

# ВВЕДЕНИЕ

Метод контурных токов (МКТ) - это аналитический метод, используемый для анализа сложных электрических цепей. Он основан на применении закона Кирхгофа для напряжений (второй закон Кирхгофа) и позволяет найти контурные токи в цепи.

Контурные токи представляют собой токи, которые циркулируют в каждом замкнутом контуре цепи. Зная эти токи, можно определить токи в каждом элементе цепи, напряжения на элементах, мощности и другие параметры.

В данной лабораторной работе требуется изучить и реализовать метод контурных токов.

Для достижения поставленных целей лабораторной работы необходимо выполнить следующие задачи:

- рассмотреть и изучить алгоритм метода контурных токов;
- реализовать метод контурных токов;
- составить блок-схему алгоритма;
- рассчитать трудоемкость алгоритма;
- экспериментально проверить работу алгоритма;
- на основании проделанной работы сделать выводы.

# 1 Аналитическая часть

## 1.1 Метод контурных токов

Метод контурных токов (МКТ) – это один из методов анализа электрических цепей, основанный на представлении цепи в виде системы контуров и использовании второго закона Кирхгофа для напряжений. В МКТ предполагается, что ток в каждом контуре образует замкнутый путь и может быть выражен через неизвестные контурные токи [5, с.22].

МКТ применяется для анализа сложных электрических цепей, состоящих из нескольких элементов (сопротивлений, источников тока и напряжения) и связанных узлов. Он позволяет найти значения контурных токов в цепи и другие электрические параметры.

Идея МКТ основана на применении закона Кирхгофа для напряжений и использовании контурных токов в качестве неизвестных переменных. Это позволяет упростить анализ сложных цепей и получить систему уравнений, которая может быть решена для определения токов в каждом контуре [4].

Надо заметить, что принципы МКТ могут быть применены не только в электрических цепях, но и в других областях, где возникают подобные системы сетей или потоков. Далее приведены примеры применения МКТ не в электрических цепях.

Гидравлические системы: МКТ может использоваться для анализа и моделирования потоков жидкости или газа в трубопроводах и системах гидравлического привода. Контурные токи могут соответствовать потокам жидкости или газа в различных частях системы, и МКТ позволяет определить распределение давления и скорости потоков.

Тепловые системы: в тепловых сетях и системах МКТ может быть использован для анализа тепловых потоков и распределения температур. Аналогично электрическим цепям, контурные токи в тепловых системах могут соответствовать потокам тепла в различных компонентах системы.

Сети передачи данных: в компьютерных сетях и телекоммуникационных системах МКТ может быть применен для

анализа потоков данных и определения нагрузки на различных узлах и линиях связи. Контурные токи могут соответствовать объемам передаваемых данных или интенсивности трафика [4].

Транспортные сети: МКТ может быть использован для моделирования потоков транспортных средств в дорожных сетях или системах общественного транспорта. Контурные токи могут представлять потоки транспортных средств по различным маршрутам и помогать оптимизировать распределение трафика или планировать маршруты [4].

МКТ является мощным методом анализа потоков и сетей, и его принципы могут быть применены в различных областях, где требуется моделирование и оптимизация потоковых процессов.

## **Вывод**

На данном этапе был изучен метод контурных токов. Кратко описаны идея и принцип работы алгоритма метода контурных токов, а также области его применения.

## **2 Конструкторская часть**

### **2.1 Разработка алгоритма**

В соответствии с аналитической частью можно составить по шагам теоретическое описание алгоритма.

Идентификация узлов и контуров: анализируется электрическая цепь и определяются все узлы и контуры. Узлы – это точки, в которых сходятся или разделяются элементы цепи, контуры – это замкнутые пути, которые проходят через элементы цепи без повторений.

Присвоение переменных контурным токам: для каждого контура вводятся переменные, обозначающие контурные токи. Количество переменных контурных токов равно количеству контуров в цепи.

Составление уравнений контурных токов: с использованием закона Кирхгофа для напряжений, второго закона Кирхгофа, составляются уравнения для каждого контура. Уравнения строятся вокруг каждого контура, и в них учитываются сопротивления элементов цепи, напряжения источников и направления контурных токов [5, с.23].

