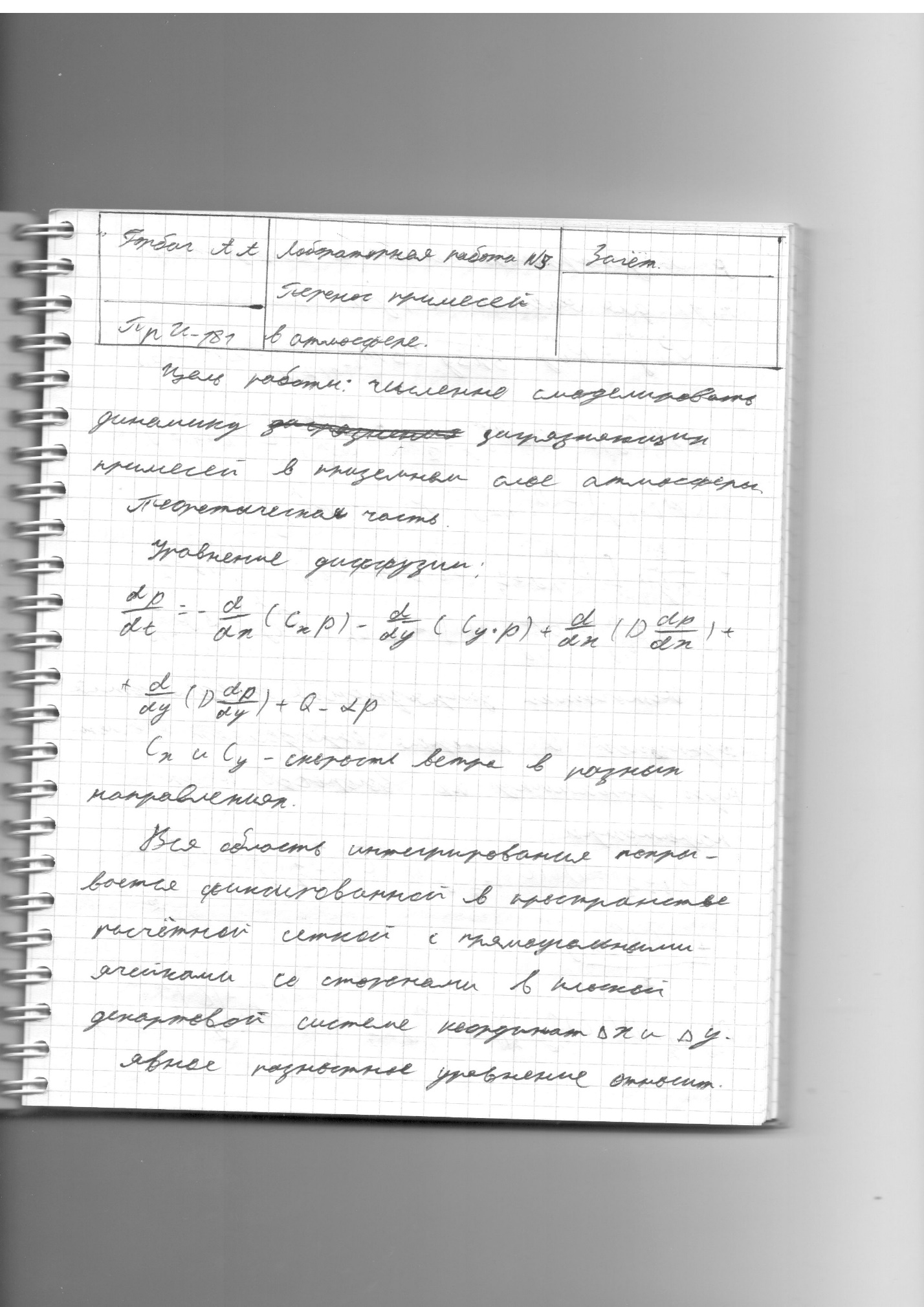
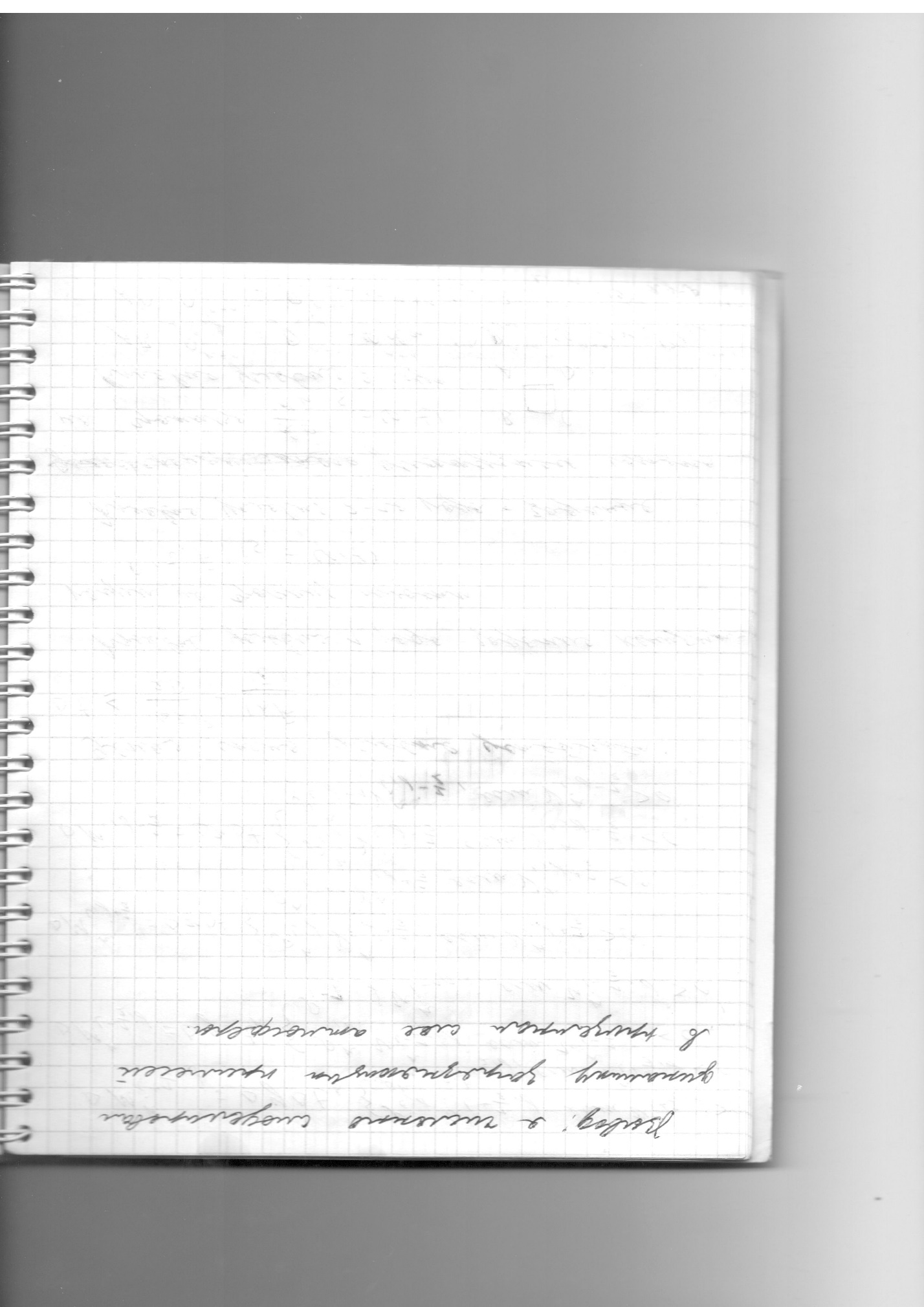
