#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

double paramA = 4;

double fi1(double t) {

return cos(t);

}

double fi2(double t) {

return sin(t);

}

double psi(double x) {

return 1-x;

}

/\*\*

\* a - диагональ, лежащая под главной (нумеруется: [1;n-1])

\* b - главная диагональ матрицы A (нумеруется: [0;n-1])

\* c - диагональ, лежащая над главной (нумеруется: [0;n-2])

\* f - правая часть (столбец)

\*/

vector<double> solveMatrix(vector<double> a, vector<double> b, vector<double> c, vector<double> f) {

//vector<double> solveMatrix(vector<vector<double>> m, vector<double> f){

int n = a.size();

vector<double> x(n);

double tmp;

for (int i = 1; i < n; i++)

{

tmp = a[i] / b[i - 1];

b[i] = b[i] - tmp \* c[i - 1];

f[i] = f[i] - tmp \* f[i - 1];

}

x[n - 1] = f[n - 1] / b[n - 1];

for (int i = n - 2; i >= 0; i--)

{

x[i] = (f[i] - c[i] \* x[i + 1]) / b[i];

}

return x;

}

vector<vector<double>> solve() {

vector<vector<double>> answer;

vector<double> a;

vector<double> b;

vector<double> c;

vector<double> d;

vector<double> tmp;

double h = 0.1;

double tau = 0.1;

double x = 0;

double tt = pow(paramA, 2) \* tau / pow(h, 2);

for (int i = 0; i < 1 / h +1; ++i) {

a.push\_back(tt);

b.push\_back(-(1 + 2\*tt));

c.push\_back(tt);

d.push\_back(-psi(x));

tmp.push\_back(psi(x));

x += h;

}

b[0] = -1 / h;

c[0] = 1 / h;

d[0] = fi1(tau);

a[a.size() - 1] = 0;

b[b.size() - 1] = 1;

d[d.size() - 1] = fi2(tau);

answer.push\_back(tmp);

answer.push\_back(solveMatrix(a, b, c, d));

double t = tau\*2;

while (t < 1) {

d = answer[answer.size() - 1];

for (int i = 0; i < d.size(); ++i) {

d[i] \*= -1;

}

b[0] = -1 / h;

c[0] = 1 / h;

d[0] = fi1(t);

a[a.size() - 1] = 0;

b[b.size() - 1] = 1;

d[d.size() - 1] = fi2(t);

answer.push\_back(solveMatrix(a, b, c, d));

t += tau;

}

return answer;

}

int main() {

ofstream output;

output.open("answer.txt");

vector<vector<double>> answer = solve();

double h = 0;

output << " ";

while (h < 1) {

output << h << ' ';

h += 0.1;

}

output << endl;

double t = 0;

for (int i = 0; i < answer.size(); ++i) {

output << t << ' ';

for (int j = 0; j < answer[i].size(); ++j) {

output << answer[i][j] << ' ';

}

output << endl;

t += 0.1;

}

output.close();

return 0;

}