Решение системы уравнений: полученная система уравнений, представляющая контурные токи, решается для определения неизвестных переменных контурных токов методом Гаусса.

Определение других параметров: зная значения контурных токов, можно определить токи в каждом элементе цепи, напряжения на элементах, мощности и другие параметры. Эти значения могут быть вычислены с использованием законов Кирхгофа для токов, первого закона Кирхгофа, и законов Ома.

Проверка и интерпретация результатов: полученные значения контурных токов и других параметров цепи проверяются на соответствие физическим законам и ожидаемым результатам. Результаты анализа могут быть интерпретированы в контексте функционирования электрической цепи.

## 2.2 Трудоемкость алгоритма

Для последующего вычисления трудоемкости алгоритмов необходимо ввести модель вычислений:

- 1) Пусть: +, -, /, %, ==, !=, <, >, <=, >=, [], \*, ++, -- – трудоемкость 1;
- 2) Трудоемкость оператора выбора if условие then A else B рассчитывается как на рисунке 1;

$$f_{if} = f_{\text{условия}} + \begin{cases} f_A, & \text{если условие выполняется,} \\ f_B, & \text{иначе.} \end{cases}$$

Рисунок 1 - Трудоемкость оператора выбора if условие then A else B

- 3) Трудоемкость цикла for рассчитывается как на рисунке 2;

$$f_{for} = f_{\text{инициализации}} + f_{\text{сравнения}} + N(f_{\text{тела}} + f_{\text{инициализации}} + f_{\text{сравнения}})$$

Рисунок 2 - Трудоемкость оператора for

- 4) Трудоемкость вызова функции равна 0;
- 5) Трудоемкость заполнения матрицы коэффициентов А равна  $O(n^2)$ ;
- 6) Трудоемкость решения СЛАУ методом Гаусса равна  $O(n^3)$ ;
- 7) Трудоемкость вывода в конце алгоритма равна  $O(n)$ .

Произведём вычисление трудоёмкости алгоритма метода контурных токов:  $O(n^3) + O(n^2) + O(n) = O(n^3)$

Таким образом, полная трудоемкость алгоритма МКТ составляет  $O(n^3)$ .

## Вывод

С учетом теоретических данных, полученных из аналитического раздела, составлено теоретическое пошаговое описание схемы метода контурных токов. Кроме того, проведена теоретическая оценка трудоемкости МКТ.

## **3 Экспериментальная часть**

### **3.1 Требования к ПО**

К программе предъявляется ряд требований:

- на вход подаётся массив матрицы смежности чисел типа double и массив чисел вектора контурных ЭДС;
- на выходе — массив типа double контурных токов и массив типа double всех токов.

### **3.2 Технические характеристики**

Технические характеристики устройства, на котором происходило тестирование:

- Операционная система: Windows 10 Pro 64 бит
- Память: 16 ГБ
- Процессор: Intel Core Pentium G4560 3.5 ГГц

Тестирование проводилось на ПК, включенном в сеть электропитания. Во время тестирования ПК был нагружен только встроенными приложениями окружения рабочего стола, окружением рабочего стола, а также непосредственно системой тестирования.

### **3.3 Тестирование программ**

В таблице 1 приведены функциональные тесты для программы, реализующей алгоритм МКТ – все тесты были пройдены успешно.

