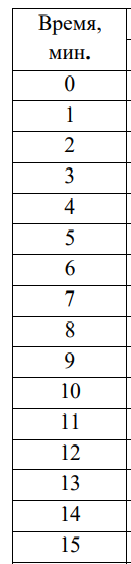
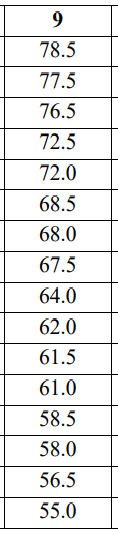
1. **Формулировка задания.**

Пусть проводятся серии экспериментов, в которых в определенные моменты времени фиксируется температура кофе, налитого в керамическую чашку, и остывающего на воздухе.

Кофе находится в стакане из материала высокой теплопроводностью, помещенной в керамический стакан (температура кофе регистрировалась с точностью до 0.5ºС, температура окружающего воздуха равна 22ºС). Требуется на основе компьютерного моделирования проверить справедливость применимости закона теплопередачи (8.6) к этому процессу и определить экспериментальное значение коэффициента r.

Указания: для решения этой задачи рекомендуется следовать следующему алгоритму.

1) Полагая, что данный эксперимент описывается моделью (8.6), и используя соотношения (8.7), напишите программу, позволяющую по полученным экспериментальным данным, приведенным в таблице 8.2 (варианты задания), определить значение параметра r закона теплообмена Ньютона. При выполнении этого задания воспользуйтесь методом наименьших квадратов, применив его к логарифму разности T(t)–TS из соотношения (8.7). Предложите, как исходя из соотношения (8.7) и данных эксперимента оценить предполагаемое начальное значение параметра r, и в соответствии с вариантом задания выполните эту оценку. Затем напишите код программы, которая, варьируя параметр r, определяет при каком его значении сумма квадратов отклонений данных эксперимента от ln(T(t)–TS) минимальна. Полученное таким образом значение r используйте затем для моделирования процесса остывания кофе.

2) С целью определения справедливости модели (8.6) для полученного значения параметра r напишите программу, моделирующую процесс теплопередачи (8.6) путем численного решения дифференциального уравнения первого порядка. Для этого используйте код программы, приведенный на рисунке 8.4, внеся необходимые изменения в обозначения переменных и код программы. Задайте следующие переменные и начальные условия: t = 0 – начальное время (мин), T = 83 – начальная температура кофе (ºС), room\_T – комнатная температура (ºС), r = – коэффициент остывания (1/мин), определенный в п.1. dt = 0.01 – шаг по времени (мин), tmax – длительность моделирования (мин), n – общее количество шагов.

3) Модифицируйте программу так, чтобы исходное значение параметра r и начальные условия можно было вводить с клавиатуры.

4) Один из методов определения точности численного решения заключается в повторении вычислений с меньшим шагом и сравнении результатов (метод Рунге-Кутта). Если решения в обоих результатах совпадают с заданной точностью, то можно ограничиться достигнутым шагом разбиения. Исходя из этих соображений, убедитесь, что выбранное значение величины шага по времени достаточно мало и не оказывает влияние на получаемую зависимость температуры от времени. Предложите способы тестирования правильности работы программы.

5) Используя дополнительный вложенный цикл, модернизируйте программу, так чтобы полученные результаты выводились в консоль и сохранялись в текстовом файле.

6) Вывод в текстовый файл можно реализовать.

7) Модифицируйте программу так, чтобы значения параметров и начальных условий можно было считывать из текстового файла инициализации.

8) Модернизируйте разработанный программный код так, чтобы значение интегрального коэффициента теплоотдачи определялось автоматически. Для этого используйте код программы, полученной в п.1) как подпрограмму.

9) С целью исследования применимости модели сравните полученные результаты моделирования с аналитическим решением. Для этого импортируйте данные численного моделирования п.2) в текстовый файл, а затем в электронных таблицах Excel постройте графики данных опыта из таблицы 8.2, а также численного и аналитического решения задачи об остывании кофе. Оформите надлежащим образом график в Excel и импортируйте его в документ Word. Пример оформления: см. рис.8.4 и 8.5.