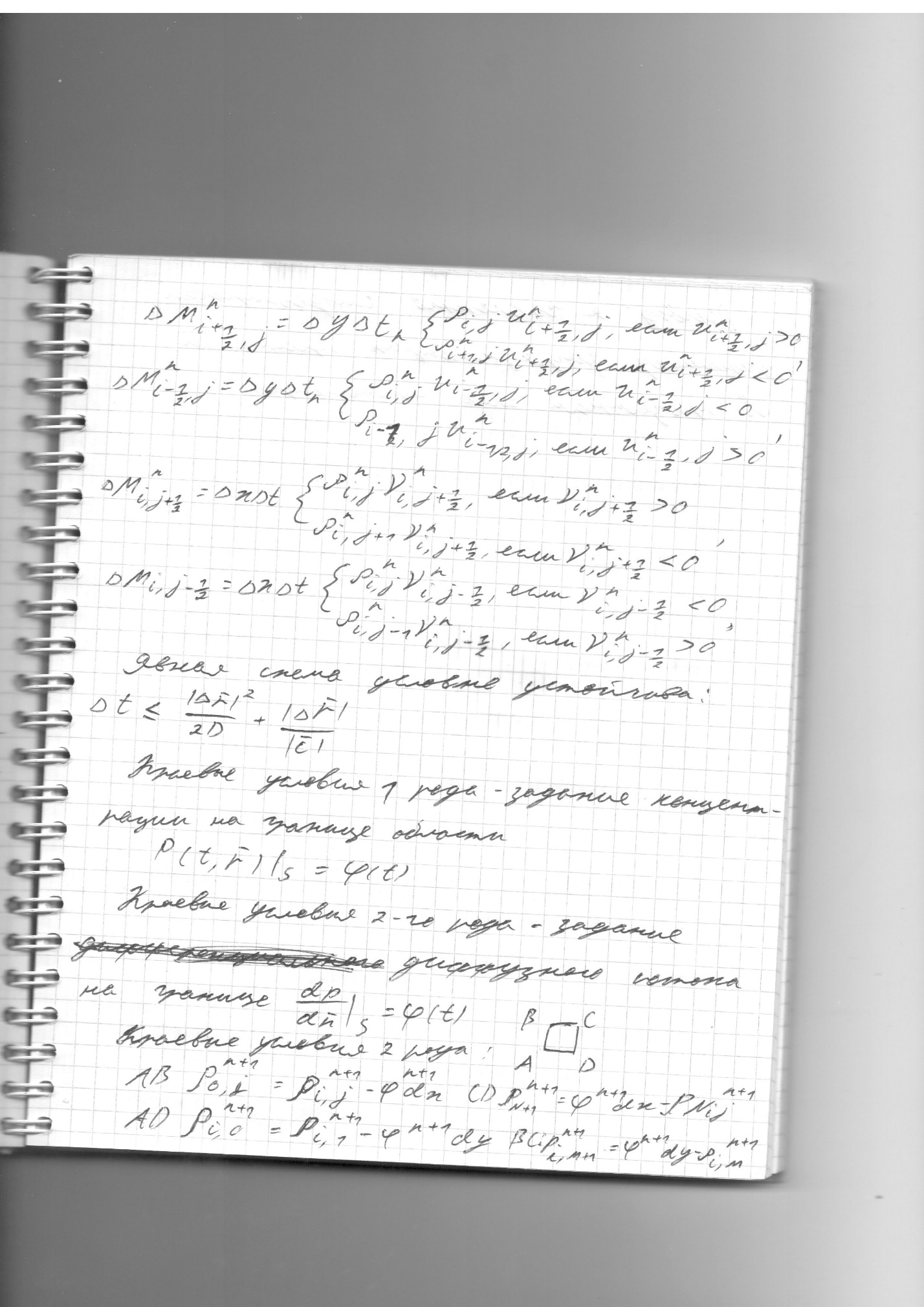