Таблица 1 – Тестирование программы алгоритма МКТ

Номер теста	Входной массив	Выходной массив	Ожидаемый результат
1	$\{\{0, 1, 0, 1\}$ $\{1, 0, 1, 0\},$ $\{0, 1, 0, 1\},$ $\{1, 0, 1, 0\}\};$ $\{7, -4, 8\}$	Loop currents: 5.25 3.5 5.75 All currents: $I_{00} = 5.25$ $I_{11} = 3.5$ $I_{22} = 5.75$ $I_{01} = -1.75$ $I_{02} = 0.5$ $I_{12} = 2.25$	Loop currents: 5.25 3.5 5.75 All currents: $I_{00} = 5.25$ $I_{11} = 3.5$ $I_{22} = 5.75$ $I_{01} = -1.75$ $I_{02} = 0.5$ $I_{12} = 2.25$
2	$\{\{0, 1, 1, 1\}$ $\{1, 0, 1, 1\},$ $\{1, 1, 0, 1\},$ $\{1, 1, 1, 0\}\};$ $\{10, 7, -4\}$	Loop currents: 2.25 3.25 0.5 All currents: $I_{00} = 2.25$ $I_{11} = 3.25$ $I_{22} = 0.5$ $I_{01} = 1$ $I_{02} = -1.75$ $I_{12} = -2.75$	Loop currents: 2.25 3.25 0.5 All currents: $I_{00} = 2.25$ $I_{11} = 3.25$ $I_{22} = 0.5$ $I_{01} = 1$ $I_{02} = -1.75$ $I_{12} = -2.75$
3	{Пустой массив}	Error: Adjacency matrix is empty.	Error: Adjacency matrix is empty.

На рисунках 3 и 4 показаны возможные варианты схемы электрических цепей из тестирования.