1. **Выполнение работы по пунктам.**

1) Код программы:

// Подключение потоков ввода-вывода

#include <iostream>

// Подключение потоков файлового ввода-вывода

#include <fstream>

// Подюключение математических функций

#include <cmath>

// Подключение форматирование вывода

#include <iomanip>

// Подключение класса цепочки символов

#include <string>

int main() {

// Установка русской локализации

setlocale(LC\_ALL, "Russian");

// Считывание исходных данных из файла

std::fstream file("data.txt");

double Ts;

file >> Ts;

int N;

file >> N;

double\* t = new double[N];

double\* T = new double[N];

for (int i = 0; i < N; i++) {

file >> t[i] >> T[i];

}

// Вычисление эмперического коэффициента методом наименьших квадратов

double A = 0, B = 0, C = 0, D = 0;

for (int i = 0; i < N; i++) {

A += std::pow(t[i], 2);

B += t[i];

C += std::log(T[i] - Ts) \* t[i];

D += std::log(T[i] - Ts);

}

double r1 = -(C \* N - B \* D) / (A \* N - std::pow(B, 2));

double cnst = (A \* D - B \* C) / (A \* N - std::pow(B, 2));

std::cout << "Значение r = " << r1 << " методом наименьших квадратов\n";

// Вычисление эмперического коэффициента методом полного перебора

double r2 = -1, mn = 1e9, eps = 1e-3;

for (double i = 0; i <= 100; i += eps) {

double sum = 0;

for (int j = 0; j < N; j++) {

sum += std::pow(-i \* t[j] + std::log(T[0] - Ts) - std::log(T[j] - Ts), 2);

}

if (sum < mn) {

mn = sum;

r2 = i;

}

}

std::cout << "Значение r = " << r2 << " методом перебора\n";

std::cout << "Моделирование процесса остывания:\n";

std::cout << std::right << std::fixed << std::setprecision(3);

std::cout << std::setw(5) << "t[i]" << ' ' << std::setw(15) << "T[i](исходное)" << ' ' << std::setw(15) << "T[i](м.н.к)" << ' ' << std::setw(15) << "T[i](перебор)" << '\n';

for (int i = 0; i < N; i++) {

std::cout << std::setw(5) << (t[i] < 10 ? "0" : "") + std::to\_string(int(t[i])) << ' ' << std::setw(15) << T[i] << ' ' << std::setw(15) << Ts + std::exp(-r1 \* t[i] + cnst) << ' ' << std::setw(15) << Ts - (Ts - T[0]) \* std::exp(-r2 \* t[i]) << '\n';

}

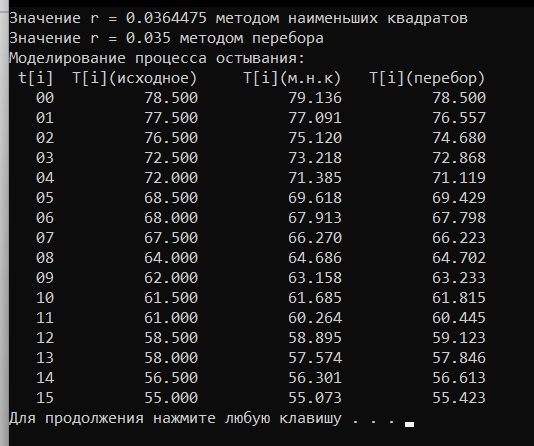
delete[] t;

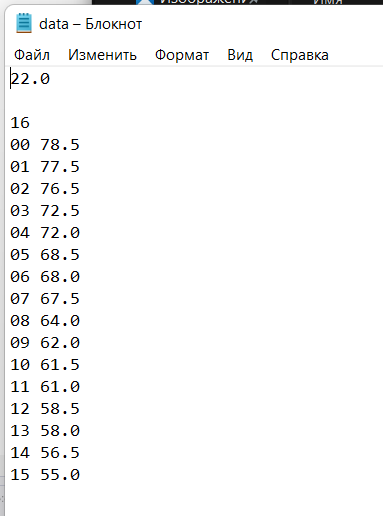
delete[] T;

system("pause");

return 0;