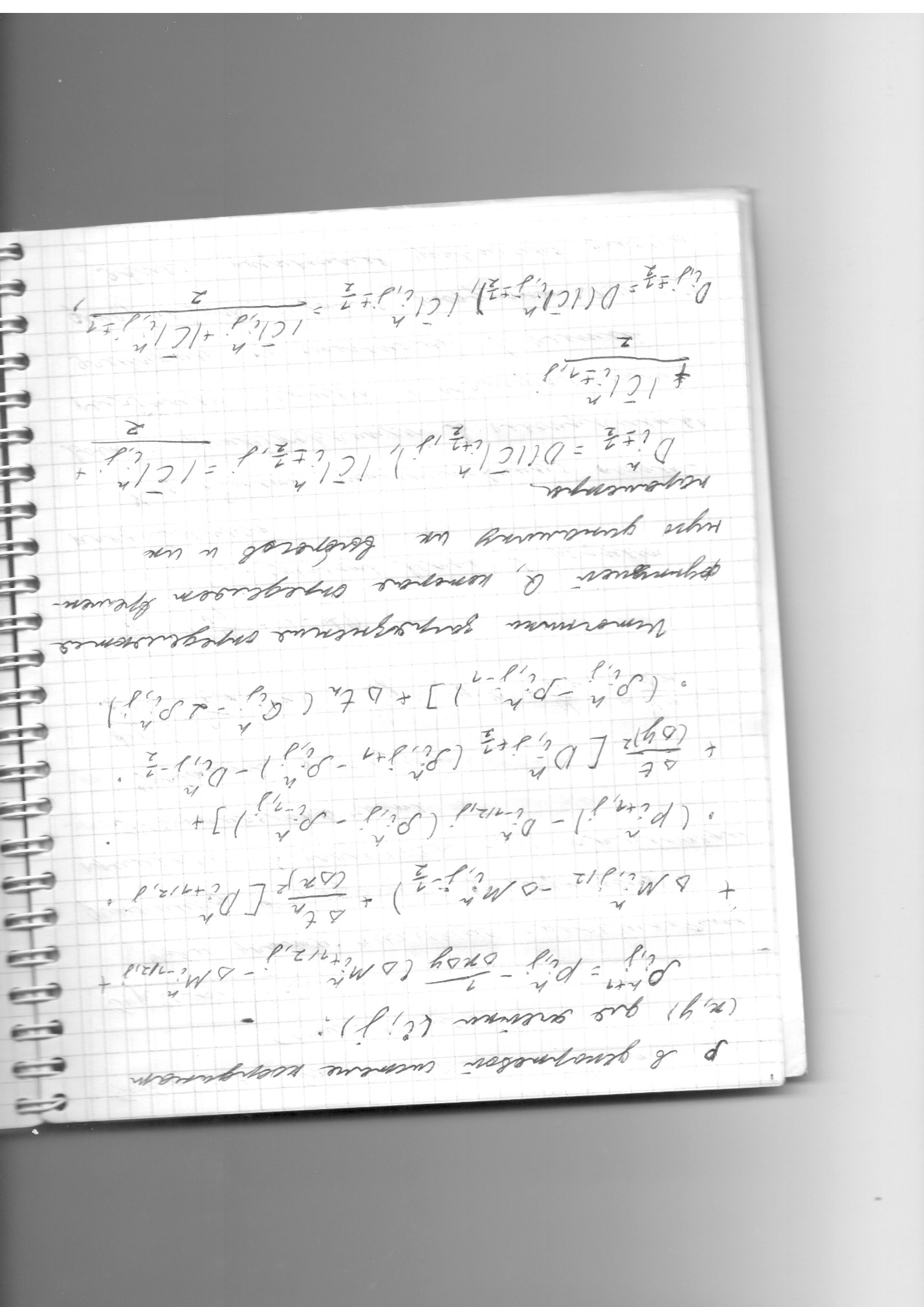
**Лабораторная работа №5.**

