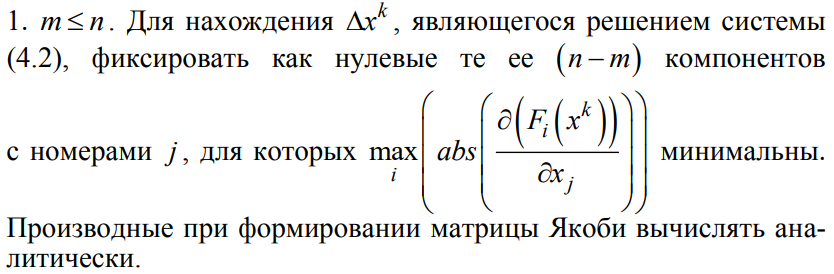
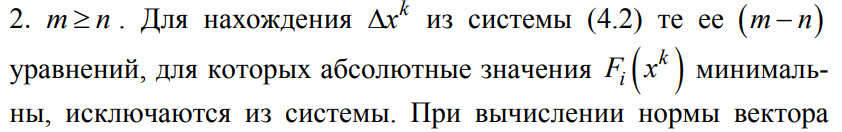
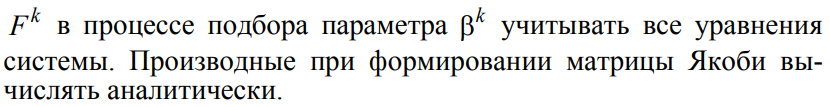
|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
| Практическое задание № 4 | | |
| по дисциплине «Численные методы» | | |
| **решение систем нелинейных уравнений методом ньютона** | | |
|  | | |
|  | Вариант 1,2,6 | овчинников иван, лойченко данила |
| Группа ПМ - 12 |  |
|  |  |
| Преподаватели | Задорожный александр геннадьевич |
|  | Леонович Дарьяна Александровна |
|  |  |
|  |  |
|  |  |
| Новосибирск, 2023 | | |

1. **Задание**

Разработать программу решения системы нелинейных уравнений (СНУ) методом Ньютона. Провести исследования метода для нескольких систем размерности от 2 до 10.









1. **Исследования**

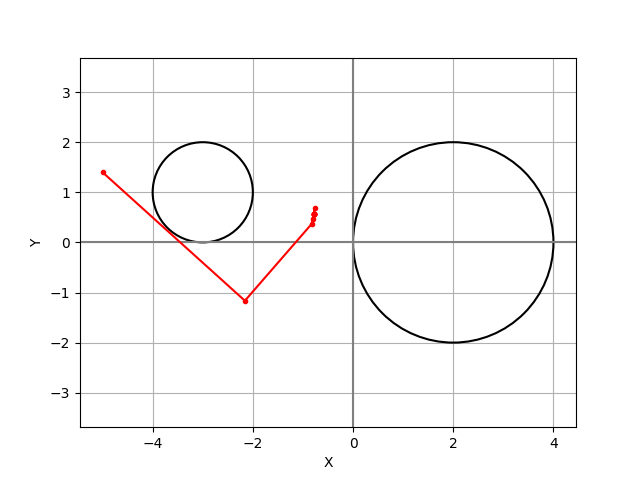
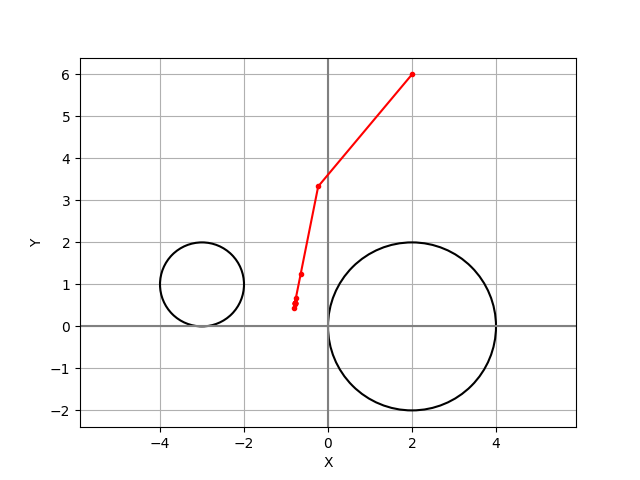
Максимальное количество итераций

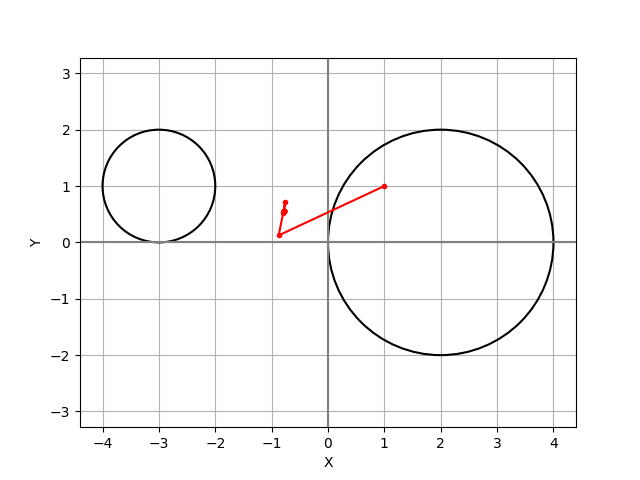
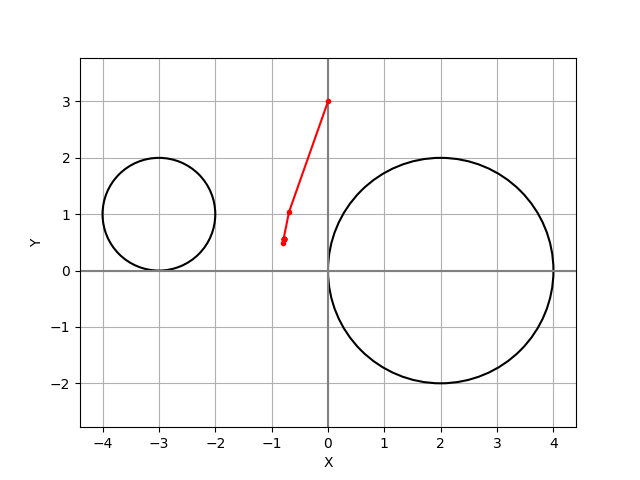
Максимальная невязка

Шаг численной производной

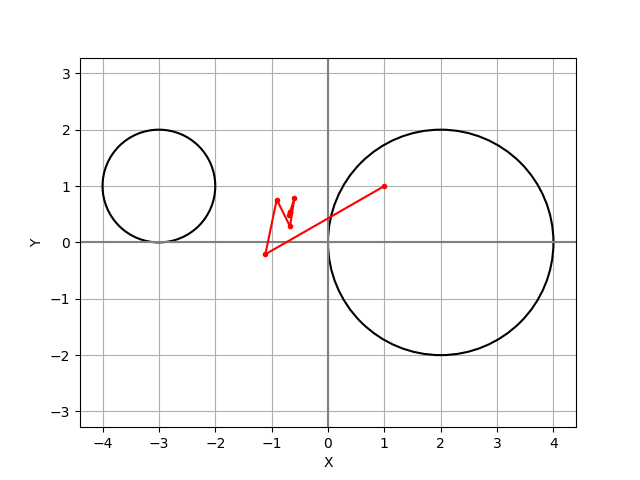
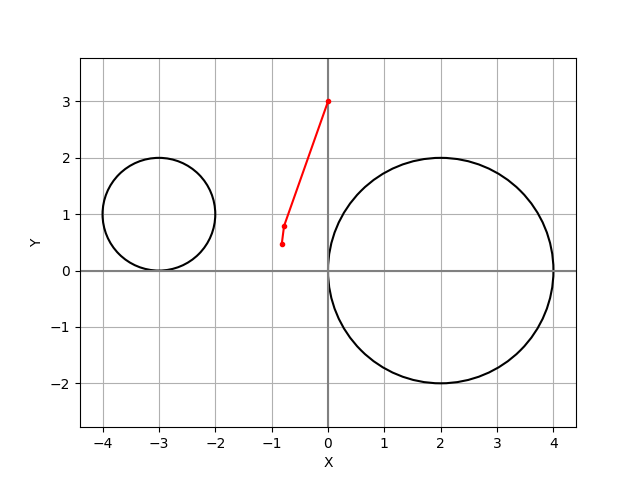
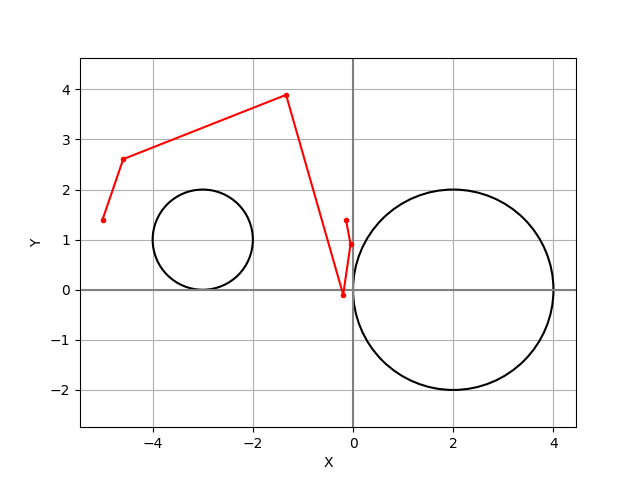
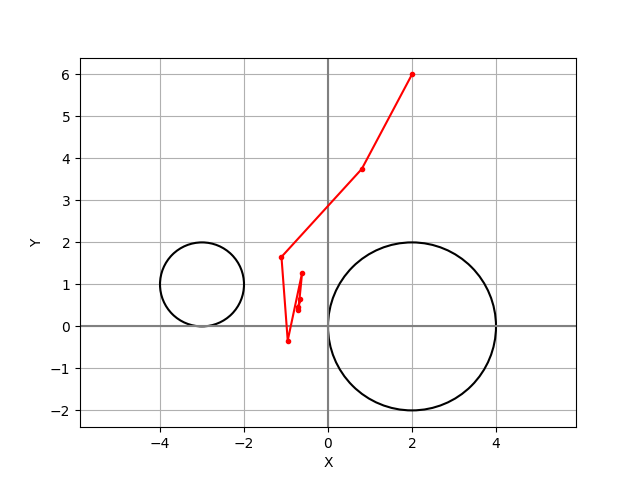
На непересекающихся окружностях

