Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

КУРСОВАЯ РАБОТА

по дисциплине: «Вычислительная математика»

Оценка параметров электрической цепи

(Вычисление емкости конденсатора в электрической цепи)

Выполнил студент гр. 3530901/90002 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Е. В. Бурков

(подпись)

Принял преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_И. А. Малышев

(подпись)

“ ” 2021 г.

Санкт-Петербург

2021

Содержание

[Задание 3](#_Toc68268906)

[Анализ цепи 5](#_Toc68268907)

[Расчёт недостающих значений характеристик элементов схемы 8](#_Toc68268908)

[Нахождения начальных условий 9](#_Toc68268909)

[Нахождение ёмкости конденсатора 9](#_Toc68268910)

[Среднеквадратичное отклонение и результат решения 10](#_Toc68268911)

[Сравнение построенной модели и наблюдаемого объекта 12](#_Toc68268912)

[Устойчивость системы 13](#_Toc68268913)

[Влияние погрешности начальных данных на точность 14](#_Toc68268914)

[Используемый тип данных и значения погрешностей для процедур 15](#_Toc68268915)

[Вывод 16](#_Toc68268916)

[Список использованных источников 17](#_Toc68268917)

[Приложение 1 18](#_Toc68268918)

# Задание

Показать, что электрическая цепь, приведенная на рисунке 1, описывается системой дифференциальных уравнений:

|  |
| --- |
|  |
| Рис. 1 Электрическая цепь |



Размыкание ключа происходит в момент времени t=0.

При этом  и . Значения  задаются преподавателем. По заданной таблице экспериментальных данных оценить емкость конденсатора С с точностью до 0.05. Воспользоваться подпрограммами FMIN и RKF45. Оценить точность результата и влияние на точность погрешности исходных данных. Значение С лежит в диапазоне от 0.5 до 2 микрофарад. В таблице напряжение  дано в вольтах, а время - в миллисекундах.

|  |  |
| --- | --- |
| t | (Вольт) |
| 0 | -1.000 |
| 0.1 | 7.777 |
| 0.2 | 12.017 |
| 0.3 | 10.701 |
| 0.4 | 5.407 |
| 0.5 | -0.843 |
| 0.6 | -5.159 |
| 0.7 | -6.015 |
| 0.8 | -3.668 |
| 0.9 | 0.283 |
| 1.0 | 3.829 |

Значения  являются решением системы уравнений:

Значение L находится путём нахождения значения определённого интеграла:

Значение является произведением числа на корень заданного уравнения:

где x\*- корень уравнения:

# Анализ цепи

Докажем, что электрическая цепь на рисунке 1 описывается данной в условии системой дифференциальных уравнений. Для этого проанализируем схему после размыкания ключа.

Для вывода соотношений воспользуемся вторым законом Кирхгофа. Формулировка: Алгебраическая сумма ЭДС, действующих в замкнутом контуре, равна алгебраической сумме падений напряжения на всех резистивных элементах в этом контуре.

Здесь термин «алгебраическая сумма» означает, что как величина ЭДС так и величина падения напряжения на элементах может быть как со знаком «+», так и со знаком «-». При этом определить знак можно по следующему алгоритму:

1. Выбираем направление обхода контура (два варианта либо по часовой, либо против).

2. Произвольно выбираем направление токов через элементы цепи.

3. Расставляем знаки для ЭДС и напряжений, падающих на элементах по правилам:

-ЭДС, создающие ток в контуре, направление которого совпадает с направление обхода контура записываются со знаком «+», в противном случае ЭДС записываются со знаком «-».

- напряжения, падающие на элементах цепи записываются со знаком «+», если ток, протекающий через эти элементы, совпадает по направлению с обходом контура, в противном случае напряжения записываются со знаком «-».