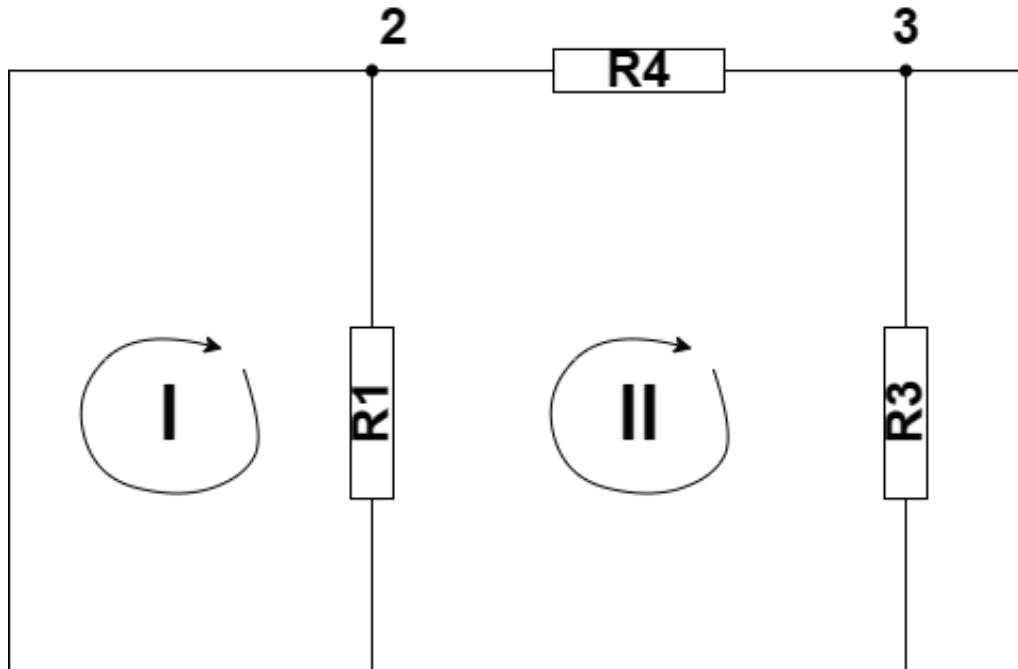


Рисунок 3 – возможный вариант схемы цепи №1 из тестирования

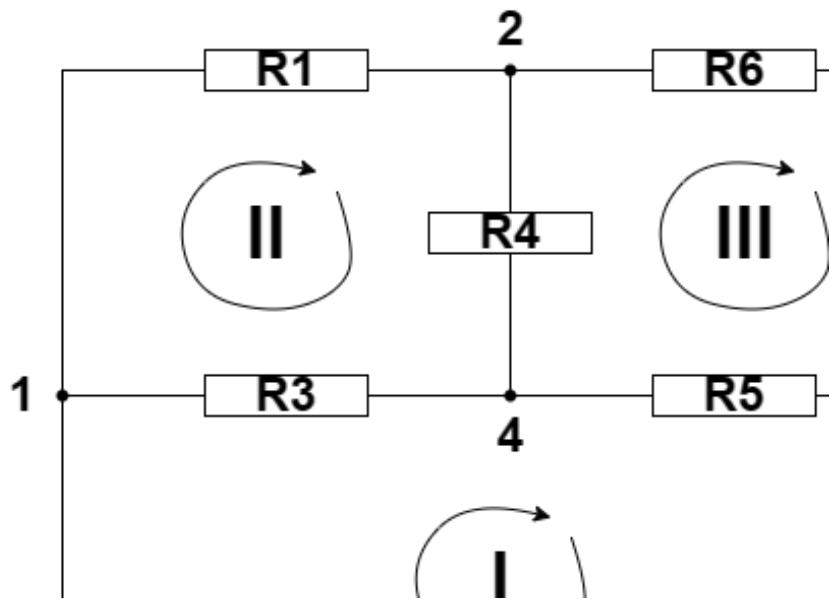


Рисунок 4 – возможный вариант схемы цепи №2 из тестирования

В таблице 3 представлены результаты по тестированию МКТ по времени.

Таблица 2 – Тестирование МКТ по времени выполнения

Количество узлов (размер матрицы)	Время, мкс
3	6.05
4	7.15
5	9.21
6	10.13

Для тестирования по времени была выбрана матрица смежности, заполненная единицами, исключая главную диагональ матрицы.

## Вывод

В результате экспериментальной части был успешно реализован алгоритм метода контурных токов: в результате работы программы выводятся контурные токи и отдельно все токи в целом.

Граф цепи задается матрицей смежности, поэтому ввиду отсутствия ограничений было принято решение для удобства пользования реализовать матрицу смежности и вектор контурных ЭДС с помощью структуры `vector`.

По результатам временных тестов прослеживается довольно четкая зависимость между размером матрицы смежности и временем выполнения алгоритма.

## **ЗАКЛЮЧЕНИЕ**

В процессе выполнения лабораторной работы были проведены анализ, подробное изучение, реализация и тестирование алгоритма метода контурных токов, были выполнены следующие задачи:

- рассмотрен и изучен алгоритм метода контурных токов;
- непосредственно реализован метод контурных токов;
- составлено пошаговое описание алгоритма;
- рассчитана трудоемкость алгоритма;
- экспериментально проверена работа алгоритма;
- на основании проделанной работы сделаны выводы.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Вычислительная линейная алгебра с примерами на MATLAB. В.И. Горбаченко, 2011. – 37 с.
2. Роберт Седжвик. Алгоритмы на C++. Фундаментальные алгоритмы и структуры данных. Algorithms in C++. — М.: «Вильямс», 2011. — 87 с.
3. Шаблон <vector> [Электронный ресурс]. Режим доступа:  
<https://learn.microsoft.com/ru-ru/cpp/standard-library/vector?view=msvc-170> (Дата обращения: 03.06.2023).
4. Метод контурных токов [Электронный ресурс]. Режим доступа:  
<https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%BA%D0%BE%D0%BD%D1%82%D1%83%D1%80%D0%BD%D1%8B%D1%85%D1%82%D0%BE%D0%BA%D0%BE%D0%B2> (Дата обращения: 31.05.2023).
5. Методическое пособие в помощь к выполнению домашних заданий по курсу «Электротехника» и «Общая электротехника» с примерами решения задач. Денисова А.В., 2014. – 22 с.
6. С.Пратта. Язык программирования C++. Лекции и упражнения, 2012. – 345 с.