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x0** | **Аналитические производные** | | | |
| **k** | **||FK||** | **b** | **результат** |
| (2, 6)  Не лежит на осях симметрии | 12 | 5.779238115466975e+00 | 1.776356839400250e-15 | -0.788462  0.557692 |
| (-5, 1.4)  Лежит на оси, соединяющей центры окружностей | 14 | 5.779238115466979e+00 | 1.776356839400250e-15 | -0.788462  0.557692 |
| (0, 3)  Лежит на оси, перпендикулярной оси, соединяющей центры окружностей, и пересекающей ее на равных расстояниях от центров окружностей | 14 | 5.779238115466978e+00 | 1.776356839400250e-15 | -0.788462  0.557692 |
| (1, 1)  Лежит внутри одной из окружностей | 16 | 5.779238115466976e+00 | 1.776356839400250e-15 | -0.788462  0.557692 |
| **Численные производные** | | | | |
| (2, 6)  Не лежит на осях симметрии | 5 | 5.761010832735749e+00 | 1.776356839400250e-15 | -0.706480  0.472157 |
| (-5, 1.4)  Лежит на оси, соединяющей центры окружностей | 6 | 1.066755774301326e+01 | 1.776356839400250e-15 | -0.140540  1.391611 |
| (0, 3)  Лежит на оси, перпендикулярной оси, соединяющей центры окружностей, и пересекающей ее на равных расстояниях от центров окружностей | 3 | 5.802782110906933e+00 | 1.776356839400250e-15 | -0.819421  0.478578 |
| (1, 1)  Лежит внутри одной из окружностей | 7 | 5.759779623499789e+00 | 1.776356839400250e-15 | -0.680016  0.538329 |



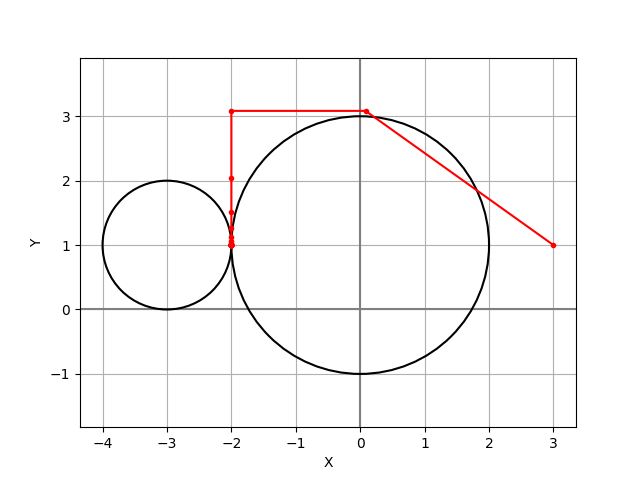
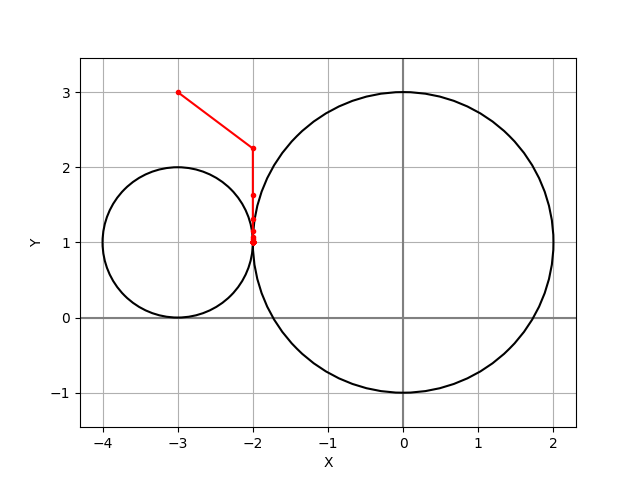


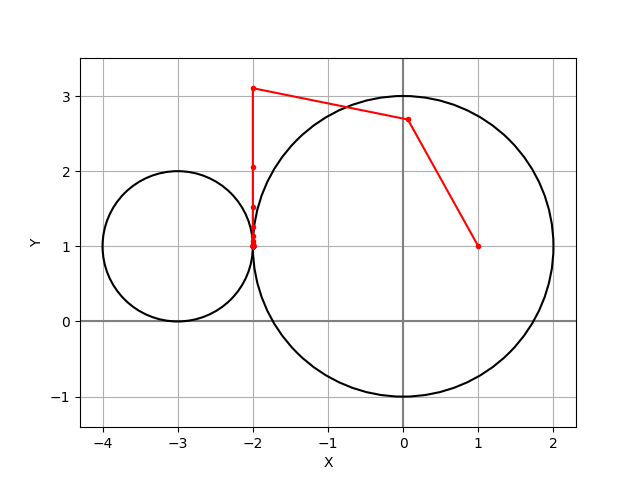
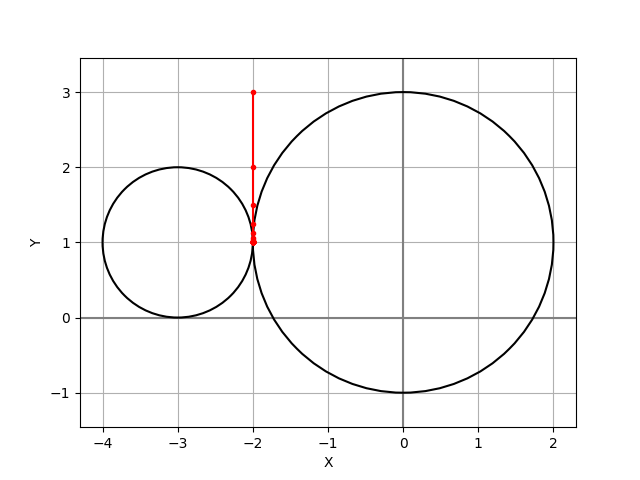
Численные производные



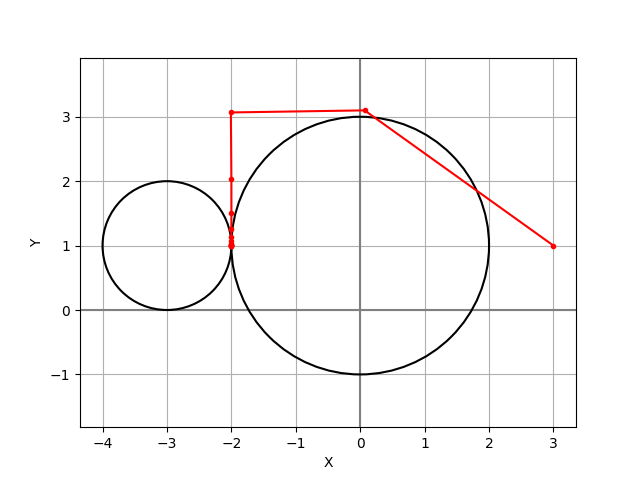
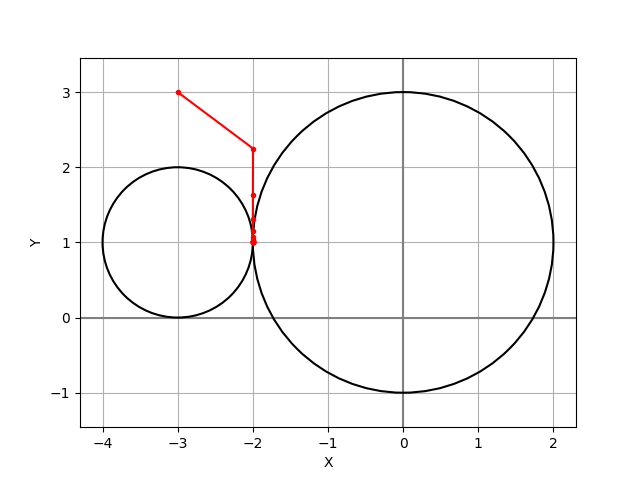
Окружности пересекаются в одной точке