Выделим в нашей схеме 2 контура и распишем для них данный закон:

Внешний контур:

|  |
| --- |
|  |
| Рис. 2 Первый контур |

В результате обхода получаем следующее соотношение:

, где:

– ЭДС самоиндукции при изменении тока в катушке

Вынеся дифференциал в левую часть уравнения, а остальные величины в правую получим:

Внутренний контур:

|  |
| --- |
|  |
| Рис. 3 Второй контур |

В результате обхода получаем следующее соотношение:

, лишь направление тока в катушке отличилось от направления обхода контура. Выпишем значения аналогично с первым контуром:

– ЭДС самоиндукции при изменении тока в катушке

Отсюда получаем:

Соотношение для тока на конденсаторе:

В ходе анализа цепи мы получили следующую систему дифференциальных уравнений, которое справедливо для данной нам электрической цепи:

Произведя тривиальные математические преобразования, можно убедиться, что данная система совпадает с данной в условии. Это означает что электрическая цепь, приведенная на рисунке 1, описывается системой дифференциальных уравнений из условия.

# Расчёт недостающих значений характеристик элементов схемы

Для нахождения значений была написана процедура findParams(), которая после своего выполнения устанавливает полученные значения в глобальные переменные.

Для нахождения значений , которые являются решением СЛАУ применим процедуру DECOMP (для LU разложения матрицы) и процедуру SOLVE для решения системы. Для нахождения значения L была применена процедура QUANC8 (адаптивные квадратурные формулы Ньютона-Котеса 8 степени). Для расчёта , значение которого является корнем уравнения была использована процедура Zeroin (данная процедура представляет собой сочетание методов бисекции и обратной квадратичной интерполяции для нахождения корня уравнения на заданном отрезке).

После вызова findParams() можно использовать printParams() и посмотреть на значения характеристик элементов схемы (рисунок 1).

|  |
| --- |
|  |
| Рис. 4 Значение характеристик элементов схемы после вызова findParams() |

# Нахождения начальных условий

Для решения данного нам дифференциального уравнения нам необходимо определить величины . Их значения можно узнать, обратившись к схеме. До размыкания ключа конденсатор имел заряд. Следовательно ток через него в начальный момент времени равен нулю ().

Исходя из того, что ключ был замкнут, то ток через не протекал (а проходил через участок цепи с более малым сопротивлением). В начальный момент времени, когда ключ разомкнули, на элементах появилось напряжение, но из-за инерции индуктивности . Следовательно, ток шёл только через левый контур и можно найти по закону Ома:

Теперь имея начальные данные для дифференциального уравнения, можно приступить к его решению.

# Нахождение ёмкости конденсатора

В прошлом пункте мы нашли начальные значения для дифференциального уравнения третьего порядка, которое описывает электрическую цепь. Теперь нам необходимо найти такое значение ёмкости конденсатора, чтобы решение дифференциального уравнения было максимально приближено к зависимости, снятой с объекта. Для этого была написана функция getDelta(x). Данная функция считает среднеквадратичное отклонение значений, полученных в ходе решения системы дифференциальных уравнений с заданной ёмкостью конденсатора, и значений, снятых с объекта. Для решения системы дифференциальных уравнений использовалась процедура RKF45.

# Среднеквадратичное отклонение и результат решения

Теперь, когда у нас есть функция, которая показывает, насколько близки зависимости можно построить график зависимости среднеквадратичного отклонения от ёмкости конденсатора. График представлен ниже:

|  |
| --- |
|  |
| Рис. 5 График среднеквадратичного отклонения |

По графику видно, что наименьшее среднеквадратичное отклонение находится в области 1 микрофарада. Использовав процедуру FMIN на необходимом промежутке, можно получить более точное значение ёмкости конденсатора, при котором квадратичное отклонение будет минимально. В ходе работы программы были получены следующие значения:

|  |
| --- |
|  |
| Рис. 6 Результаты работы программы |

При работе процедуры RKF45 печатался флаг, и каждый раз его значение было равно двум, что указывает на “нормальную” работу функции.

Исходя из результатов работы функции FMIN можно сказать, что ёмкость конденсатора равняется **одному микрофараду**.