}

Скриншоты:

2) Код программы:

// Подключение потоков ввода-вывода

#include <iostream>

// Подключение форматирование вывода

#include <iomanip>

// Подключение класса цепочки символов

#include <string>

// Подключение потоков файлового ввода-вывода

#include <fstream>

int main() {

// Установка русской локализации

setlocale(LC\_ALL, "Russian");

// Установка начальных значений

double t = 0.0, T = 83.0;

double roomT = 22.0;

double r = 0.0364475;

double dt = 0.01, tmax = 0.2;

int n = tmax / dt;

std::cout << std::right << std::fixed << std::setprecision(3);

std::cout << std::setw(6) << "i(шаг)" << ' ' << std::setw(8) << "t(время)" << ' ' << std::setw(14) << "T(температура)" << ' ' << std::setw(23) << "Tnew(новая температура)" << '\n';

for (int i = 0; i <= n; i++) {

double cT = -r \* (T - roomT) \* dt;

std::cout << std::setw(6) << (i < 10 ? "0" : "") + std::to\_string(i) << ' ' << std::setw(8) << t << ' ' << std::setw(14) << T << ' ' << std::setw(23) << T + cT << '\n';

T += cT; t += dt;

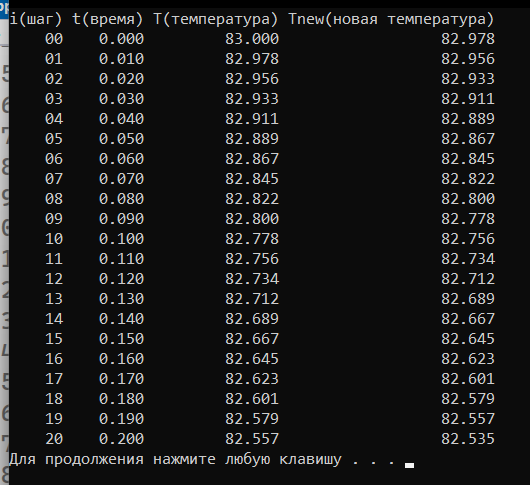
}

system("pause");

return 0;

}

Скриншоты:



3) Код программы:

// Подключение потоков ввода-вывода

#include <iostream>

// Подключение форматирование вывода

#include <iomanip>

// Подключение класса цепочки символов

#include <string>

// Подключение потоков файлового ввода-вывода

#include <fstream>

int main() {

// Установка русской локализации

setlocale(LC\_ALL, "Russian");

double t, T;

double roomT;

double r;

double dt, tmax;

std::cout << "Введите начальное время: ";

std::cin >> t;

std::cout << "Введите начальную температуру кофе: ";

std::cin >> T;

std::cout << "Введите комнатную температуру: ";

std::cin >> roomT;

std::cout << "Введите коэффициент остывания: ";

std::cin >> r;

std::cout << "Введите шаг по времени: ";

std::cin >> dt;

std::cout << "Введите длительность моделирования: ";

std::cin >> tmax;

int n = tmax / dt;

std::cout << std::right << std::fixed << std::setprecision(3);

std::cout << std::setw(6) << "i(шаг)" << ' ' << std::setw(8) << "t(время)" << ' ' << std::setw(14) << "T(температура)" << ' ' << std::setw(23) << "Tnew(новая температура)" << '\n';

for (int i = 0; i <= n; i++) {

double cT = -r \* (T - roomT) \* dt;

std::cout << std::setw(6) << (i < 10 ? "0" : "") + std::to\_string(i) << ' ' << std::setw(8) << t << ' ' << std::setw(14) << T << ' ' << std::setw(23) << T + cT << '\n';

