**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**

**Done by:** Kakibay A

**Checked by:** Zhandaulet E.

**Almaty**

**2020**

**Sweep method for one-dimensional Poisson equation**

Let’s consider one-dimensional Poisson equation :

In numeric for is:

Here we see 3 unknowns, it means we work with implicate scheme. So write it as follows:

Here,

The sweep method is based on the assumption that the desired unknowns are related by a recurrent relationship:

Here, ,

U0

Un

By using boundaries we can find it helps to find next ones:

Problem:

,

x=0: , , ;

x=1: , , ;

Solution in C++ programming language:

#include <iostream>

#include<cmath>

#include<fstream>

using namespace std;

int main() {

int n = 100;

ofstream fout("task1.xls");

double dx = 0.01,maxdif = 0.0;

double A, B, C;

double ua[101], u[101],a[101],b[101],D[101];

A = 1.0 / (dx\*dx);

B = -2.0 / (dx\*dx);

C = 1.0 / (dx\*dx);

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

ua[i] = (-dx\*dx\*i\*i/4)- (cos(2\*dx\*i))/8 + (cos(2)-7)/8\*dx\*i + 9.0/8.0;

}

u[n] = 0.0;

u[0] = 1.0;

a[1] = 0.0;

b[1] = 1.0;

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

D[i] = -pow(sin(dx \* i), 2);

}

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

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

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

}

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

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

}

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

{

if (maxdif < abs(u[i] - ua[i])) {

maxdif = abs(u[i] - ua[i]);

}

}

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

fout << u[i] << "\t" << ua[i]<< "\t" << endl;

}

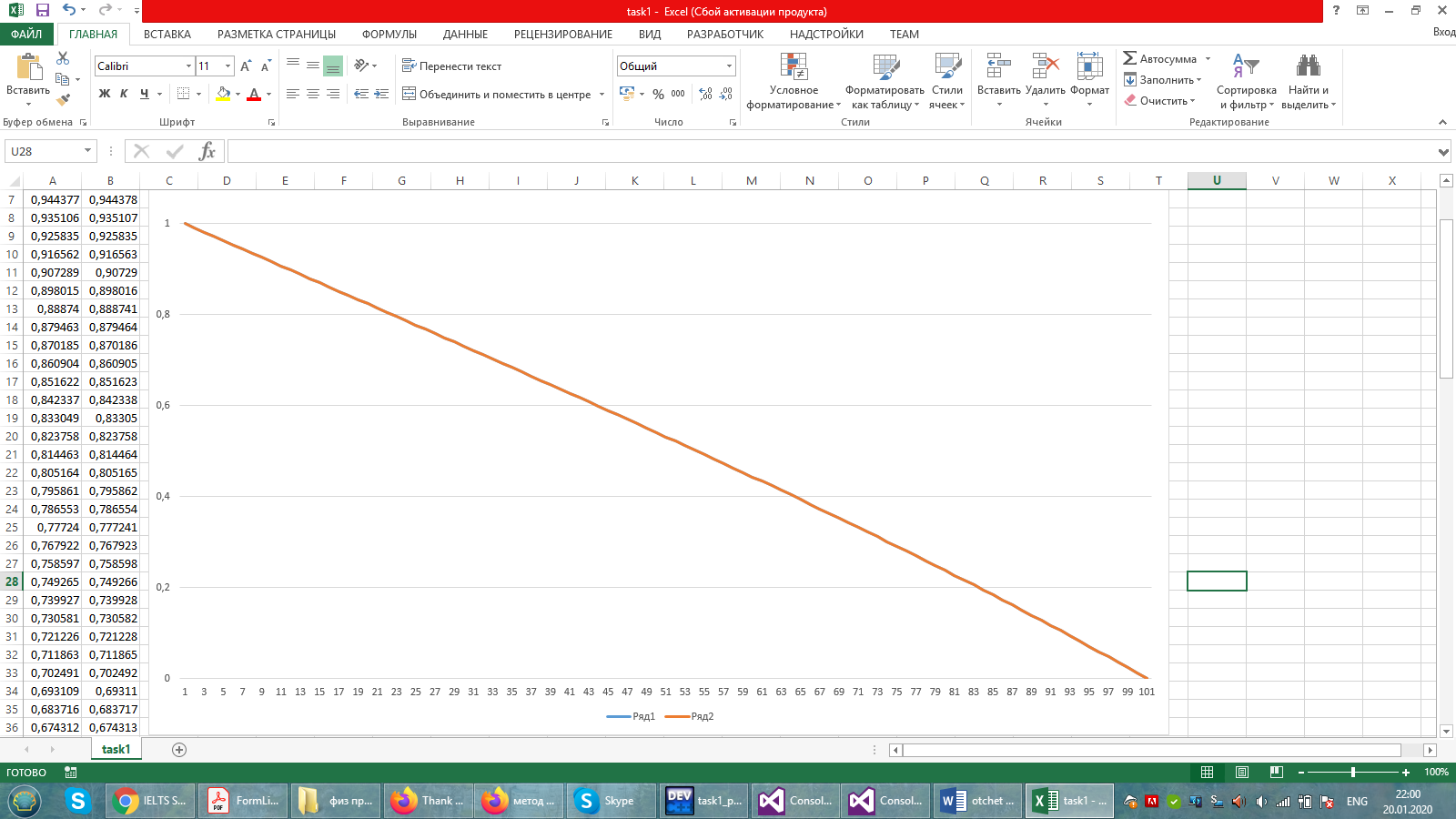
cout << "maxdif= " << maxdif << endl;

system("pause");

return 0;