# ПРИЛОЖЕНИЕ А

## Листинг программы

Листинг 1 – Алгоритм метода контурных токов

```
#include <iostream>
#include <numeric>
#include <vector>

using namespace std;

//Метод контурных токов (МКТ)

//Решение СЛАУ
vector<double> solveLinearSystem(vector<vector<double>>& A, vector<double>& b) {
    int n = A.size();

    //Прямой ход метода Гаусса
    for (int k = 0; k < n - 1; k++) {
        for (int i = k + 1; i < n; i++) {
            double factor = A[i][k] / A[k][k];
            for (int j = k + 1; j < n; j++) {
                A[i][j] -= factor * A[k][j];
            }
            b[i] -= factor * b[k];
        }
    }

    //Обратный ход метода Гаусса
    vector<double> x(n);
    x[n - 1] = b[n - 1] / A[n - 1][n - 1];
    for (int i = n - 2; i >= 0; i--) {
        double sum = 0.0;
        for (int j = i + 1; j < n; j++) {
            sum += A[i][j] * x[j];
        }
        x[i] = (b[i] - sum) / A[i][i];
    }

    return x;
}

//contour_currents реализует МКТ, функция принимает два параметра:
//adj_matrix, и loop_EMF
//adj_matrix - матрица смежности цепи, loop_EMF - вектор контурных ЭДС

vector<double> contour_currents(vector<vector<double>> adj_matrix,
                                  vector<double> loop_EMF) {
    int num_nodes = adj_matrix.size(); //Число узлов
    int num_loops = num_nodes - 1; //Число уравнений (контуров) = число
    узлов - 1

    //Формирование системы уравнений
    vector<vector<double>> A(num_loops, vector<double>(num_loops));
    ... (rest of the code)
```

```

//Заполнение матрицы коэффициентов A

    //Если все контурные токи направить по (или против) часовому направлению
    //то все
        //недиагональные элементы будут взяты с отрицательным знаком т.к. через
        //контур i ток будет течь в прямом направлении, а через контур j в
        //обратном

    for (int i = 0; i < num_loops; i++) {
        for (int j = 0; j < num_loops; j++) {
            if (i == j) {
                A[i][j] =
                    accumulate(adj_matrix[i].begin(), adj_matrix[i].end(), 0.0);
            } else {
                A[i][j] = -adj_matrix[i][j];
            }
        }
    }

    //Решение системы уравнений
    vector<double> loop_currents(num_loops);
    loop_currents = solveLinearSystem(A, loop_EMF);

    return loop_currents;
}

int main() {
    //Матрица смежности для цепи в виде вектора векторов
    //Элемент (i, j) отвечает сопротивлению между i-ым и j-ым узлом
    vector<vector<double>> adj_matrix = {
        {0, 1, 0, 1},
        {1, 0, 1, 0},
        {0, 1, 0, 1},
        {1, 0, 1, 0}}; //size-узловая цепь с соответствующими сопротивлениями
    //между узлами

    //Вектор контурных ЭДС (знак ЭДС учтен для выбранного
    ///(по/против часовой) направления контуров)
    vector<double> loop_EMF(adj_matrix.size() - 1);
    loop_EMF = {7, -4, 8}; //Вектор контурных ЭДС для каждого контура в цепи

    try {
        //Проверка на пустоту adj_matrix
        if (adj_matrix.empty()) {
            throw runtime_error("Adjacency matrix is empty.");
        }

        //Проверка на пустоту loop_EMF
        if (loop_EMF.empty()) {
            throw runtime_error("Loop EMF vector is empty.");
        }

        //Векторы не пусты, продолжаем выполнение кода
    }

    } catch (const exception& e) {
        cout << "Error: " << e.what() << endl;
    }

    //Непосредственно расчет контурных токов
    vector<double> result = contour_currents(adj_matrix, loop_EMF);
}

```

```

//ВЫВОД

//Вывод матрицы смежности adj_matrix электрической цепи
cout << "Adjacency matrix: " << endl;
for (vector<double> node : adj_matrix) {
    for (double i : node) {
        cout << i << " ";
    }
    cout << endl;
}

//Вывод вектора контурных ЭДС loop_EMF электрической цепи
cout << "Vector of contour EMF: " << endl;
for (double volt : loop_EMF) {
    cout << volt << " ";
}
cout << endl;

//Вывод контурных токов электрической цепи
cout << "Loop currents:" << endl;
for (double i : result) {
    cout << i << " ";
}
cout << endl;

//Расчет и вывод всех токов в цепи
cout << "All currents:" << endl;
for (int i = 0; i < result.size(); i++) {
    cout << "I_" << i << i << " = " << result[i] << endl;
}
for (int i = 0; i < result.size(); i++) {
    for (int j = 0; j < i; j++) {
        cout << "I_" << j << i << " = " << result[i] - result[j] << endl;
    }
}

return 0;
}

```