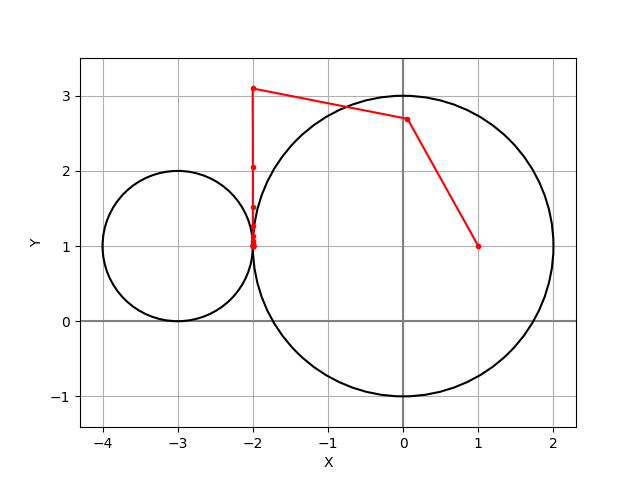
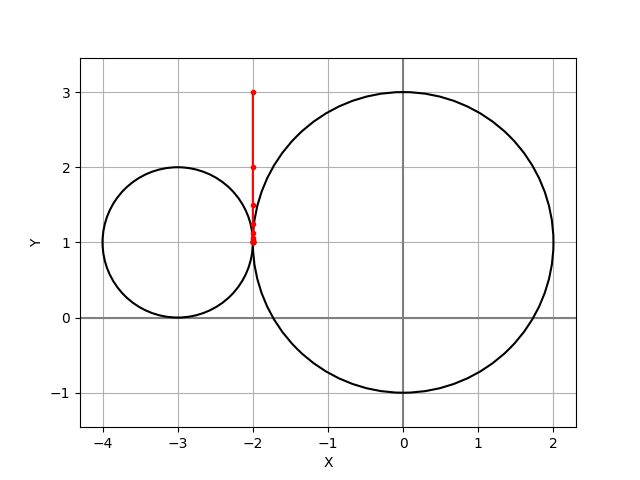
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x0** | **Аналитические производные** | | | |
| **k** | **Погрешность** | **b** | **результат** |
| (-3, 3)  Не лежит на осях симметрии | 26 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -2.000000  1.000000 |
| (3, 1)  Лежит на оси, соединяющей центры окружностей | 27 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -2.000000  1.000000 |
| (-2, 3)  Лежит на оси, перпендикулярной оси, соединяющей центры окружностей, и пересекающей ее на равных расстояниях от центров окружностей | 26 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -2.000000  1.000000 |
| (1, 1)  Лежит внутри одной из окружностей | 28 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -2.000000  1.000000 |
| **Численные производные** | | | | |
| (-3, 3)  Не лежит на осях симметрии | 16 | 0.00E+00  -3.40E-04 | 4.768371582031250e-07 | -2.000000  1.000340 |
| (3, 1)  Лежит на оси, соединяющей центры окружностей | 15 | 0.00E+00  -3.40E-04 | 1.000000000000000e+00 | -2.000000  1.000370 |
| (-2, 3)  Лежит на оси, перпендикулярной оси, соединяющей центры окружностей, и пересекающей ее на равных расстояниях от центров окружностей | 15 | 0.00E+00  -2.65E-04 | 1.000000000000000e+00 | -2.000000  1.000265 |
| (1, 1)  Лежит внутри одной из окружностей | 14 | 0.00E+00  -5.08E-04 | 1.000000000000000e+00 | -2.000000  1.000508 |
| **Исследование на шаг** | | | | |
| **x0** | **h** | **Итерации** | **Невязка** | |
| (1;1) | 1 | 645 | 3.534248876500839e-06 | |
| 0.9 | 582 | 3.531692074038355e-06 | |
| … | | | | |
| (1;1) | 1e-2 | 25 | 2.985594357795353e-06 | |
| 1e-3 | 23 | 1.477330402670230e-06 | |
| 1e-4 | 24 | 2.821362637630145e-06 | |
| 1e-5 | 26 | 2.092464867805808e-06 | |
| 1e-6 | 29 | 1.108885610626435e-06 | |
| 1e-7 | - | - | |
| (1;1) | 1e-8 | 14 | 1.463481147154726e-06 | |
| 1e-9 | 14 | 1.463294816985123e-06 | |
| 1e-10 | 14 | 1.466873779113013e-06 | |
| 1e-11 | 14 | 1.475365776436473e-06 | |
| 1e-12 | 14 | 1.652157419758577e-06 | |
| 1e-13 | 15 | 2.036976827302254e-06 | |
| 1e-14 | 36 | 3.412833448027155e-06 | |
| 1e-15 | - | - | |





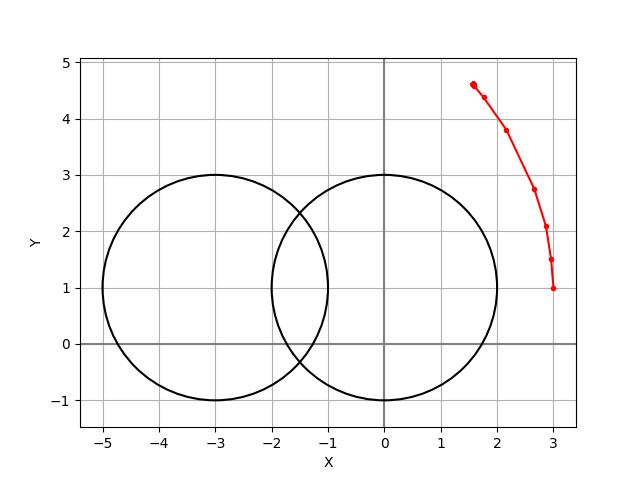
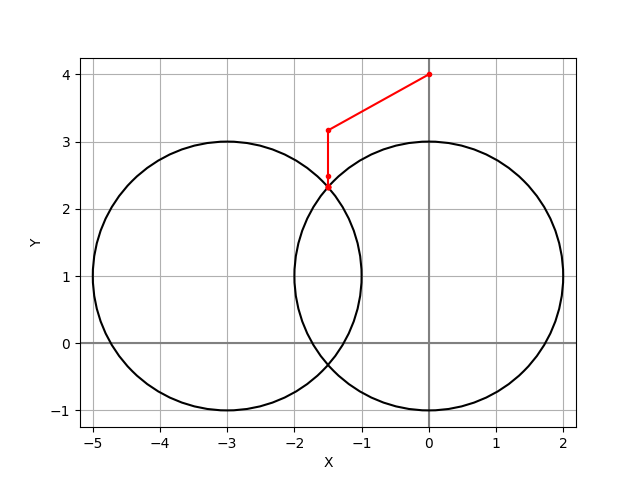
Численные производные

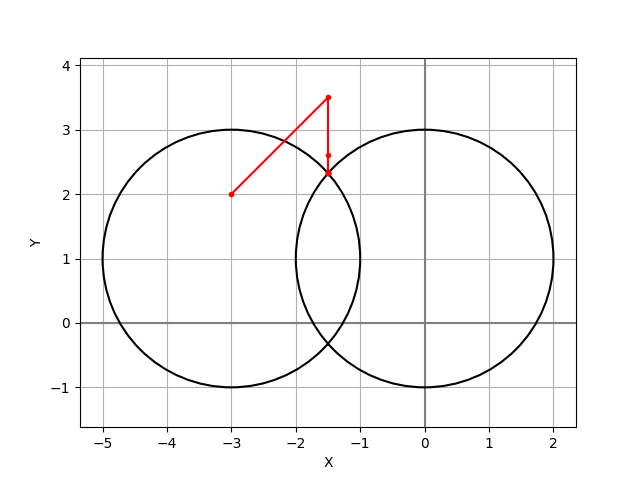
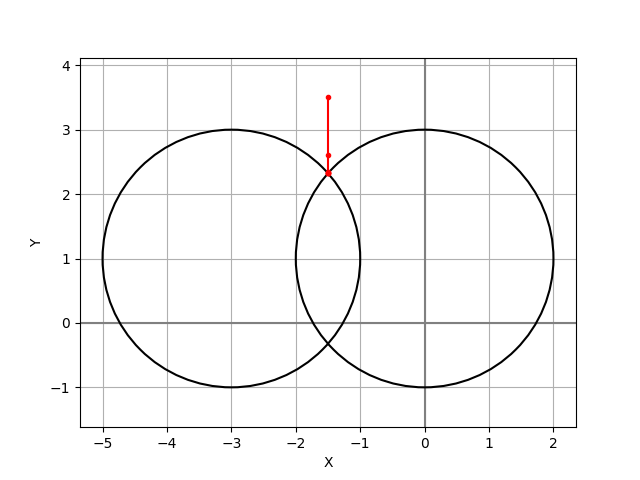




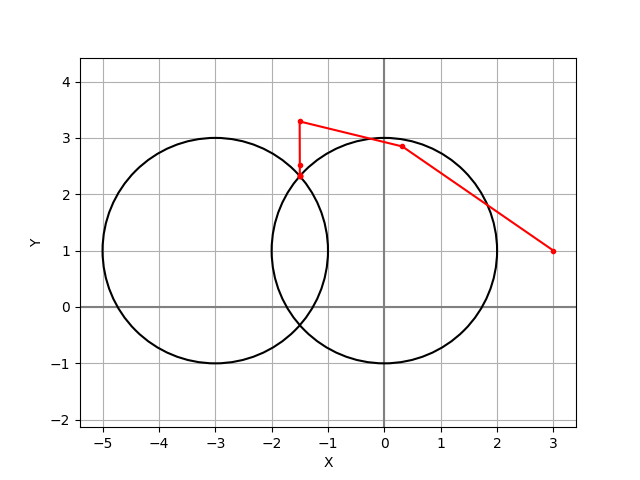
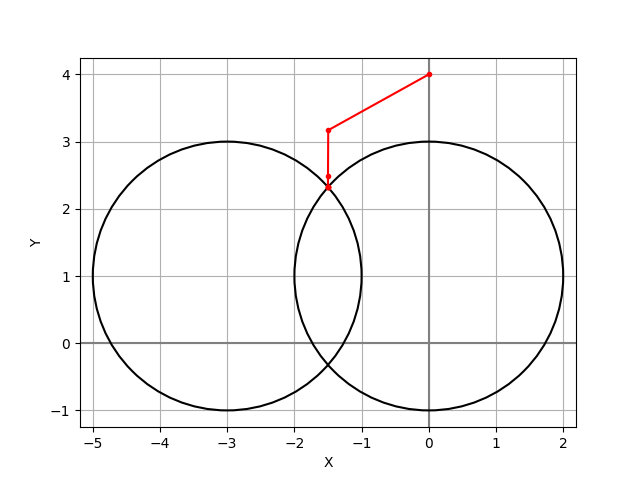
Окружности пересекаются в двух точках