# Сравнение построенной модели и наблюдаемого объекта

Для наглядности полученных данных приведём график, на котором красным цветом показана зависимость напряжения от времени для нашей модели (система дифференциальных уравнений) , а синим данные, снятые с объекта. Для модели было выбрано 100 точек с интервалом секунд, значения в которых считала процедура RKF45. А для экспериментальных данных точки соединены прямыми. Рисунок ниже показывает, что наша модель довольно неплохо описывает объект электрической цепи.

|  |
| --- |
|  |
| Рис. 7 Сравнение данных, полученных с объекта и модели |

# Устойчивость системы

Для наблюдения устойчивости системы проведём следующее исследование. Возьмём данные, снятые с реального объекта, полученные данные с модели. Далее выведем с учётом того, что значения уменьшены на 1% и назовём данные . Так же сделаем для C и выведем .

|  |
| --- |
|  |
| Рис. 7 Исследование устойчивости системы |

По рисунку выше видно, что значения U отклонились не более чем на 3%. Только при t = 0.9 мс данные с возмущенными значениями отличаются сильно от экспериментальных. Из этого следует, что полученная модель достаточно устойчива, а матрица системы скорей всего хорошо обусловлена.

# Влияние погрешности начальных данных на точность

Проведём исследования, насколько изменения начальных данных повлияет на решение задачи. Предположим, что вольтметр при измерениях выдал нам не 3 знака после запятой а меньше. В таблице ниже отражена зависимость ответа от количества знаков после запятой:

Табл. 1 Зависимость ответа от знаков после запятой

в экспериментальных данных

|  |  |
| --- | --- |
| Количество знаков после запятой в | Ёмкость конденсатора C, мкФ |
| 3 | 1.00263671875 |
| 2 | 1.00263671875 |
| 1 | 1.00263671875 |
| 0 | 0.99681640625 |

Посмотрим, как это повлияет на график зависимости (рисунок 8).

|  |
| --- |
|  |
| Рис. 8 Сравнение полученных моделей |

По таблице 1 и рисунку 8 видно, что количество знаков после запятой особо не влияют на ответ и полученную модель электрической цепи. Максимальное различие между значениями напряжения в разных моделях оценивается как 0.11 В с t = 0.00094 сек. Это является хорошим результатом. Несмотря на сокращения количества значащих разрядов в данных из условия, решение всё равно осталось качественно верным.

# Используемый тип данных и значения погрешностей для процедур

До этого пункта во всех модулях программы использовался тип double, который предоставляет нам двойную точность (14 значащих разрядов). Давайте изменим тип на float (одинарная точность) и посмотрим, как изменятся результаты.

Ёмкость конденсатора при использовании double = 0.00000100263672;

Ёмкость конденсатора при использовании float = 0.00000100263662;

Можно заметить, что результаты не сильно отличаются друг от друга.

После всех расчётов результат во float отличается от результата в double на 8 значащем разряде, что для него допустимо.

Ранее в каждой процедуре использовалось одно значение для границ относительной и абсолютной погрешностей равное “Поиграемся” с данным значением и посмотрим, как изменятся результаты.

Табл. 2 Зависимость ответа от значений границ погрешностей

|  |  |
| --- | --- |
| RE и AE | Ёмкость конденсатора C, мкФ |
|  | 1.00263671875 |
|  | 1.00263671875 |
|  | 0.99681640625 |
|  | 1.00845703125 |

Границы погрешностей качественно не изменили наше решение, ёмкость конденсатора так и осталась близкой к 1 мкФ.

# Вывод

В ходе выполнения данной работы была произведена параметрическая аппроксимация объекта электрической цепи. Используя законы физики, было доказано, что цепь действительно описывается системой дифференциальных уравнений. В результате расчётов было получено, что ёмкость конденсатора равна **1 микрофараду**. При данном значении ёмкости отклонение данных, снятых с модели, от данных, снятых с объекта, не больше 0.021 вольт, что является хорошим приближением. Была исследована зависимость решения от точности начальных данных и от границ погрешностей функций для вычисления. Во время выполнения работы не возникало проблем с вычислениями на компьютере. Все процедуры из библиотеки Форсайта проработали в “нормальном” режиме. Я считаю это связано с тем, что условия задачи специально подобраны так, чтобы лишить исполнителя многих проблем.