T += cT; t += dt;

}

system("pause");

return 0;

}

4) Код программы:

// Подключение потоков ввода-вывода

#include <iostream>

// Подключение форматирование вывода

#include <iomanip>

// Подключение класса цепочки символов

#include <string>

// Подключение потоков файлового ввода-вывода

#include <fstream>

int main() {

// Установка русской локализации

setlocale(LC\_ALL, "Russian");

double t, T;

double roomT;

double r;

double dt, tmax;

std::cout << "Введите начальное время: ";

std::cin >> t;

std::cout << "Введите начальную температуру кофе: ";

std::cin >> T;

std::cout << "Введите комнатную температуру: ";

std::cin >> roomT;

std::cout << "Введите коэффициент остывания: ";

std::cin >> r;

std::cout << "Введите шаг по времени: ";

std::cin >> dt; std::cout << "Введите длительность моделирования: ";

std::cin >> tmax;

int n = tmax / dt;

// Увеличение числа шагов до момента достижения точности

double prev = -1;

double eps = 1e-3;

while (true) {

double curT = T, curt = t;

for (int i = 0; i <= n; i++) {

curT += -r \* (T - roomT) \* dt;

curt += dt;

}

if (prev == -1) {

prev = curT;

continue;

}

if (std::abs(prev - curT) < eps) {

break;

}

dt /= 2;

n \*= 2;

}

std::cout << std::right << std::fixed << std::setprecision(3);

std::cout << std::setw(6) << "i(шаг)" << ' ' << std::setw(8) << "t(время)" << ' ' << std::setw(14) << "T(температура)" << ' ' << std::setw(23) << "Tnew(новая температура)" << '\n';

for (int i = 0; i <= n; i++) {

double cT = -r \* (T - roomT) \* dt;

std::cout << std::setw(6) << (i < 10 ? "0" : "") + std::to\_string(i) << ' ' << std::setw(8) << t << ' ' << std::setw(14) << T << ' ' << std::setw(23) << T + cT << '\n';

T += cT; t += dt;

}

system("pause");

return 0;

}

5) Код программы:

// Подключение потоков ввода-вывода

#include <iostream>

// Подключение форматирование вывода

#include <iomanip>

// Подключение класса цепочки символов

#include <string>

// Подключение потоков файлового ввода-вывода

#include <fstream>

int main() {

// Установка русской локализации

setlocale(LC\_ALL, "Russian");

double t, T;

double roomT;

double r;

double dt, tmax;

std::cout << "Введите начальное время: ";

std::cin >> t;

std::cout << "Введите начальную температуру кофе: ";

std::cin >> T;

std::cout << "Введите комнатную температуру: ";

std::cin >> roomT;

std::cout << "Введите коэффициент остывания: ";

std::cin >> r;

std::cout << "Введите шаг по времени: ";

std::cin >> dt; std::cout << "Введите длительность моделирования: ";

std::cin >> tmax;

int n = tmax / dt;

// Увеличение числа шагов до момента достижения точности

double prev = -1;

double eps = 1e-3;

while (true) {

double curT = T, curt = t;

for (int i = 0; i <= n; i++) {

curT += -r \* (T - roomT) \* dt;

curt += dt;

}

if (prev == -1) {

prev = curT;

continue;

}

if (std::abs(prev - curT) < eps) {

break;

}

dt /= 2;

n \*= 2;

}

std::fstream file("data.txt", std::ios::trunc | std::ios::out);

std::cout << std::right << std::fixed << std::setprecision(3);

std::cout << std::setw(6) << "i(шаг)" << ' ' << std::setw(8) << "t(время)" << ' ' << std::setw(14) << "T(температура)" << ' ' << std::setw(23) << "Tnew(новая температура)" << '\n';

file << std::right << std::fixed << std::setprecision(3);

file << std::setw(6) << "i(шаг)" << ' ' << std::setw(8) << "t(время)" << ' ' << std::setw(14) << "T(температура)" << ' ' << std::setw(23) << "Tnew(новая температура)" << '\n';