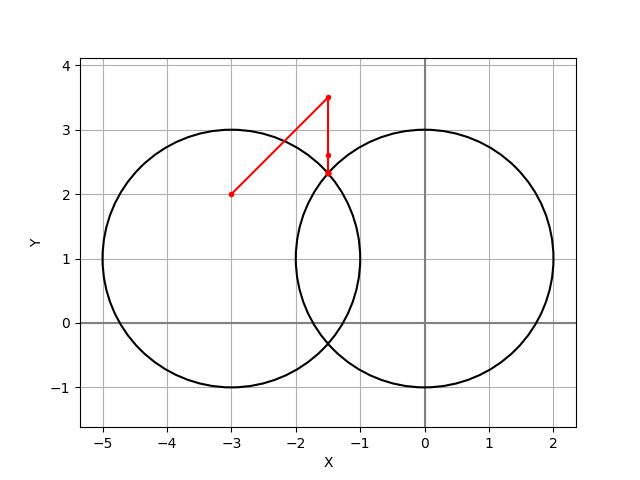
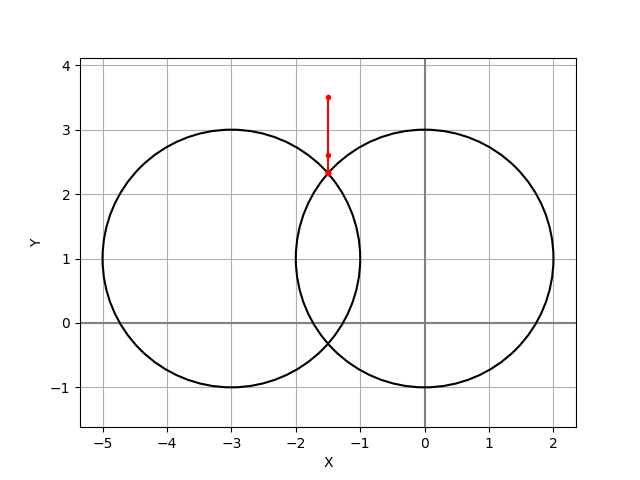
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x0** | **Аналитические производные** | | | |
| **k** | **Погрешность** | **b** | **результат** |
| (0, 4)  Не лежит на осях симметрии | 7 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -1.500000  2.322876 |
| (3, 1)  Лежит на оси, соединяющей центры окружностей | 28 | -3.07E+00  -2.28E+00 | 1.776356839400250e-15 | 1.569039  4.603870 |
| (-1.5, 3.5)  Лежит на оси, перпендикулярной оси, соединяющей центры окружностей, и пересекающей ее на равных расстояниях от центров окружностей | 6 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -1.500000  2.322876 |
| (-3, 2)  Лежит внутри одной из окружностей | 7 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -1.500000  2.322876 |
| **Численные производные** | | | | |
| (0, 4)  Не лежит на осях симметрии | 6 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -1.500000  2.322876 |
| (3, 1)  Лежит на оси, соединяющей центры окружностей | 7 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -1.500000  2.322876 |
| (-1.5, 3.5)  Лежит на оси, перпендикулярной оси, соединяющей центры окружностей, и пересекающей ее на равных расстояниях от центров окружностей | 5 | -1.500000  2.322876 | 1.000000000000000e+00 | -1.500000  2.322876 |
| (-3, 2)  Лежит внутри одной из окружностей | 7 | -1.500000  2.322876 | 1.000000000000000e+00 | -1.500000  2.322876 |





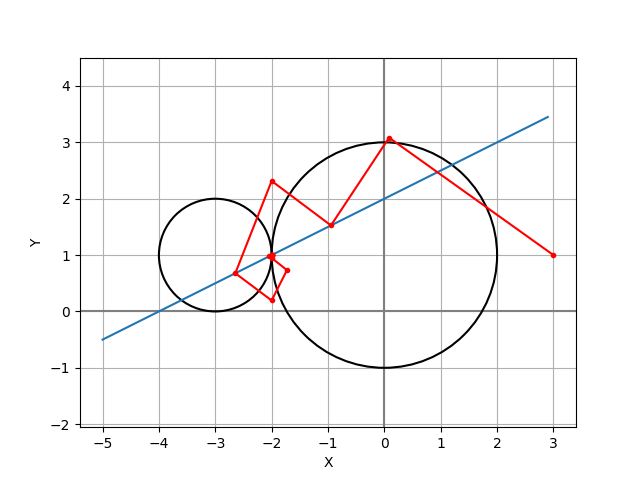
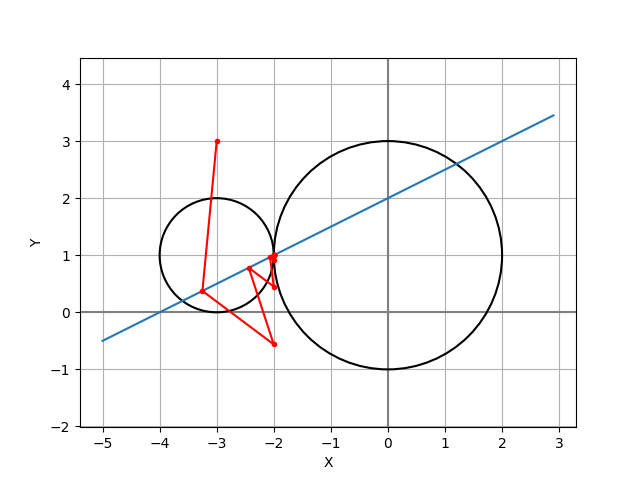
Численные производные

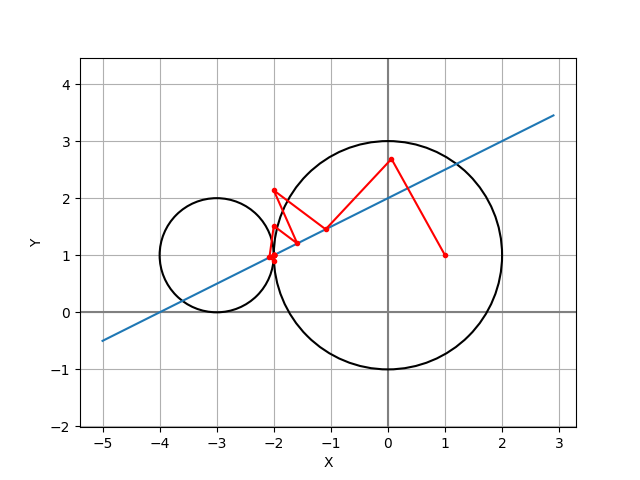
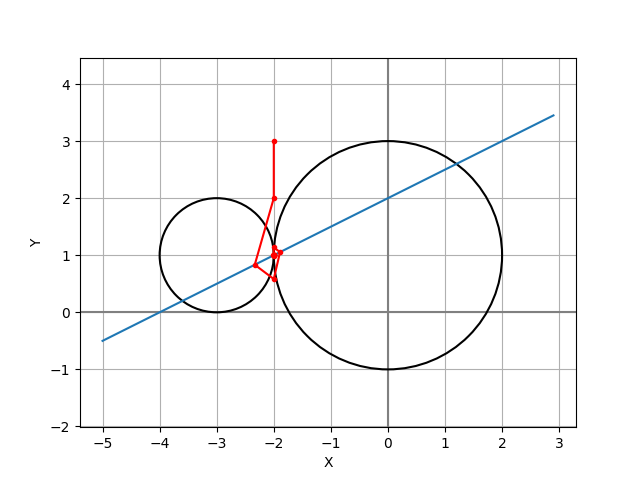




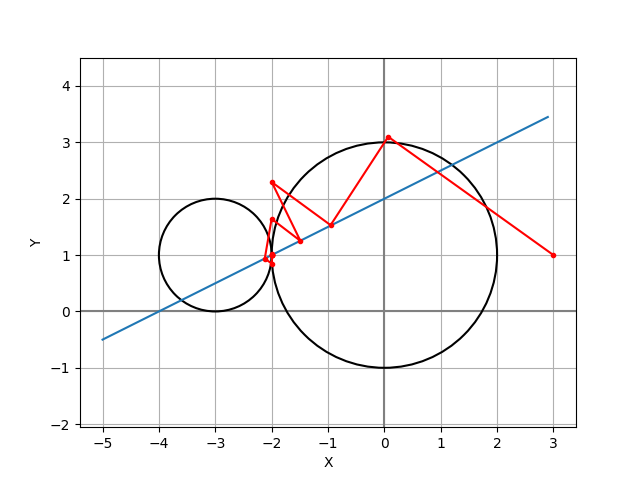
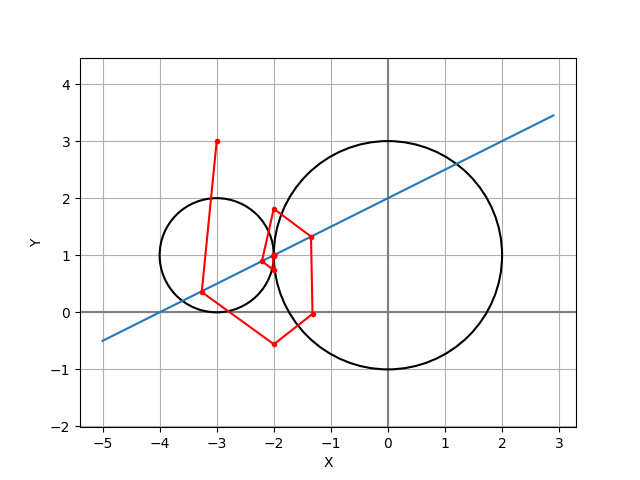
*Две пересекающиеся окружности и прямая*