# Список использованных источников

1. ГОСТ 7.32-2001. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления

2. Устинов С. М., Зимницкий В. А. - Вычислительная математика, 2009

# Приложение 1

Листинг программ на языке C++ и Python представлен ниже:

|  |
| --- |
| #include <iostream> #include "main.h" #include <cmath> #include <fstream> #include <iomanip>  double R; double R1; double R2; double R3; double E1; double E2; double L; double L1; double L3; double C; double C\_result; double U\_exp[11] = {  -1.000,  7.777,  12.017,  10.701,  5.407,  -0.843,  -5.159,  -6.015,  -3.668,  0.283,  3.829 };  double abserr = 1e-8; double relerr = 1e-8;  double quancFun(double x) {  return (log(1 + x)) / (1 + x \* x); }  double zeroinFun(double x) {  return exp(x) - 2 \* (x - 1) \* (x - 1); }  void rkfFun(double t, double \*y, double \*dy) {  dy[0] = (E1 - E2 - y[2] + (y[1] \* R2) - (y[0] \* (R + R2))) / L1;  dy[1] = (E2 + y[2] + (y[0] \* R2) - (y[1] \* (R + R2))) / L3;  dy[2] = (y[0] - y[1]) / C; }   void findParams() {   **MATRIX**(Q);   Q[0][0] = 16.0;  Q[0][1] = -18.0;  Q[0][2] = 24.0;  Q[1][0] = -18.0;  Q[1][1] = 49.0;  Q[1][2] = -42.0;  Q[2][0] = 24.0;  Q[2][1] = -42.0;  Q[2][2] = 46.0;   **VECTOR**(B, 3);  B[0] = 304.0;  B[1] = 218.0;  B[2] = 166;  **INTVECTOR**(IPTV, ndim);  **VECTOR**(WORK, ndim);  double cond;  decomp(**ndim**, Q, &cond, IPTV, WORK);  solve(**ndim**, Q, B, IPTV);  R = B[0];  R2 = B[1];  E2 = B[2];  R1 = R;  R3 = R;  int nofun;  double errest;  double flag = 1;  quanc8(quancFun, 0, 1.0, abserr, relerr, &L, &errest, &nofun, &flag);  L = L \* 0.1469517;  L1 = L;  L3 = L;  double x1 = -1000.0;  double x2 = 1000.0;  E1 = Zeroin(zeroinFun, x1, x2, relerr);  E1 = E1 \* 18.75217;  }  double test(double x) {  return x \* x - 3 \* x; }  void printParams() {  printf("R=%lf\n", R);  printf("R1=%lf\n", R1);  printf("R2=%lf\n", R2);  printf("R3=%lf\n", R3);  printf("L=%lf\n", L);  printf("L1=%lf\n", L1);  printf("L3=%lf\n", L3);  printf("E1=%lf\n", E1);  printf("E2=%lf\n", E2); }  double getDelta(double x) {  double y[3] = {E1 / R1, 0.0, -1.0};  double t\_end = 0;  double work[21];  int iwork[10];  int flag = 1;  double delta = 0.0;  C = x;  for (int i = 0; i < 11; ++i) {  double current\_t = i \* 0.1 / 1000;  RKF45(rkfFun, 3, y, &t\_end, &current\_t, &relerr, &abserr, &flag, work, iwork);  delta = delta + pow(U\_exp[i] - y[2], 2);  }  *// std::cout << "Solving a differential equation with C=" << x <<  // " value of the RKF45 flag is " << flag << "\n";* return sqrt(delta / 11); }  void calculateC() {  double count = 0;  int flag = 0;  C\_result = (fmin(0.5e-6, 2e-6, getDelta, 1e-8, count, flag));  std::cout << std::fixed << std::setprecision(20) << "C\_result = " << " " << C\_result << "\n"; }  void writeOutput() {  std::ofstream outf("output.txt");  double y[3] = {E1 / R, 0, -1.0};  double t\_end = 0;  double work[21];  int iwork[10];  int flag = 1;  printf("calculate with %lf\n", C);  double current\_t = 0.0;  while (current\_t <= 0.00101) {  RKF45(rkfFun, 3, y, &t\_end, &current\_t, &relerr, &abserr, &flag, work, iwork);  std::cout << std::fixed << std::setprecision(14) << current\_t << " " << y[2] << "\n";  current\_t += 0.00001;  } }    void printResult() {  std::ofstream outf("output.txt");  double y[3] = {E1 / R, 0, -1.0};  double t\_end = 0;  double work[21];  int iwork[10];  int flag = 1;  std::cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";  std::cout << "t(ms) | OBJECT | MODEL | DIFFERENCE\n";  std::cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";  for (int i = 0; i < 11; ++i) {  double current\_t = i \* 0.1 / 1000;  RKF45(rkfFun, 3, y, &t\_end, &current\_t, &relerr, &abserr, &flag, work, iwork);   if (y[2] < 0) {  std::cout << std::fixed << std::setprecision(4) << current\_t << std::setprecision(3) << " | " << U\_exp[i]  << " | " << std::setprecision(7) << y[2] << " | " << std::abs(y[2] - U\_exp[i]) << " " << "\n";  } else {  if (y[2] > 10.0) {  std::cout << std::fixed << std::setprecision(4) << current\_t << std::setprecision(3) << " | "  << U\_exp[i]  << " | " << std::setprecision(7) << y[2] << " | " << std::abs(y[2] - U\_exp[i]) << " " << "\n";  } else {  std::cout << std::fixed << std::setprecision(4) << current\_t << std::setprecision(3) << " | "  << U\_exp[i]  << " | " << std::setprecision(7) << y[2] << " | " << std::abs(y[2] - U\_exp[i]) << " "  << "\n";  }   }   }  std::cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"; }  void writeDeltaOutput() {  std::ofstream outf("DeltaOutput.txt");  C = 0.5e-6;  while (C < 2e-6) {  outf << std::fixed << std::setprecision(14) << C << " " << getDelta(C) << "\n";  C += 0.01e-6;  } }   int main() {   findParams();  printParams();  calculateC();  C = C\_result;  printResult(); } |

