis for function found");  
}  
  
template <std::size_t T>  
double solveSimplexMethodMaxRaw(std::vector<std::array<double, T>>& matrix, std::array<double, T>& function) {  
    auto preparedMatrix = getMatrixForSimplexMethod(matrix, function);
```

```

// Строим симплекс-таблицу, копируя в неё матрицу matrix
std::vector<std::array<double, T>> simplexMatrix(preparedMatrix.matrix);
// Добавляем новую строку - целевую функцию, умножая её коэф. yi на -1
simplexMatrix.push_back(function);
for (int i = 0; i < T; i++)
    simplexMatrix.back()[i] *= -1;

// Бесконечный цикл
while (true) {
    // Найдём наибольший по модулю отрицательный элемент в последней строке, кроме свободного члена.
    int minColumnIndex = -1;
    for (int i = 0; i < T - 1; i++) {
        if (simplexMatrix.back()[i] < 0 && (minColumnIndex == -1 || simplexMatrix.back()[minColumnIndex] >
            ↪ simplexMatrix.back()[i]))
            minColumnIndex = i;
    }

    // Такой элемент найден?
    if (minColumnIndex == -1) {
        // Решение получено, можно выходить из цикла
        break;
    }

    // Определим генеральный элемент таблицы
    int minRowIndex = -1;
    for (int i = 0; i < simplexMatrix.size() - 1; i++) {
        if (simplexMatrix[i][minColumnIndex] <= EPS) continue;
        if (minRowIndex == -1) minRowIndex = i;
        else if (simplexMatrix[minRowIndex].back() / simplexMatrix[minRowIndex][minColumnIndex] >
            simplexMatrix[i].back() / simplexMatrix[i][minColumnIndex]) minRowIndex = i;
    }

    // Такой элемент найден?
    if (minRowIndex == -1) {
        // Решение получить невозможно, возвращаем ошибку
        throw std::invalid_argument("No solution");
    }

    // Преобразуем матрицу к новому базисному виду
    subtractLineFromOther(simplexMatrix, minRowIndex, minColumnIndex);
}

// Возвращаем свободный член в последней строке
return simplexMatrix.back().back();
}

template <std::size_t T>
double solveSimplexMethodMax(std::vector<std::array<double, T>>& matrix, std::array<double, T>& function) {
    // Подготовим матрицу таким образом, чтобы в целевой функции были использованы только свободные переменные.
    // Также отберём опорное решение, в котором все свободные члены больше или равны нулю
    auto preparedMatrix = getMatrixForSimplexMethod(matrix, function);