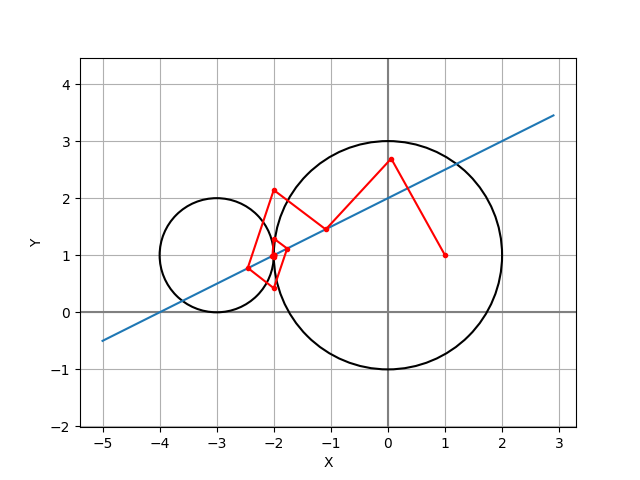
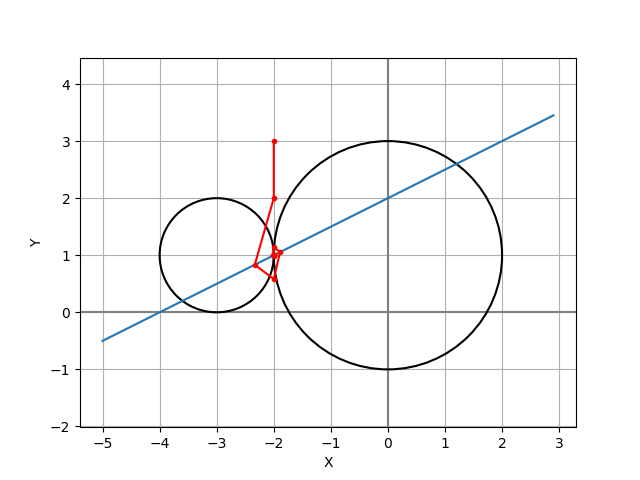
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x0** | **Аналитические производные** | | | |
| **k** | **||FK||** | **b** | **результат** |
| (-3, 3)  Не лежит на осях симметрии | 14 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -2.000000  1.000000 |
| (3, 1)  Лежит на оси, соединяющей центры окружностей | 14 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -2.000000  1.000000 |
| (-2, 3)  Лежит на оси, перпендикулярной оси, соединяющей центры окружностей, и пересекающей ее на равных расстояниях от центров окружностей | 13 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -2.000000  1.000000 |
| (1, 1)  Лежит внутри одной из окружностей | 15 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -2.000000  1.000000 |
| **Численные производные** | | | | |
| (-3, 3)  Не лежит на осях симметрии | 13 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -2.000000  1.000000 |
| (3, 1)  Лежит на оси, соединяющей центры окружностей | 13 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -2.000000  1.000000 |
| (-2, 3)  Лежит на оси, перпендикулярной оси, соединяющей центры окружностей, и пересекающей ее на равных расстояниях от центров окружностей | 11 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -2.000000  1.000000 |
| (1, 1)  Лежит внутри одной из окружностей | 13 | 0.00E+00  0.00E+00 | 1.000000000000000e+00 | -2.000000  1.000000 |





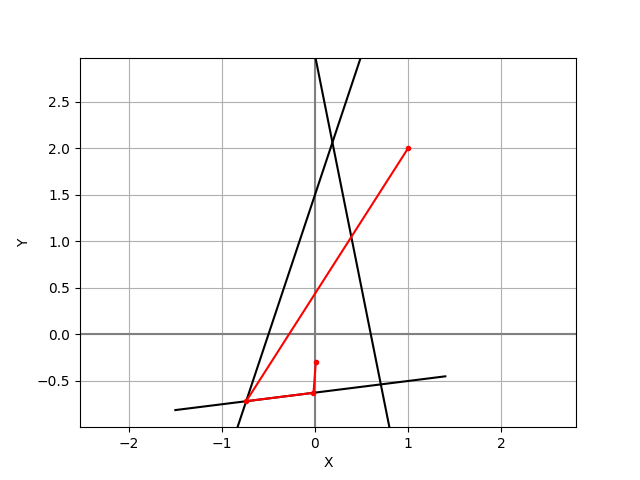
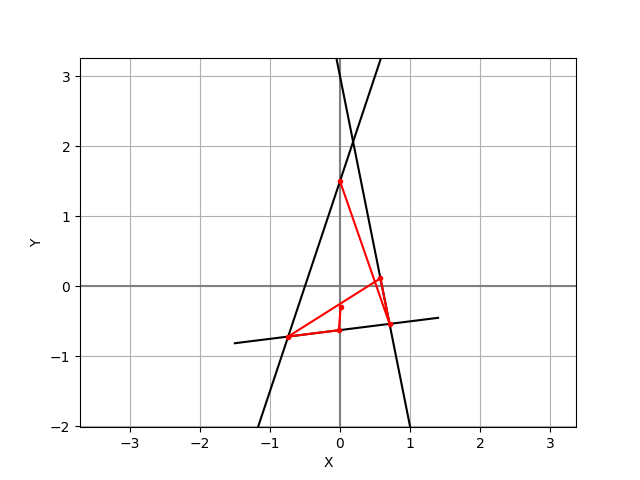
Численные производные

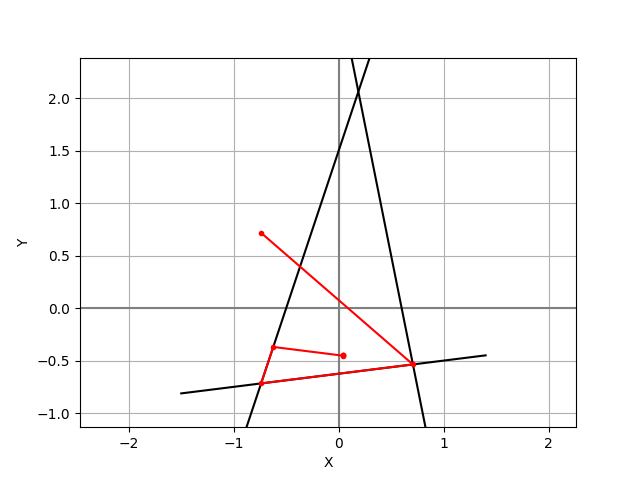
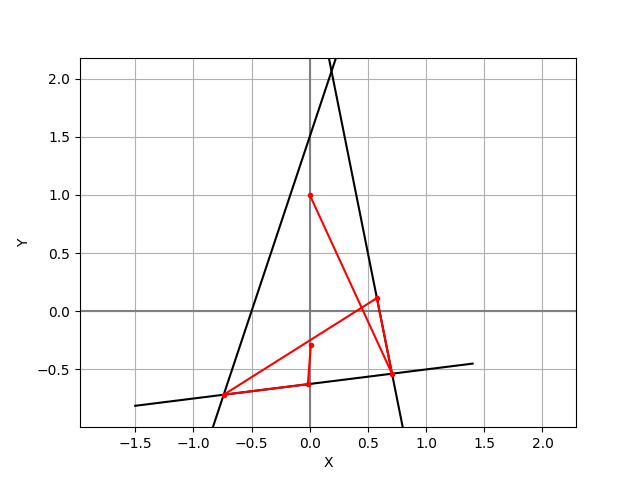




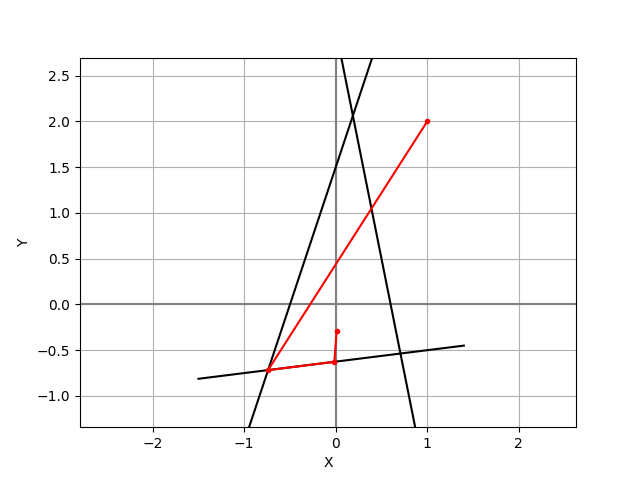
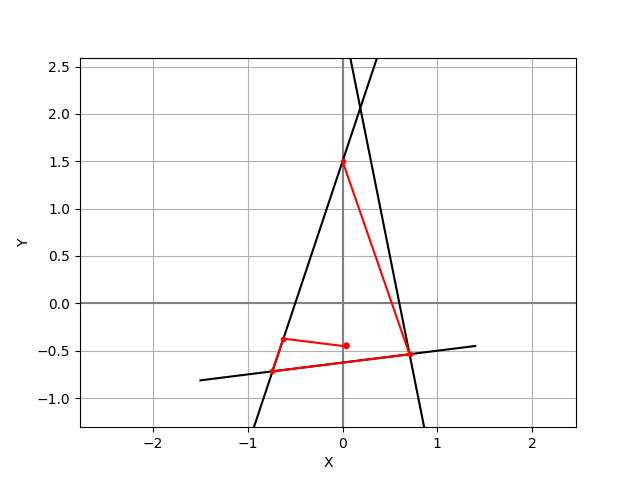
Три попарно пересекающиеся прямые

