**Ministry of education and science of Republic of Kazakhstan**

**Kazakh National University named after al-Farabi**



**Faculcy**: “Mechanics and Mathematis”

**Department**: “Mathematical and Computer Modelling”

**Report-8**

**Done by:** Kakibay A

**Checked by:** Zhandaulet E.

**Almaty**

**2020**

**2D Poisson equation**

F=0;

u[i][j] = (-(u0[i][j + 1] + u0[i][j - 1]) / pow(dy, 2) - (u0[i + 1][j] + u0[i - 1][j]) / pow(dx, 2)) / (-2\*((1/pow(dx,2)))+(1/pow(dy,2)));

#include <iostream>

#include <fstream>

#include <cmath>

using namespace std;

int main() {

const int n = 101;

int iter = 0;

double dx = 1.0 / (n - 1), dy = 1.0 / (n - 1), dif = 0.0, eps = pow(10, -5);

double u0[n][n], u[n][n];

double f[n][n];

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

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

u0[i][j] = 0.0;

u[i][j] = 0.0;

f[i][j] = 0.0;

}

}

do {

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

{

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

u0[0][j] = 0.0;

u[0][j] = 0.0;

u0[n-1][j] = dy\*j;

u[n-1][j] = dy\*j;

u0[i][n-1] = dx\*i;

u[i][n-1] = dx\*i;

u0[i][0] = -dy\*dx\*i + u0[i][1];

u[i][0] = -dy\*dx\*i + u[i][1];

//u0[i][1] = dx\*i\*dy\*dy + u0[i][0];

//u[i][1] = dx\*i\*dy\*dy + u0[i][0];

}

}

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

for (int j = 1; j < n - 1; j++)

u[i][j] = 0.25 \* (u0[i + 1][j] + u0[i - 1][j] + u0[i][j + 1] + u0[i][j - 1]);

//u[i][j] = (-(u0[i][j + 1] + u0[i][j - 1]) / pow(dy, 2) - (u0[i + 1][j] + u0[i - 1][j]) / pow(dx, 2)) / (-2\*((1/pow(dx,2)))+(1/pow(dy,2)));

dif = 0.0;

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

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

if (dif< abs(u[i][j] - u0[i][j])) {

dif = abs(u[i][j] - u0[i][j]);

}

}

}

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

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

u0[i][j] = u[i][j];

}

}

iter++;

} while (dif > eps);

ofstream fout("task8.dat");

fout << "VARIABLES = \"X\",\"Y\",\"u\"" << endl;

fout << "ZONE I=" << n << ",J=" << n << ",F=POINT" << endl;

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

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

fout << i \* dx << '\t' << j \* dy << '\t' << u[i][j] << endl;

}

}

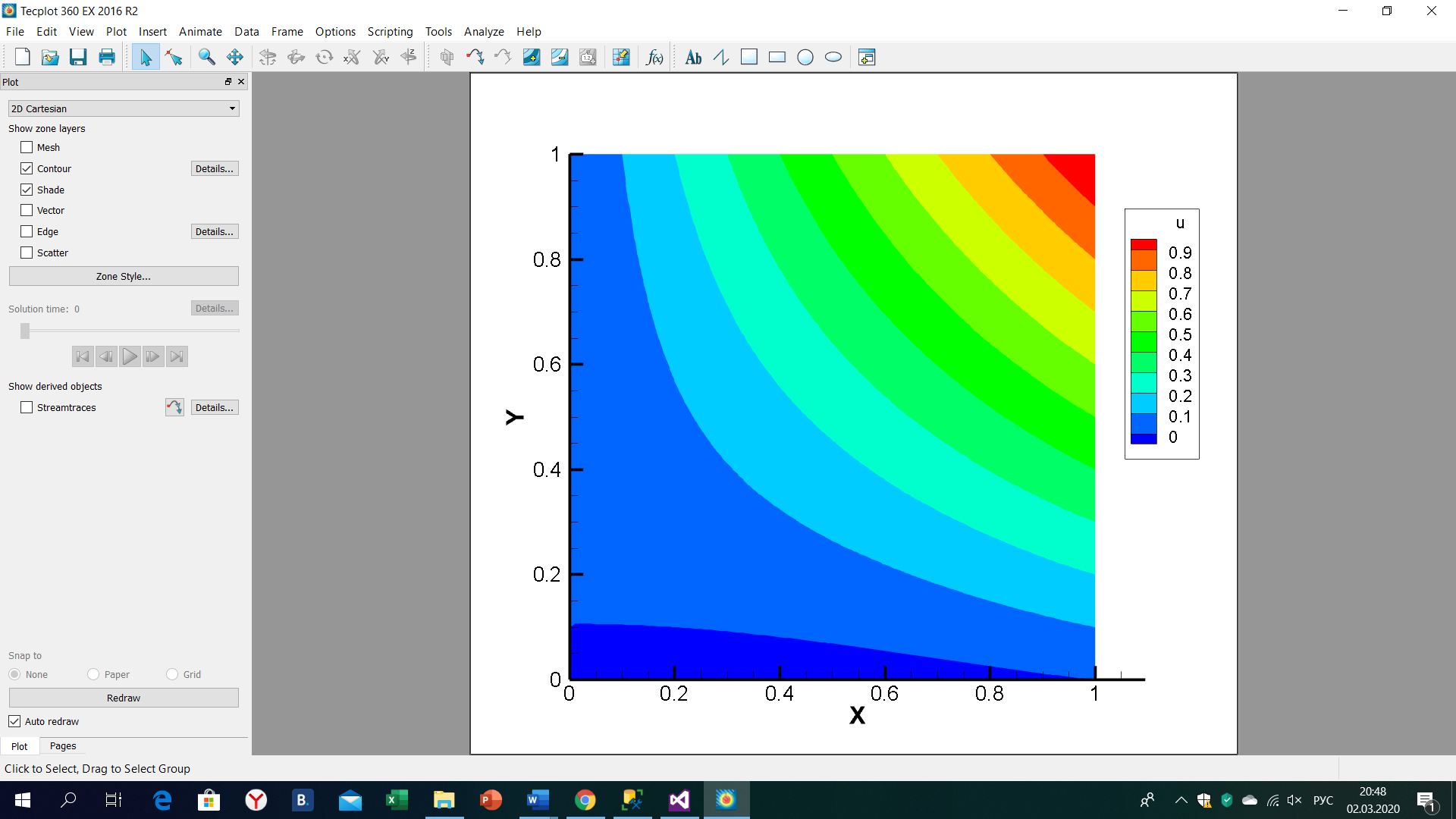
cout << "max difference: " << dif << endl;

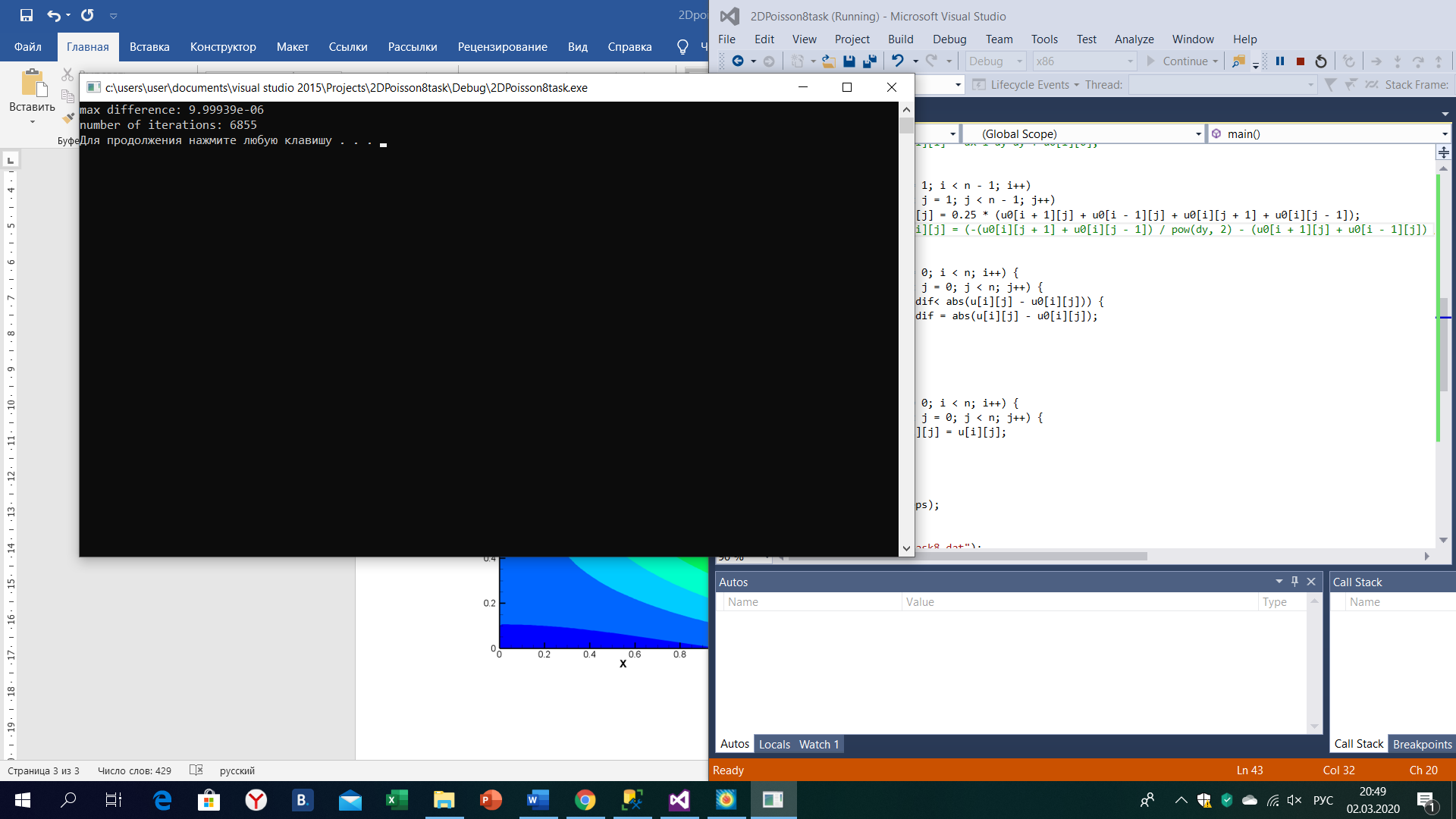
cout << "number of iterations: " << iter << endl;

system("pause");

return 0;

}





Conclusion:

We considered 2D Poisson equation, elliptic type, and we used Jacobi method of iteration. Also we can denote that function is 0, so we see that it becomes as Laplace equation. Here we used Jacobi method because in Poisson equation we donnot have time.We wrote 4 boundaries and one of them was given by Neumann boundary condition at y=0. Main task was to write boundaries rightly.