**Практическая часть.** 



**Практическая часть.**

Код программы:

#include <iostream>

#include <clocale>

#include <string>

#include <fstream>

#include <Windows.h>

#include <cmath>

#include <iomanip>

#define PI 3.1415926535

using namespace std;

//Функция записи в файл для визуализации

void In\_VTK\_file(double \*\*P, int N, double dx, string filename)

{

ofstream file(filename);

//Шапка VTK файла

file << "# vtk DataFile Version 5.0" << endl;

file << "Example 3D regular grid VTK file." << endl;

file << "ASCII" << endl;

file << "DATASET UNSTRUCTURED\_GRID" << endl;

file << "POINTS " << (N + 1) \* (N + 1) << " float" << endl;

//Запись точек в файл

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

{

float x = i \* dx;

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

{

float y = j \* dx;

file << x << " " << y << " 0" << endl;

}

}

//Шапка для данных о точках

file << "POINT\_DATA " << (N + 1) \* (N + 1) << endl;

file << "SCALARS P float" << endl;

file << "LOOKUP\_TABLE default" << endl;

//Запись значений в каждой точке

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

{

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

{

file << fixed << setprecision(4) << P[i][j] << endl;

}

}

file.close();

}

//Источник выброса

double Q(double x, double y)

{

if (x == 100 && y == 50)

return 2000; //Мощность выброса

else

return 0;

}

int main()