    // Вызовем симплекс метод на преобразованной матрице
    return solveSimplexMethodMaxRaw(preparedMatrix.matrix, function);
}

```

}

## Ссылка на репозиторий

Результат выполнения программы:

Zmax: 24

Дополнительно проведём тестирование при помощи программы отбора базисных решений (см. лабораторная работа №1 задание 2).

Результат выполнения программы:

Обнаружено опорное решение: {0; 0; 2; 0; 12; 26}

Значение функции  $z(B)$ : 0

Обнаружено опорное решение: {0; 0; 3.85714; 4.33333; 20.6667; 0}

Значение функции  $z(B)$ : -21.6667

Обнаружено опорное решение: {0; 8.66667; 0.761905; 0; 20.6667; 0}

Значение функции  $z(B)$ : 16.8

Обнаружено опорное решение: {6; 0; 0; 5.33333; 10.6667; 0}

Значение функции  $z(B)$ : 9.33333

Обнаружено опорное решение: {1; 9; 0; 0; 19; 0}

Значение функции  $z(B)$ : 24

Zmax: 24

Оптимальный план: {1; 9; 0; 0; 19; 0}

Результаты выполнения программ совпали.

## Результаты вычислений:

Имеем целевую функцию  $z = 6x_1 + 2x_2 - 5x_4 \rightarrow \max$ . и систему уравнений

$$\begin{cases} -x_1 + 3x_2 + 6x_4 + x_6 = 26, \\ 5x_1 + x_2 + 7x_3 - 3x_4 = 14, \\ 2x_1 - x_2 - 2x_4 + x_5 = 12, \end{cases}$$

Приведём систему уравнений к базисному виду. Базисными переменными выберем  $x_3$ ,  $x_5$  и  $x_6$ . Построим расширенную матрицу:

$$\begin{pmatrix} -1 & 3 & 0 & 6 & 0 & 1 & 26 \\ 5 & 1 & 7 & -3 & 0 & 0 & 14 \\ 2 & -1 & 0 & -2 & 1 & 0 & 12 \end{pmatrix}$$

Поменяем местами 1 и 2 строки, 2 и 3.

$$\begin{pmatrix} 5 & 1 & 7 & -3 & 0 & 0 & 14 \\ 2 & -1 & 0 & -2 & 1 & 0 & 12 \\ -1 & 3 & 0 & 6 & 0 & 1 & 26 \end{pmatrix}$$

Выберем ведущий элемент  $x_3$  в первой строке:

$$\begin{pmatrix} 5 & 1 & \|7\| & -3 & 0 & 0 & 14 \\ 2 & -1 & 0 & -2 & 1 & 0 & 12 \\ -1 & 3 & 0 & 6 & 0 & 1 & 26 \end{pmatrix}$$

Разделим ведущую строку на 7.

$$\begin{pmatrix} \frac{5}{7} & \frac{1}{7} & 1 & -\frac{3}{7} & 0 & 0 & 2 \\ 2 & -1 & 0 & -2 & 1 & 0 & 12 \\ -1 & 3 & 0 & 6 & 0 & 1 & 26 \end{pmatrix}$$

Привели систему к базисному виду

$$\begin{cases} x_3 = 2 - (\frac{5}{7}x_1 + \frac{1}{7}x_2 - \frac{3}{7}x_4) \\ x_5 = 12 - (2x_1 - x_2 - 2x_4) \\ x_6 = 26 - (-x_1 + 3x_2 + 6x_4) \end{cases}$$

Построим симплекс-таблицу

Баз. пер.	Св. чл.	$x_1 \downarrow$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
$\leftarrow x_3$	2	$\frac{5}{7}$	$\frac{1}{7}$	1	$-\frac{3}{7}$	0	0
$x_5$	12	2	-1	0	-2	1	0
$x_6$	26	-1	3	0	6	0	1
$z$	0	-6	-2	0	5	0	0

Баз. пер.	Св. чл.	$x_1$	$x_2 \downarrow$	$x_3$	$x_4$	$x_5$	$x_6$
$x_1$	$2\frac{4}{5}$	1	$\frac{1}{5}$	$1\frac{2}{5}$	$-\frac{3}{5}$	0	0
$x_5$	$6\frac{2}{5}$	0	$-1\frac{2}{5}$	$-2\frac{4}{5}$	$-\frac{4}{5}$	1	0
$\leftarrow x_6$	$28\frac{4}{5}$	0	$3\frac{1}{5}$	$1\frac{2}{5}$	$5\frac{2}{5}$	0	1
$z$	$16\frac{4}{4}$	0	$-\frac{4}{5}$	$8\frac{2}{5}$	$1\frac{2}{5}$	0	0

Баз. пер.	Св. чл.	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
$x_1$	1	1	0	$1\frac{5}{16}$	$-\frac{15}{16}$	0	$\frac{1}{16}$
$x_2$	9	0	1	$\frac{7}{16}$	$1\frac{11}{16}$	0	$\frac{5}{16}$
$x_5$	19	0	0	$-2\frac{3}{16}$	$1\frac{9}{16}$	1	$\frac{7}{16}$
$z$	24	0	0	$8\frac{3}{4}$	$2\frac{3}{4}$	0	$\frac{1}{4}$



Ответ:  $z_{max} = 24$ . Координаты точки максимума:  $x_1 = 1, x_2 = 9, x_3 = 0, x_4 = 0, x_5 = 19, x_6 = 0$ . Результаты ручных вычислений совпали с результатами выполнения программы.

**Вывод:** в ходе лабораторной работы разработали и отладили программу, находящую оптимальное решение в системе линейных уравнений для целевой функции, и использующей симплекс метод.