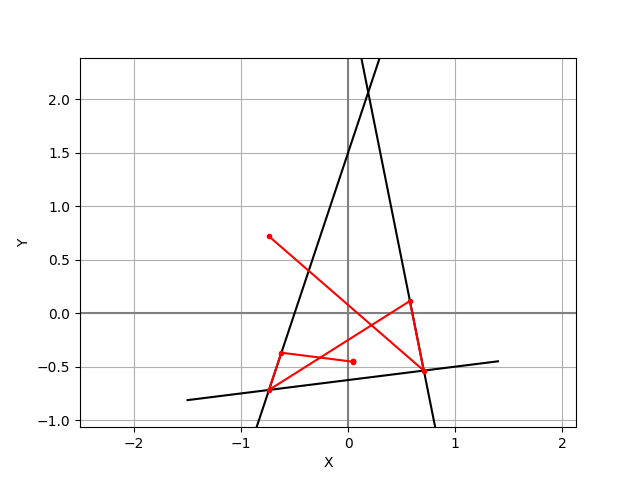
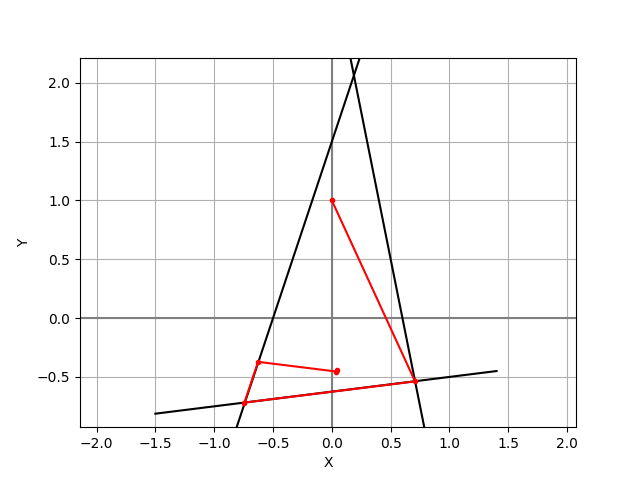
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вариант 2 (исключение уравнений из системы)** | | | | |
| **x0** | **Аналитические производные** | | | |
| **k** | **||FK||** | **b** | **результат** |
| (0, 1.5)  Начальное приближение находится на одной из прямых | 5 | 5.554808478695461e+00 | 1.776356839400250e-15 | 0.009519  -0.290802 |
| (1, 2)  Начальное приближение находится не в области пересечения прямых | 3 | 5.554808478695461e+00 | 1.776356839400250e-15 | 0.009519  -0.290802 |
| (0, 1)  Начальное приближение находится в области пересечения прямых | 5 | 5.554808478695461e+00 | 1.776356839400250e-15 | 0.009519  -0.290802 |
| (-0.739, 0.717)  Начальное приближение находится в точке пересечения двух прямых | 7 | 5.439051859129036e+00 | 1.776356839400250e-15 | 0.042576  -0.443418 |
| **Численные производные** | | | | |
| (0, 1.5)  Начальное приближение находится на одной из прямых | 9 | 5.436746487565357e+00 | 1.776356839400250e-15 | 0.040801  -0.444646 |
| (1, 2)  Начальное приближение находится не в области пересечения прямых | 3 | 5.555652044797532e+00 | 1.776356839400250e-15 | 0.010107  -0.290530 |
| (0, 1)  Начальное приближение находится в области пересечения прямых | 5 | 5.435163891487758e+00 | 1.776356839400250e-15 | 0.039604  -0.444551 |
| (-0.739, 0.717)  Начальное приближение находится в точке пересечения двух прямых | 7 | 5.438671535204976e+00 | 1.776356839400250e-15 | 0.042291  -0.443445 |





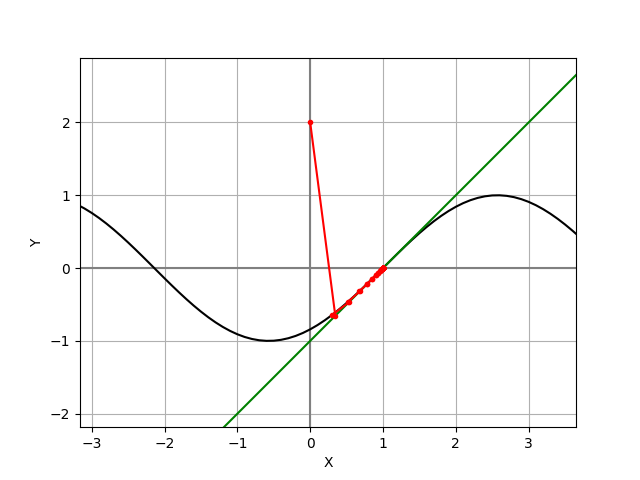
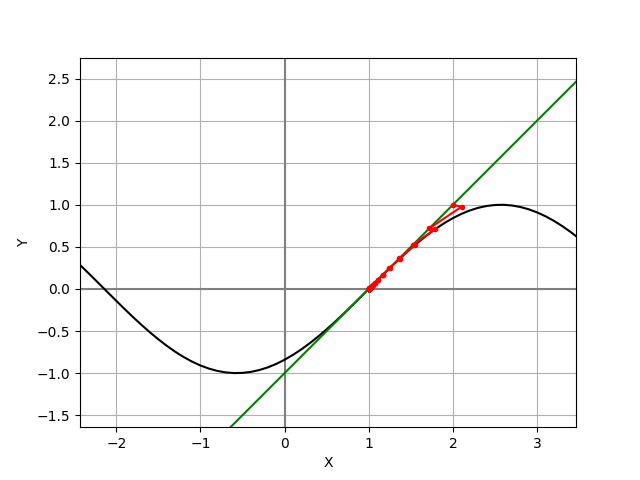
Численные производные

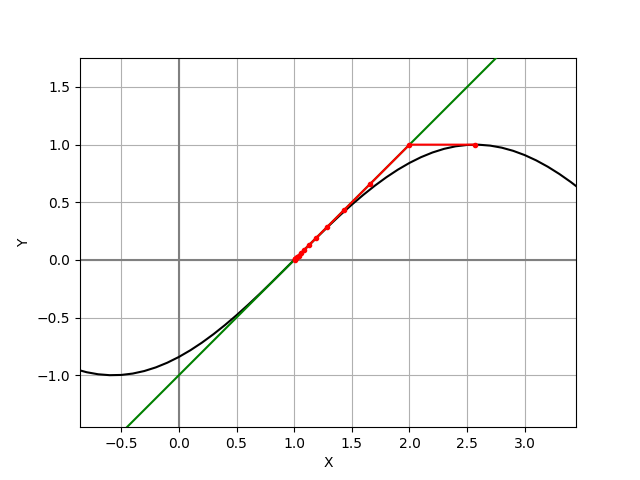




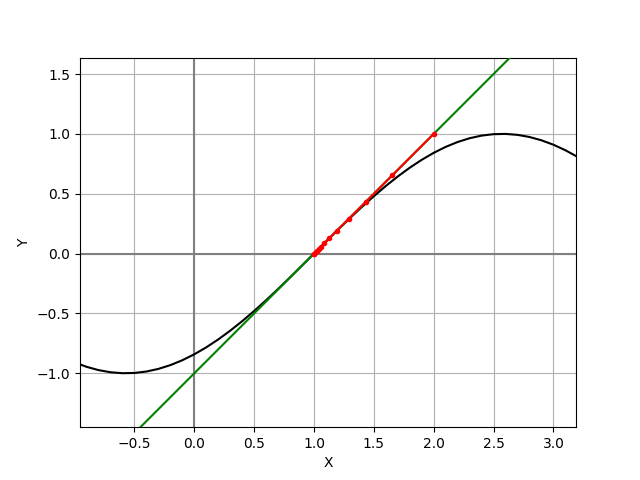
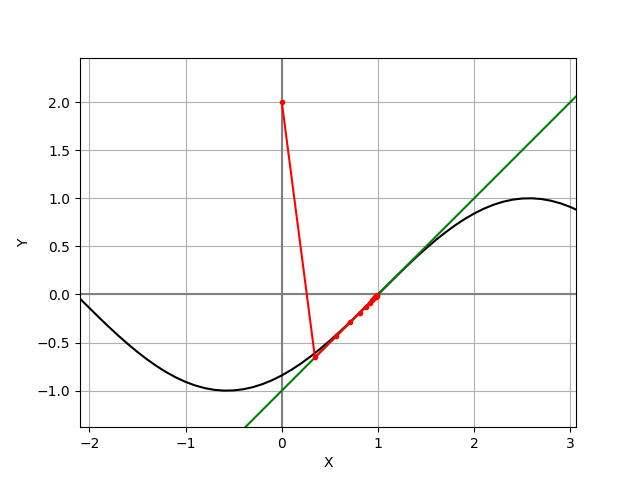
Синусоида и прямая

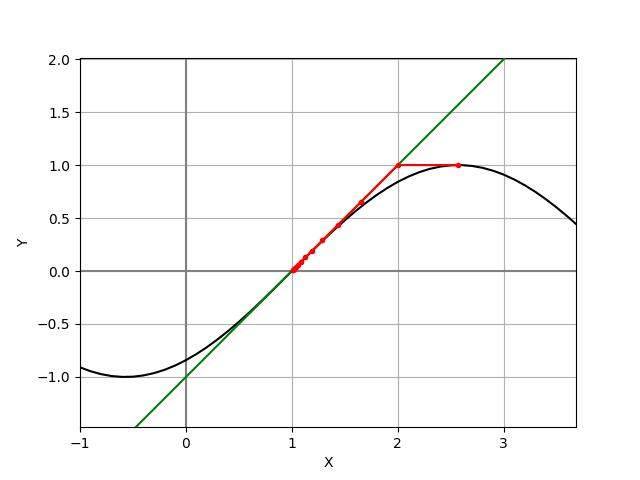
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x0** | **Аналитические производные** | | | |
| **k** | **Погрешность** | **b** | **результат** |
| (0, 2)  Приближение не находится ни на одном из уравнений | 12 | 7.48E-03  7.48E-03 | 1.000000000000000e+00 | 0.992523  -0.007477 |
| (2, 1)  Приближение находится на прямой | 15 | -2.22E-03  -2.22E-03 | 1.000000000000000e+00 | 1.002215  0.002215 |
| (2.571, 1)  Приближение находится в локальном экстремуме | 15 | -3.32E-03  -3.32E-03 | 1.000000000000000e+00 | 1.003323  0.003323 |
| **Численные производные** | | | | |
| (0, 2)  Приближение не находится ни на одном из уравнений | 13 | 1.07E-02  1.07E-02 | 6.250000000000000e-02 | 0.993773  -0.006227 |
| (2, 1)  Приближение находится на прямой | 12 | 5.23E-04  5.23E-04 | 1.000000000000000e+00 | 0.999477  -0.000523 |
| (2.571, 1)  Приближение находится в локальном экстремуме | 13 | -6.69E-03  -6.69E-03 | 3.725290298461914e-09 | 1.006688  0.006688 |