{

setlocale(LC\_ALL, "Rus");

double dx = 10; //Шаг по x

double dt; //Шаг по времени

double t = 0; //Начальное время

double t\_end = 500; //Конечное время

double x\_last = 3000;

double C = 10; //Ветер

double D = 100; //Диффузия

double alpha = 45.0 \* PI / 180.0; //Угол направления ветра

double M1, M2, M3, M4;

M1 = M2 = M3 = M4 = 0;

double K1, K2, K3, K4, K5;

double k = 0.05; //Коэффициент Куранта

double Cx = C \* cos(alpha);

double Cy = C \* sin(alpha);

int i, j;

int N = x\_last / dx;

//Плотность примеси в текущий момент времени

double\*\* P = new double\* [N + 2.0];

for (int i = 0; i < N + 2.0; i++)

{

P[i] = new double[N + 2.0];

}

//Плотность примеси в текущий момент времени

double\*\* P\_next = new double\* [N + 2.0];

for (int i = 0; i < N + 2.0; i++)

{

P\_next[i] = new double[N + 2.0];

}

//Начальные условия

for (i = 0; i <= N + 1; i++)

{

for (j = 0; j <= N + 1; j++)

{

P[i][j] = 0;

P\_next[i][j] = 0;

}

}

int n = 0;

//Условие устойчивости

dt = k \* min((dx \* dx) / (2 \* D) + dx / abs(Cx), (dx \* dx) / (2 \* D) + dx / abs(Cy));

t += dt;

//Основные расчеты

do

{

for (i = 1; i <= N; i++)

{

for (j = 1; j <= N; j++)

{

if (Cx > 0)

{

M1 = P[i][j] \* Cx \* dx \* dt;

M2 = P[i - 1][j] \* Cx \* dx \* dt;

}

else if (Cx < 0)

{

M1 = P[i + 1][j] \* Cx \* dx \* dt;

M2 = P[i][j] \* Cx \* dx \* dt;

}

if (Cy > 0)

{

M3 = P[i][j] \* Cy \* dx \* dt;

M4 = P[i][j - 1] \* Cy \* dx \* dt;

}

else if (Cy < 0)

{

M3 = P[i][j + 1] \* Cy \* dx \* dt;

M4 = P[i][j] \* Cy \* dx \* dt;

}

//Основная формула

//Разбита на 5 частей, чтобы удобней было с ней работать

K1 = P[i][j];

K2 = 1.0 / (dx \* dx) \* (M1 - M2 + M3 - M4);

K3 = (dt / (dx \* dx)) \* (D \* (P[i + 1][j] - P[i][j]) - D \* (P[i][j] - P[i - 1][j]));

K4 = (dt / (dx \* dx)) \* (D \* (P[i][j + 1] - P[i][j]) - D \* (P[i][j] - P[i][j - 1]));

K5 = Q(i \* dx, j \* dx) \* dt;

P\_next[i][j] = K1 - K2 + K3 + K4 + K5;

}

}

//Граничные условия

for (j = 0; j <= N + 1; j++)

{

P\_next[0][j] = P\_next[1][j];

P\_next[N + 1][j] = P\_next[N][j];

}

for (i = 0; i <= N + 1; i++)

{

P\_next[i][0] = P\_next[i][1];

P\_next[i][N + 1] = P\_next[i][N];

}

//Меняем массивы местами

for (i = 0; i <= N + 1; i++)

{

for (j = 0; j <= N + 1; j++)

{

P[i][j] = P\_next[i][j];

}

}

//Запись в файл

if (n % (int)(0.5 \* N) == 0)

{

auto s = to\_string(n);

//Строка названия файла с данными

string filename = "test\\p" + s + ".vtk";

In\_VTK\_file(P, N, dx, filename);

}

n++;

t += dt;

}

while (t <= t\_end);

//Очистка памяти

for (int i = 0; i < N + 2.0; i++)

{

delete[] P[i];

}

for (int i = 0; i < N + 2.0; i++)

{

delete[] P\_next[i];

}

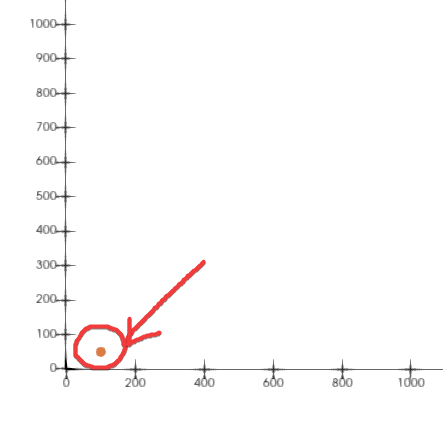
system("pause");