|  |
| --- |
| import codecs import re import matplotlib.pyplot as plt import pandas as pd   def Make\_plot(name, r\_dict, exp\_dict):  frame = pd.DataFrame({'T': [float(item) for item in r\_dict.keys()]})  frame["U"] = [float(item) for item in r\_dict.values()]  frame["U2"] = exp\_dict.values()  df = frame  plt.title(name, loc='center', fontsize=12,  fontweight=0, color='red')  plt.style.use('fivethirtyeight')  plt.plot(df['T'], df['U'], marker='', color="red", linewidth=1, alpha=0.9, label="Model")  plt.plot(df['T'], df['U2'], marker='', color="blue", linewidth=1, alpha=0.9, label="Experimental")  plt.xlabel("T, seconds")  plt.ylabel("Uc, volt ")  plt.legend()  plt.show()   if \_\_name\_\_ == '\_\_main\_\_':  array = [-1.000,  7.777,  12.017,  10.701,  5.407,  -0.843,  -5.159,  -6.015,  -3.668,  0.283,  3.829]  t = 0.0  exp\_data = {}  for i in range(10):  diff = array[i+1] - array[i]  print(t)  for j in range(10):  exp\_data[t] = array[i] + j / 10 \* diff  t = t + 0.00001  reader = codecs.open("output.txt")  reader.readline()  next\_line = reader.readline()  data = {}  while not next\_line.startswith("end"):  groups = re.match("(-?\w+\.\w+|\w+) (-?\w+\.\w+|\w+)", next\_line).groups()  data[groups[0]] = groups[1]  next\_line = reader.readline()  Make\_plot("Comparing model and object", data, exp\_data) |