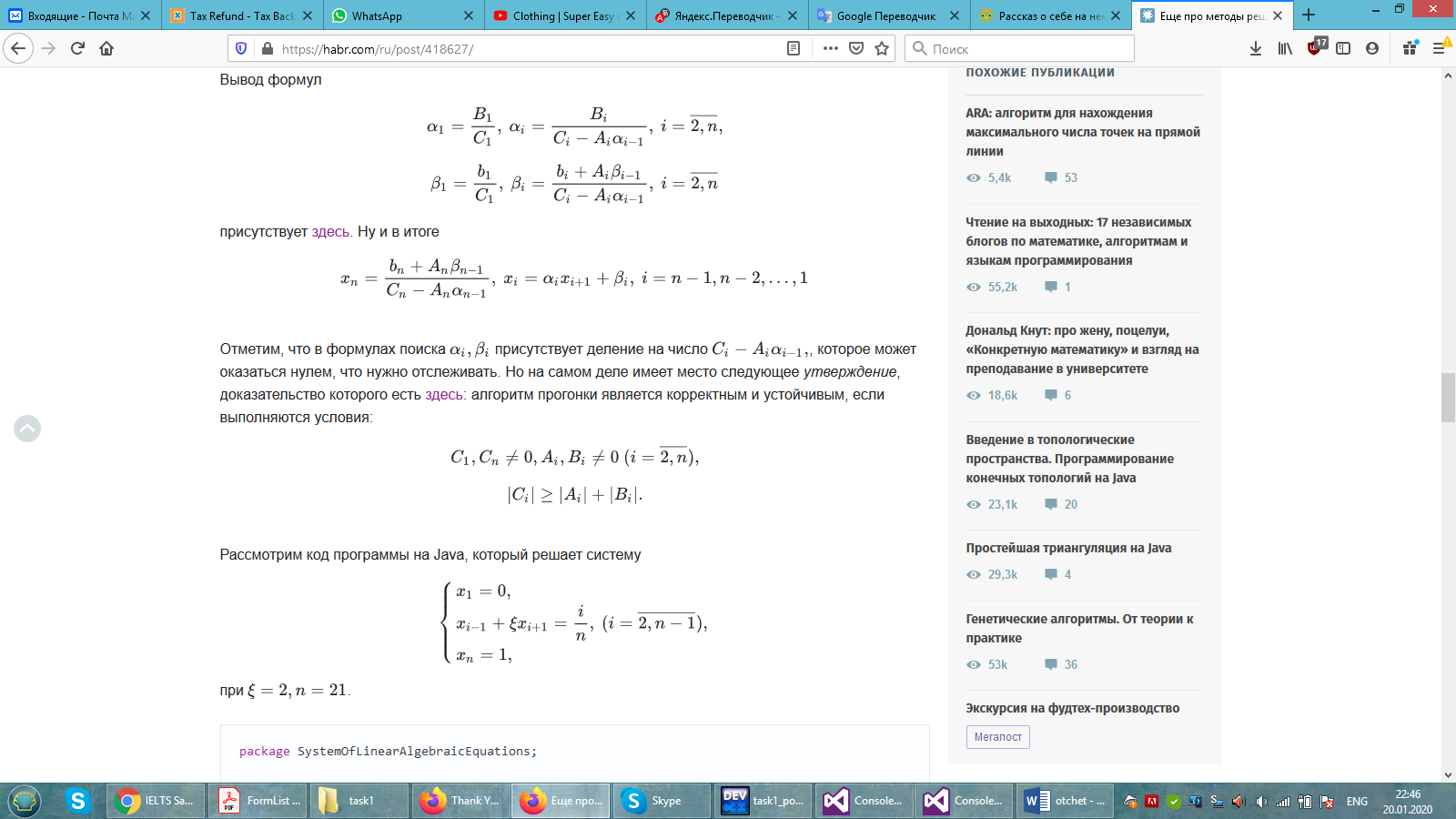
}

**Graphic :**



**Conclusion:**

Sweep method helps to sequentially eliminate unknowns. The advantage of this method is that the solution of two auxiliary systems is very simple. We considered one-dimensional Poisson equation with three unknowns, which means we have implicate scheme, that’s why we use Sweep method. The sweep method algorithm is correct and stable if the conditions are met:

.

If the original system is not degenerate, then the coefficients Ci and bi can not go to zero at the same time. This ensures that the process is correct.

For comparison rightness of numeric solution we solved it in analytic form. In this case considered sinusoid function. I have showed before steps of solving, here is important to know boundaries. At the end their graphics matched but with small maximum difference. It means we got right solution of problem.