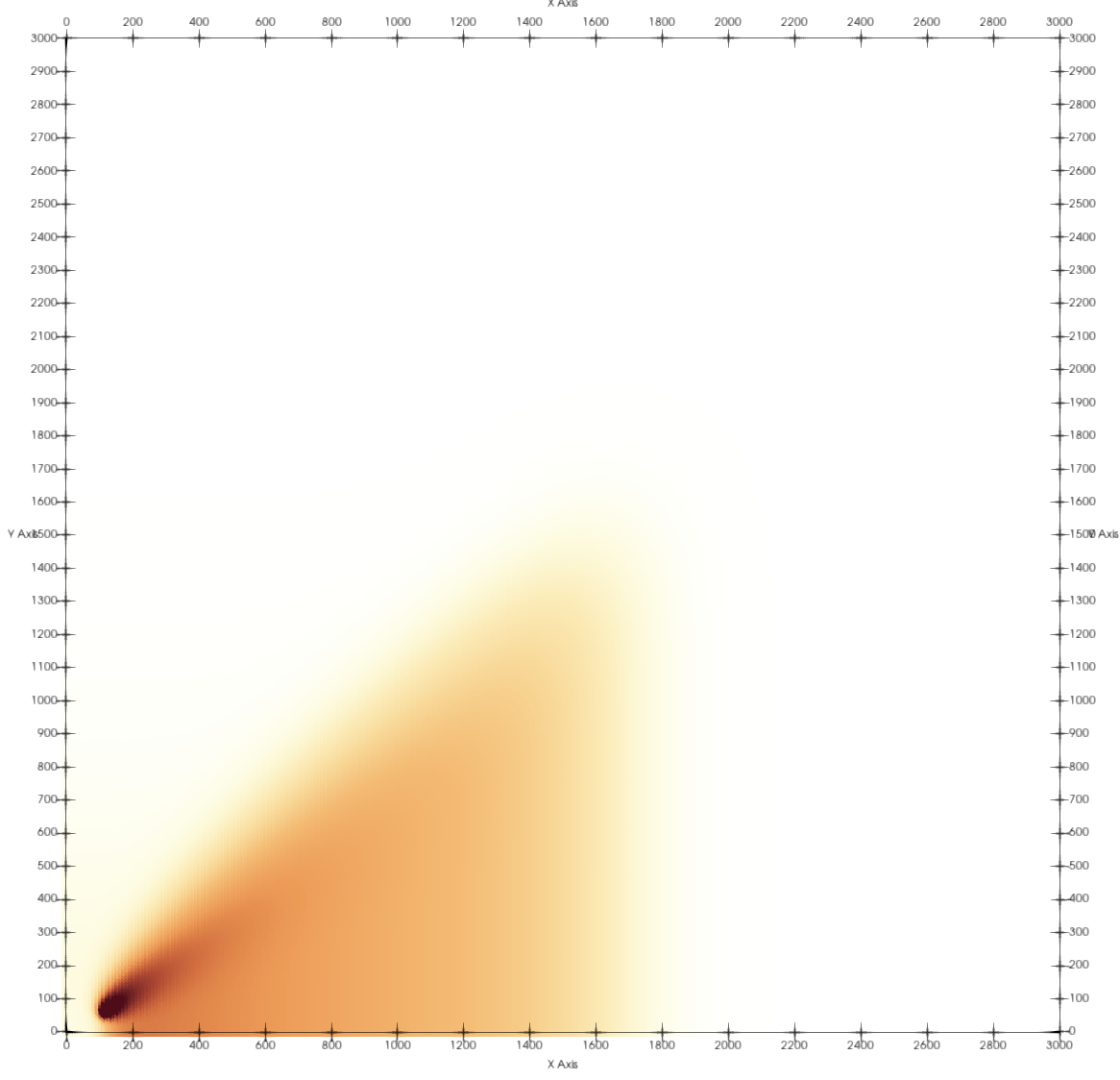
return 0;

}

**Графики.**

Была создана анимация переноса примесей в атмосфере.

Источник загрязнения в нулевой момент времени: 

Промежуточный момен времени:

Конечный момент времени (500).