for (int i = 0; i <= n; i++) {

double cT = -r \* (T - roomT) \* dt;

std::cout << std::setw(6) << (i < 10 ? "0" : "") + std::to\_string(i) << ' ' << std::setw(8) << t << ' ' << std::setw(14) << T << ' ' << std::setw(23) << T + cT << '\n';

file << std::setw(6) << (i < 10 ? "0" : "") + std::to\_string(i) << ' ' << std::setw(8) << t << ' ' << std::setw(14) << T << ' ' << std::setw(23) << T + cT << '\n';

T += cT; t += dt;

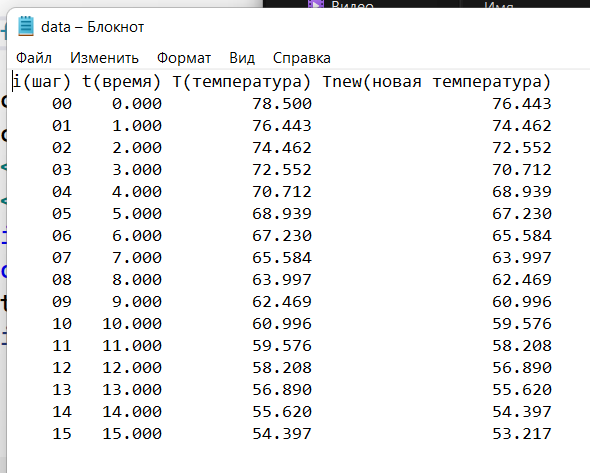
}

system("pause");

return 0;

}

Скриншоты:



6) Код программы:

// Подключение потоков ввода-вывода

#include <iostream>

// Подключение форматирование вывода

#include <iomanip>

// Подключение класса цепочки символов

#include <string>

// Подключение потоков файлового ввода-вывода

#include <fstream>

int main() {

// Установка русской локализации

setlocale(LC\_ALL, "Russian");

double t, T;

double roomT;

double r;

double dt, tmax;

// Считывание из файла

std::fstream fileIn("dataIn.txt");

fileIn >> t;

fileIn >> T;

fileIn >> roomT;

fileIn >> r;

fileIn >> dt;

fileIn >> tmax;

int n = tmax / dt;

// Увеличение числа шагов до момента достижения точности

double prev = -1;

double eps = 1e-3;

while (true) {

double curT = T, curt = t;

for (int i = 0; i <= n; i++) {

curT += -r \* (T - roomT) \* dt;

curt += dt;

}

if (prev == -1) {

prev = curT;

continue;

}

if (std::abs(prev - curT) < eps) {

break;

}

dt /= 2;

n \*= 2;

}

std::fstream fileOut("dataOut.txt");

std::cout << std::right << std::fixed << std::setprecision(3);

std::cout << std::setw(6) << "i(шаг)" << ' ' << std::setw(8) << "t(время)" << ' ' << std::setw(14) << "T(температура)" << ' ' << std::setw(23) << "Tnew(новая температура)" << '\n';

file << std::right << std::fixed << std::setprecision(3);

file << std::setw(6) << "i(шаг)" << ' ' << std::setw(8) << "t(время)" << ' ' << std::setw(14) << "T(температура)" << ' ' << std::setw(23) << "Tnew(новая температура)" << '\n';

for (int i = 0; i <= n; i++) {

double cT = -r \* (T - roomT) \* dt;

std::cout << std::setw(6) << (i < 10 ? "0" : "") + std::to\_string(i) << ' ' << std::setw(8) << t << ' ' << std::setw(14) << T << ' ' << std::setw(23) << T + cT << '\n';

file << std::setw(6) << (i < 10 ? "0" : "") + std::to\_string(i) << ' ' << std::setw(8) << t << ' ' << std::setw(14) << T << ' ' << std::setw(23) << T + cT << '\n';

T += cT; t += dt;

}

system("pause");

return 0;

}

7) График: