**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-7**

**Done by:** Kakibay A

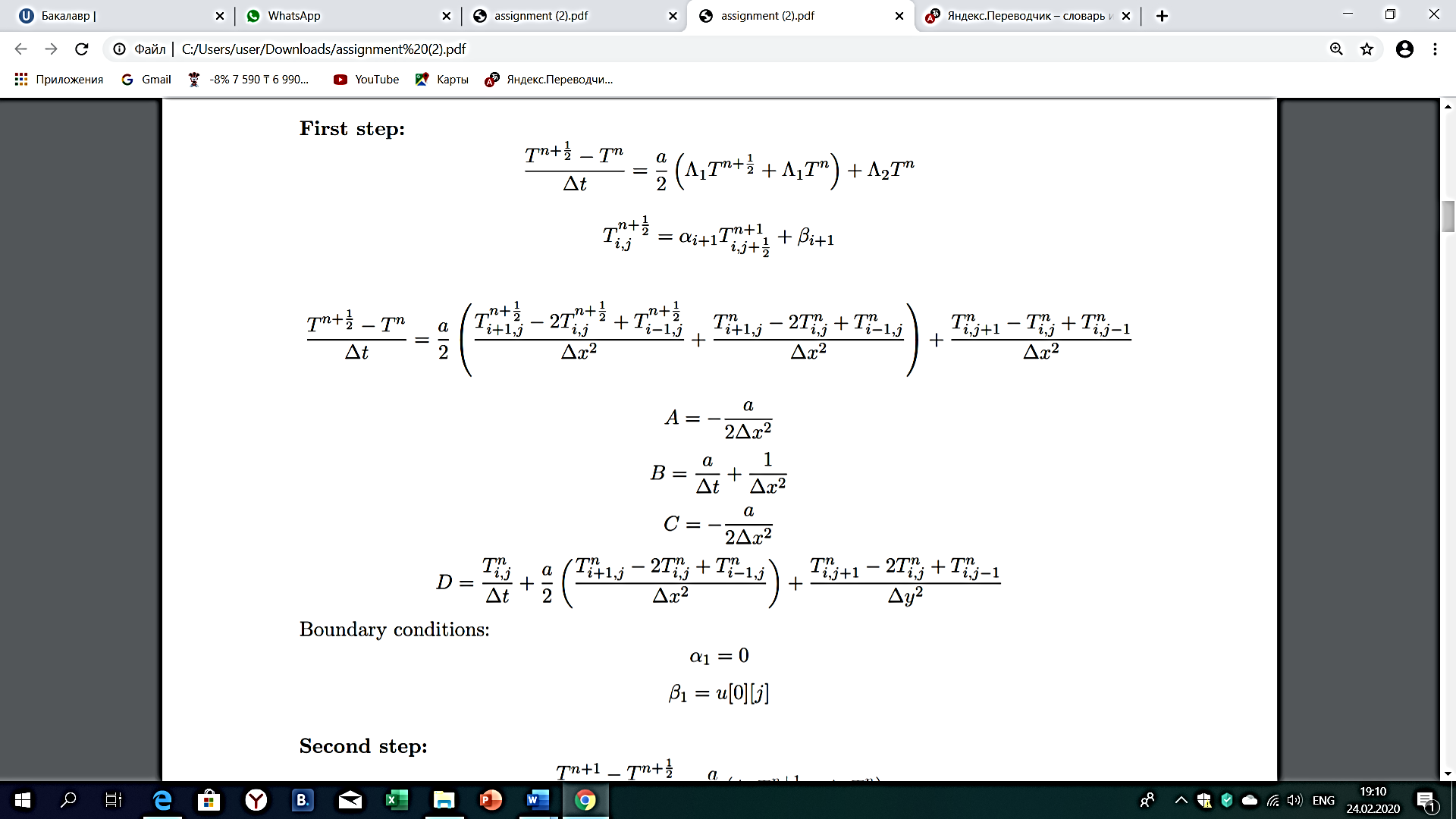
**Checked by:** Zhandaulet E.

**Almaty**

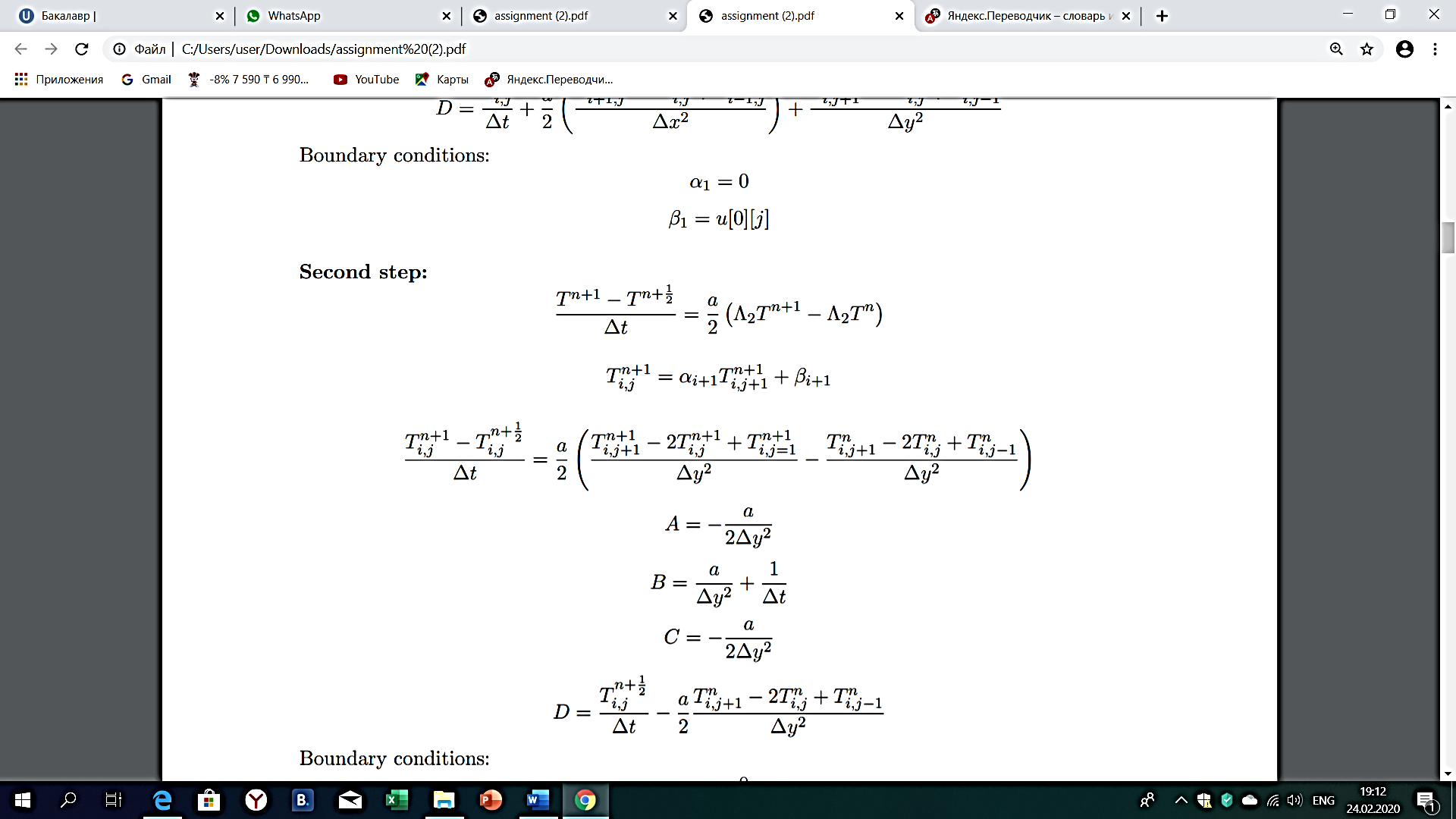
**2020**

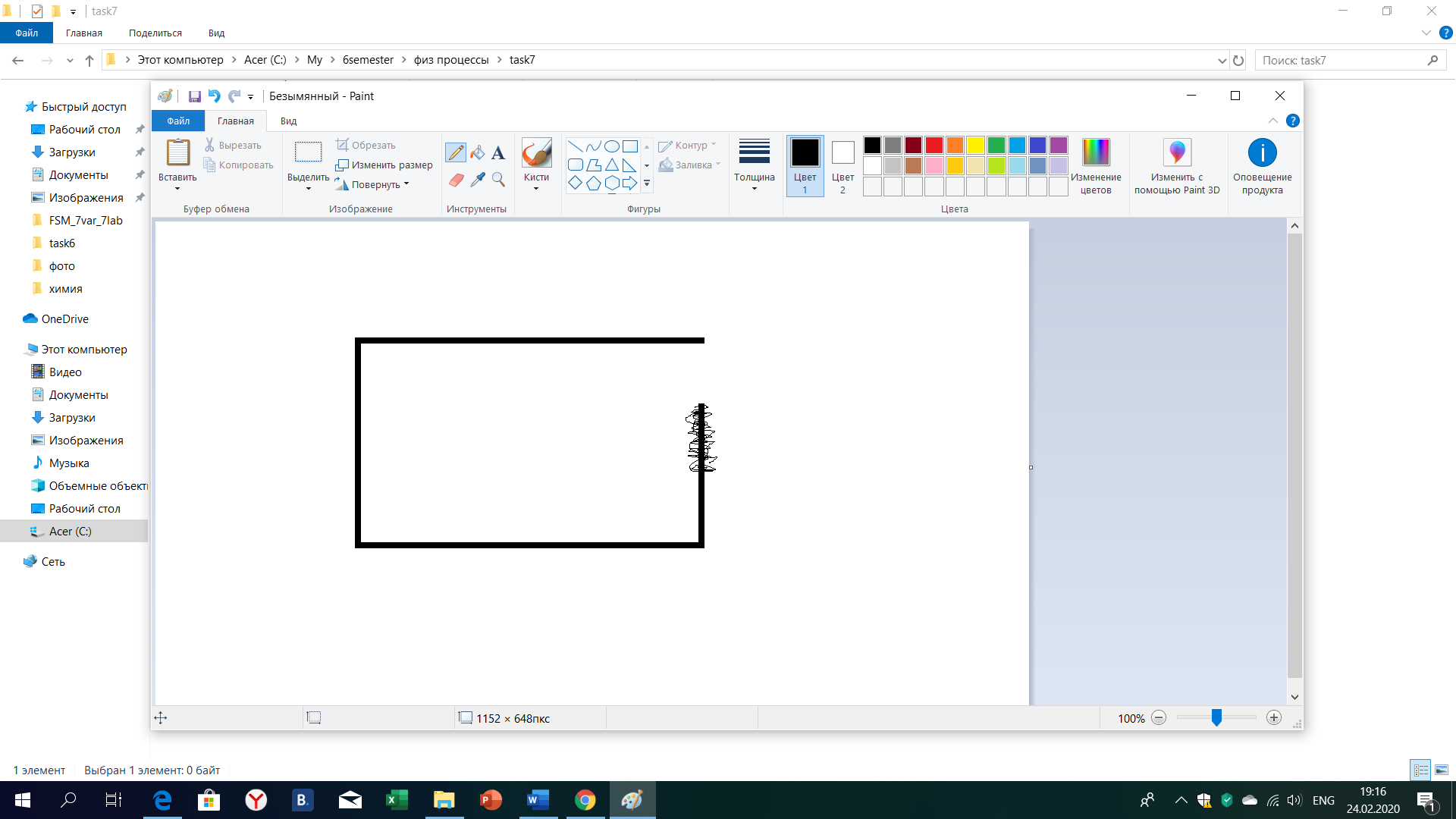
,

1)



2)





#include<iostream>

#include<fstream>

#include <cmath>

#include<math.h>

using namespace std;

int main() {

int const n = 31;

int iter = 0;

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

double dx = 1.0 / (n - 1), dy = 1.0 / (n - 1), dt = 0.25 \* dx \* dx, eps = pow(10, -7), maxdif = 0.0;

double A, B, C, D[n][n], a[n], b[n];

A = -1.0 / (2.0 \* dx \* dx);

B = 1.0 / dt + 1.0 / (dx \* dx);

C = -1.0 / (2.0 \* dx \* dx);

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

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

u0[i][j] = 0.0;

u[i][j] = 0.0;

}

}

for (int j = (n - 1) / 3; j <= (n - 1) \* 2 / 3; j++) {

u0[n - 1][j] = 1.0;

u[n - 1][j] = 1.0;

}

do {

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

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

D[i][j] = u0[i][j] / dt + (u0[i + 1][j] - 2.0 \* u0[i][j] + u0[i - 1][j]) / (2.0 \* dx \* dx) + (u0[i][j + 1] - 2.0 \* u0[i][j] + u0[i][j - 1]) / (dy \* dy);

}

}

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

a[1] = 0.0;

b[1] = 0.0;

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

a[i + 1] = (-A )/ (B + C \* a[i]);

b[i + 1] = (D[i][j] - C \* b[i]) / (B + C \* a[i]);

}

if (j >(n - 1) \* 2 / 3 && j < (n - 1)) {

u[n - 1][j] = b[n - 1] / (1 - a[n - 1]);

}

for (int i = n - 2; i >= 1; i--) {

u[i][j] = a[i + 1] \* u[i + 1][j] + b[i + 1];

}

}

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

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

D[i][j] = u[i][j] / dt - (u0[i][j + 1] - 2.0 \* u0[i][j] + u0[i][j - 1]) / (2.0 \* dy \* dy);

}

}

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

a[1] = 0.0;

b[1] = 0.0;

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

a[j + 1] = -A / (B + C \* a[j]);

b[j + 1] = (D[i][j] - C \* b[j]) / (B + C \* a[j]);

}

u[i][n - 1] = 0.0;

for (int j = n - 2; j >= 0; j--) {

u[i][j] = a[j + 1] \* u[i][j + 1] + b[j + 1];

}

}

maxdif = 0.0;

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

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

if (maxdif < fabs(u[i][j] - u0[i][j])) {

maxdif = fabs(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 (maxdif > eps);

ofstream fout("MDSH.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;

}

}

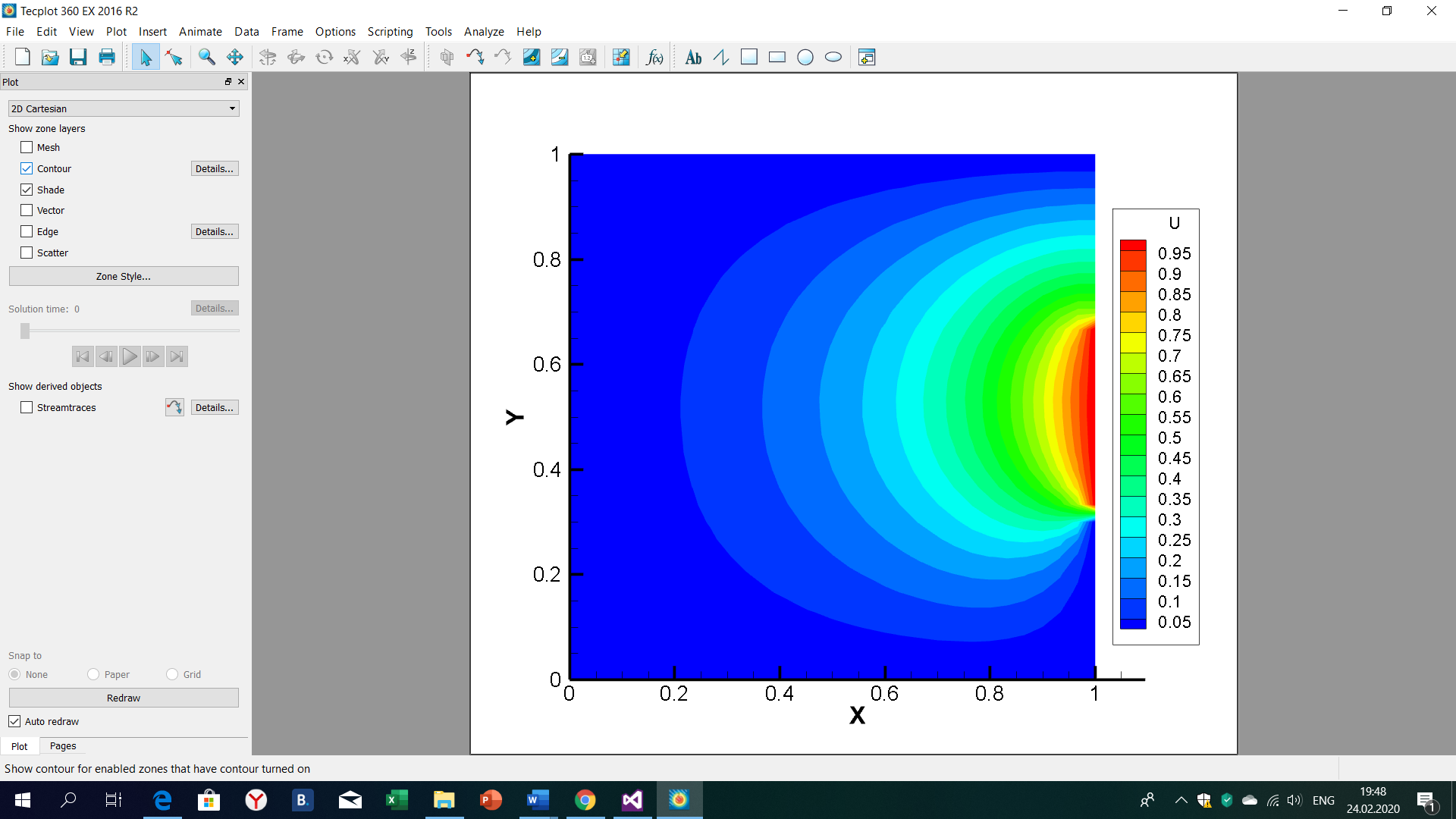
cout << maxdif << endl;

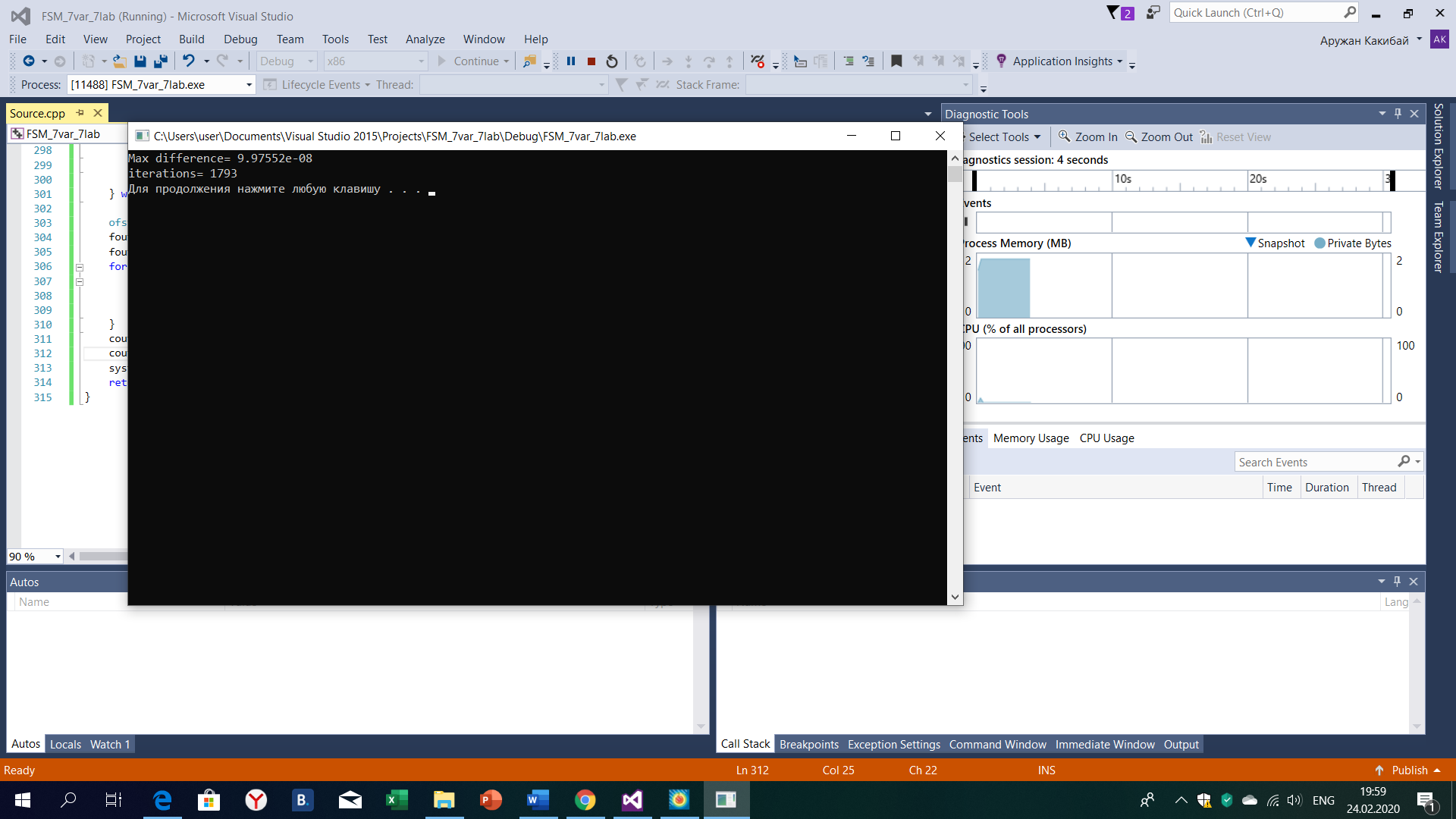
cout <<iter << endl;

system("pause");

return 0;

}





Conclusion:

Heat conduction equation is an equation of parabolic type. To get sufficient answer we need to give right initial and boundary conditions. We said that initial condition for this equation is 0 and boundary conditions were given as in image. In order to put heat we found points where it located and gave boundary condition by Dirichle T=1, so an important moment was to give right outlet of heat by Neumann as u[n - 1][j] = b[n - 1] / (1 - a[n - 1]).

Also we used fractional step method because this method is absolute stable.