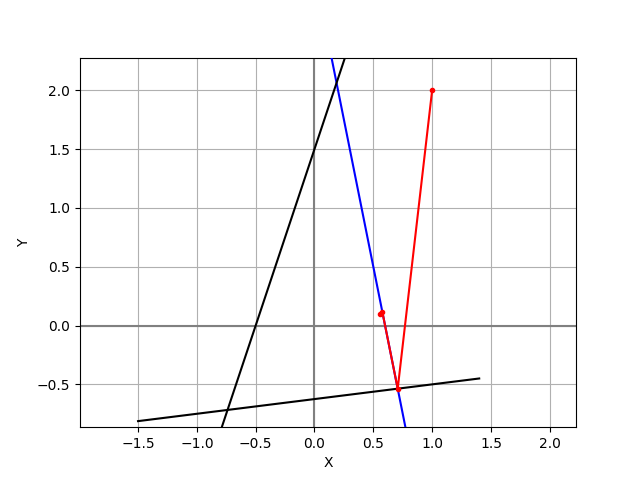
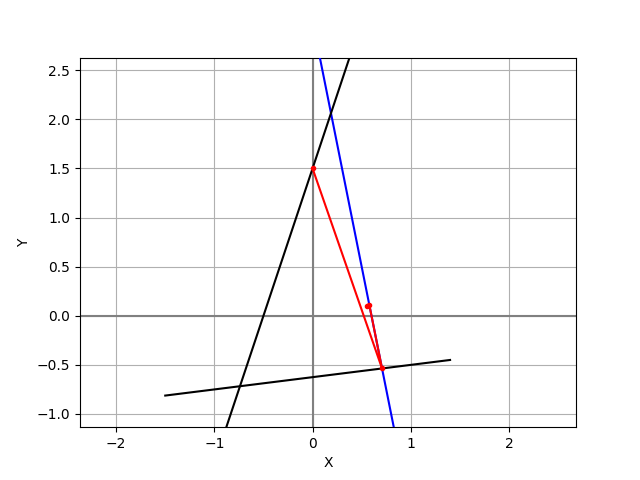
Численные производные

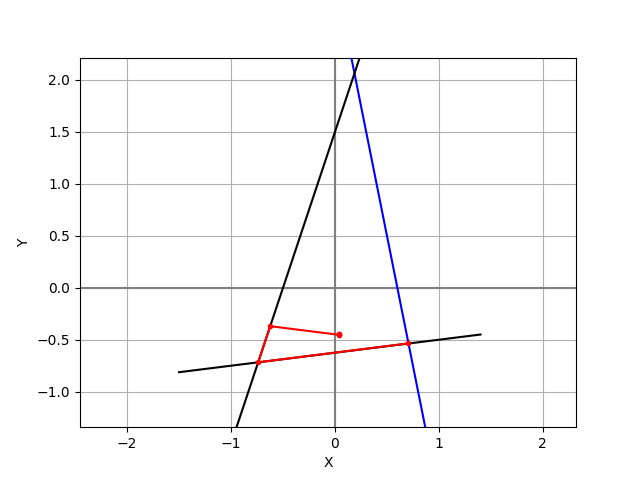




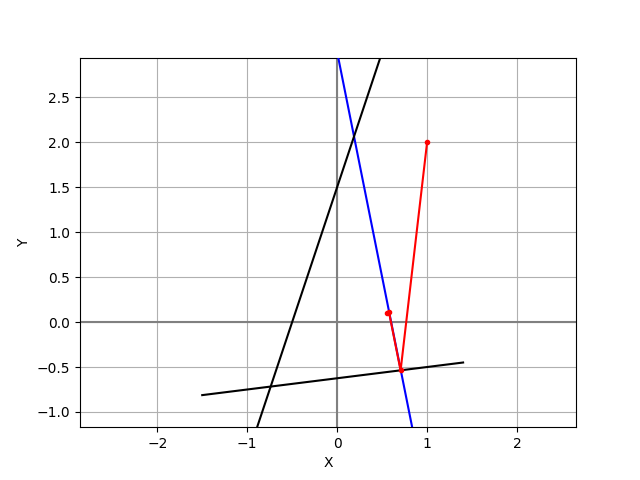
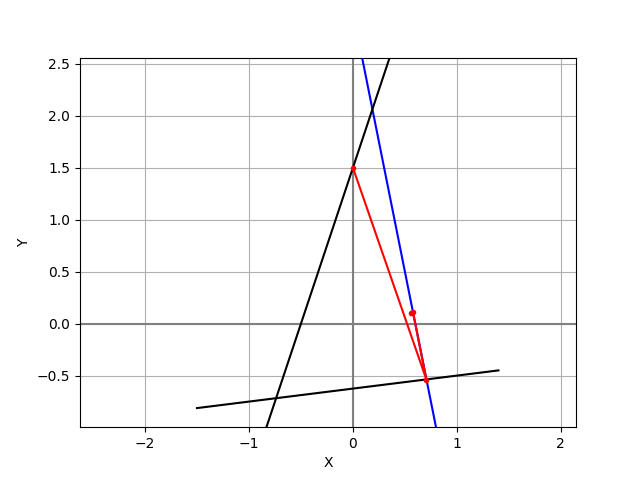
Три попарно пересекающиеся прямые (влияние взвешивания)

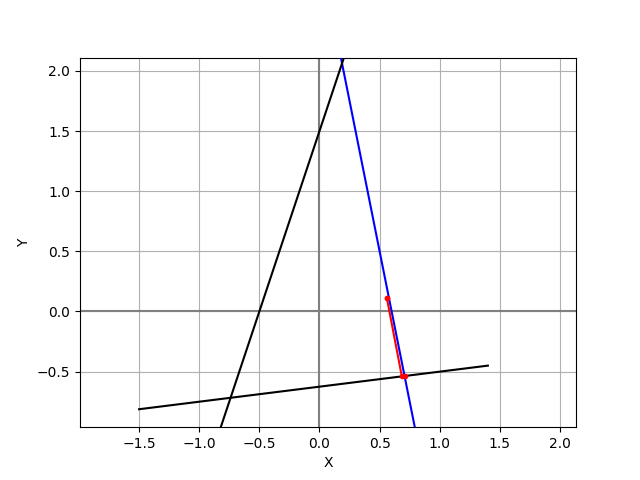
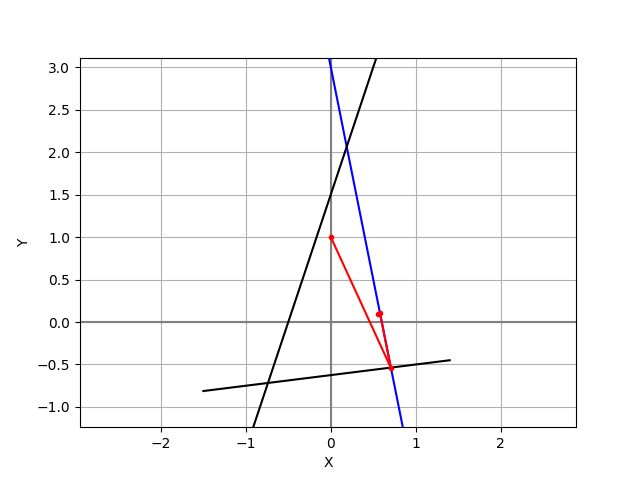
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вариант 2 (исключение уравнений из системы)** | | | | |
| **x0** | **Аналитические производные** | | | |
| **k** | **||FK||** | **b** | **результат** |
| (0, 1.5)  Начальное приближение находится на одной из прямых | 3 | 8.158068991863509e+00 | 1.776356839400250e-15 | 0.556793  0.100208 |
| (1, 2)  Начальное приближение находится не в области пересечения прямых | 3 | 8.158068991863507e+00 | 1.776356839400250e-15 | 0.556793  0.100208 |
| (0, 1)  Начальное приближение находится в области пересечения прямых | 3 | 8.158068991863509e+00 | 1.776356839400250e-15 | 0.556793  0.100208 |
| (0.707, -0.535)  Начальное приближение находится в точке пересечения двух прямых | 15 | 5.439051859129036e+00 | 1.776356839400250e-15 | 0.042576  -0.443418 |
| Численные производные | | | | |
| (0, 1.5)  Начальное приближение находится на одной из прямых | 3 | 8.151281312458016e+00 | 1.776356839400250e-15 | 0.557063  0.097523 |
| (1, 2)  Начальное приближение находится не в области пересечения прямых | 3 | 8.165466849619706e+00 | 1.776356839400250e-15 | 0.554730  0.100248 |
| (0, 1)  Начальное приближение находится в области пересечения прямых | 3 | 8.152245897861976e+00 | 1.776356839400250e-15 | 0.556974  0.097831 |
| (0.707, -0.535)  Начальное приближение находится в точке пересечения двух прямых | 3 | 8.175951237431290e+00 | 1.776356839400250e-15 | 0.559571  0.110525 |





Численные производные





1. **Выводы**

* При решении системы, в которой содержатся окружности, наиболее быстрая сходимость наблюдается если начальное приближение лежит на оси, перпендикулярной оси, соединяющей центры окружностей, и пересекающей ее на равных расстояниях от центров окружностей.

Наиболее медленная сходимость наблюдается если начальное приближение находится внутри одной из окружностей

При вычислении производной теоретически чем меньше шаг – тем лучше. Однако при слишком маленьких шагах возникает большая погрешность из-за деления маленького числа на другое маленькое число

* Использование численных производных в общем случае значительно ускоряет сходимость, но решение содержит бóльшую погрешность.

Численные производные позволяют решить систему в случае, когда аналитические производные порождают вырожденный Якобиан, за счет погрешности в вычислении.

Использование численных производных в системах, где нет аналитического решения, сходимость значительно хуже, чем при использовании аналитических.

* Добавление прямой в СНУ значительно ускоряет сходимость

1. **Приложение**

**SNE.cpp:**

#include "SNE\_Header.h"

#include "Matrix.h"

#pragma region Derivatives

double GetFDerivative1(int i, int j, double\* vars)

{

// i - номер уранения системы, j - номер переменной для дифференцирования

if (i == 0) {

if (j == 0)

return 2 \* (vars[0] - 2);

else if (j == 1)

return 2 \* vars[1];

}

else if (i == 1) {

if (j == 0)

return 2 \* (vars[0] + 3);

else if (j == 1)

return 2 \* (vars[1] - 1);

}

}

double GetFDerivative2(int i, int j, double\* vars)

{

// i - номер уранения системы, j - номер переменной для дифференцирования

if (i == 0) {

if (j == 0)

return 2 \* vars[0];

else if (j == 1)

return 2 \* (vars[1] - 1);

}

else if (i == 1) {

if (j == 0)

return 2 \* (vars[0] + 3);

else if (j == 1)

return 2 \* (vars[1] - 1);

}

}

double GetFDerivative3(int i, int j, double\* vars)

{

// i - номер уранения системы, j - номер переменной для дифференцирования

if (i == 0) {

if (j == 0)

return 2 \* vars[0];

else if (j == 1)

return 2 \* (vars[1] - 1);

}

else if (i == 1) {

if (j == 0)

return 2 \* (vars[0] + 3);

else if (j == 1)

return 2 \* (vars[1] - 1);

}

}

double GetFDerivative4(int i, int j, double\* vars)

{

// i - номер уранения системы, j - номер переменной для дифференцирования

if (i == 0) {

if (j == 0)

return 2 \* vars[0];

else if (j == 1)

return 2 \* (vars[1] - 1);

}

else if (i == 1) {

if (j == 0)

return 2 \* (vars[0] + 3);

else if (j == 1)

return 2 \* (vars[1] - 1);

}

else if (i == 2) {

if (j == 0)

return 1;

else if (j == 1)

return -2;

}

}

double GetFDerivative5(int i, int j, double\* vars)

{

// i - номер уранения системы, j - номер переменной для дифференцирования

if (i == 0) {

if (j == 0)

return 5;

else if (j == 1)

return 1;

}

else if (i == 1) {

if (j == 0)

return -1;

else if (j == 1)

return 8;

}

else if (i == 2) {

if (j == 0)

return 6;

else if (j == 1)

return -2;

}

}

double GetFDerivative6(int i, int j, double\* vars)

{

// i - номер уранения системы, j - номер переменной для дифференцирования

if (i == 0) {

if (j == 0)

return 1;

else if (j == 1)

return -1;

}

else if (i == 1) {

if (j == 0)

return cos(vars[0] - 1);

else if (j == 1)

return -1;

}

}

#pragma endregion Derivatives

#pragma region Functions

double GetF1(int i, double\* vars)

{

// i - номер уравнения системы

if (i == 0)

return (vars[0] - 2) \* (vars[0] - 2) + vars[1] \* vars[1] - 4;

else if (i == 1)

return (vars[0] + 3) \* (vars[0] + 3) + (vars[1] - 1) \* (vars[1] - 1) - 1;

}

double GetF2(int i, double\* vars)

{

// i - номер уравнения системы

if (i == 0)

return vars[0] \* vars[0] + (vars[1] - 1) \* (vars[1] - 1) - 4;

else if (i == 1)

return (vars[0] + 3) \* (vars[0] + 3) + (vars[1] - 1) \* (vars[1] - 1) - 1;

}

double GetF3(int i, double\* vars)

{

// i - номер уравнения системы

if (i == 0)

return vars[0] \* vars[0] + (vars[1] - 1) \* (vars[1] - 1) - 4;

else if (i == 1)

return (vars[0] + 3) \* (vars[0] + 3) + (vars[1] - 1) \* (vars[1] - 1) - 4;

}

double GetF4(int i, double\* vars)

{

// i - номер уравнения системы

if (i == 0)

return vars[0] \* vars[0] + (vars[1] - 1) \* (vars[1] - 1) - 4;

else if (i == 1)

return (vars[0] + 3) \* (vars[0] + 3) + (vars[1] - 1) \* (vars[1] - 1) - 1;

else if (i == 2)

return vars[0] - 2 \* vars[1] + 4;

}

double GetF5(int i, double\* vars)

{

// i - номер уравнения системы

if (i == 0)

return 5 \* vars[0] + vars[1] - 3;

else if (i == 1)

return -vars[0] + 8 \* vars[1] + 5;

else if (i == 2)

return 6 \* vars[0] -2 \* vars[1] + 3;

}

double GetF6(int i, double\* vars)

{

// i - номер уравнения системы

if (i == 0)

return vars[0] - vars[1] - 1;

else if (i == 1)

return sin(vars[0] - 1) - vars[1];

}

#pragma endregion Functions

double SNE::GetFDerivativeNumeric(int i, int j)

{

std::copy(varsK, varsK + n, temp);

temp[j] += numericStep;

return (funcs[testNum](i, temp) - funcs[testNum](i, varsK)) / numericStep;

}

void SNE::Input(FILE\* params, FILE\* x0)

{

fscanf\_s(params, "%d", &maxIter);

fscanf\_s(params, "%d", &n);

fscanf\_s(params, "%d", &m);

fscanf\_s(params, "%lf", &solutionEps);

fscanf\_s(params, "%lf", &betaEps);

fscanf\_s(params, "%lf", &numericStep);

fscanf\_s(params, "%d", &useNumericDerivative);

AllocateMemory();

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

fscanf\_s(x0, "%lf", &varsK[i]);

}

void SNE::AllocateMemory()

{

Ak = new double \*[m];

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

Ak[i] = new double[n];

indexes = new int[n];

varsK = new double[n];

deltaVarsK = new double[n];

Fk = new double[m];

Fv = new double[m];

temp = new double[(m > n) ? m : n];

}

void SNE::CalculateAk()

{

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

{

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

if (useNumericDerivative)

Ak[i][j] = GetFDerivativeNumeric(i, j);

else

Ak[i][j] = derivatives[testNum](i, j, varsK);

}

}

}

void SNE::CalculateFk(double \*x, double\* out)

{

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

{

out[i] = -funcs[testNum](i, x);

}

}

int SNE::CalculateXkNext(double normPrev)

{

double betaK, normCurr, actualBeta;

for (betaK = 1, normCurr = normPrev + 1, actualBeta = -1; betaK > betaEps && normCurr > normPrev; betaK /= 2)

{

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

temp[i] = varsK[i] + betaK \* deltaVarsK[i];

actualBeta = betaK;

CalculateFk(temp, Fv);

normCurr = CalculateNorm(Fv);

}

std::copy(temp, temp + n, varsK);

printf\_s("\t%.15e", actualBeta);

if (betaK <= betaEps)

return 1;

else

return 0;

}

double SNE::CalculateNorm(double\* in)

{

double temp = 0;

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

temp += in[i] \* in[i];

return sqrt(temp);

}

int SNE::SolveSNU()

{

DenseMatrix denseMatrix;

double FkNorm, F0Norm;

if (n > m)

return -1; // невозможно решить СЛАУ

CalculateFk(varsK, Fk); // вычисляется Fk от начального приближения

FkNorm = F0Norm = CalculateNorm(Fk); // норма F0

printf\_s("Iteration\tNorm Fk\tBeta k\n");

FILE\* graphFile;

fopen\_s(&graphFile, "GraphSolutions.txt", "w");

OutputDataForGraph(graphFile);

int i;

for (i = 0; i < maxIter && FkNorm / F0Norm > solutionEps; i++)

{

CalculateFk(varsK, Fk); // вычисляется Fk от xk данной итерации

CalculateAk(); // вычисляется матрица Ak в плотном формате

if (m > n)

ResolveDimensions2(); // в случае m>n избавляемся от m-n элементов

else if (n > m)

ResolveDimensions1(); // в случае n>m избавляемся от n-m элементов

FkNorm = CalculateNorm(Fk); // норма Fk

denseMatrix.Input(n, Ak, Fk);

denseMatrix.SolveSLAE();

deltaVarsK = denseMatrix.x; // ввод и решение СЛАУ с помощью метода Гаусса из 1 лаб, получаем Δx

printf\_s("%d\t\t%.15e", i + 1, FkNorm); // вывод итерации и текущей нормы

if (CalculateXkNext(FkNorm) == 1) // итерационный процесс x(k+1)=xk + β Δx

return 1; // выход был из-за β < eps

if (n > m)

OrderSolution();

PrintSolutionVector();

OutputDataForGraph(graphFile);

printf\_s("\n");

}

fclose(graphFile);

if (i >= maxIter)

return 2;

else

return 0;

}

void SNE::ResolveDimensions2()

{

// сортировка Fk и Ak

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

for (int j = 0; j < m - 1; j++) {

if (fabs(Fk[j]) < fabs(Fk[j + 1]))

{

std::swap(Fk[j], Fk[j + 1]);

std::swap(Ak[j], Ak[j + 1]);

}

}

}

}

void SNE::ResolveDimensions1()

{

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

indexes[i] = i;

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

{

temp[j] = fabs(Ak[0][j]);

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

{

if (fabs(Ak[i][j]) < temp[j])

temp[j] = fabs(Ak[i][j]);

}

}

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

for (int j = 0; j < m - 1; j++) {

if (temp[j] < temp[j + 1])

{

std::swap(temp[j], temp[j + 1]);

std::swap(indexes[j], indexes[j + 1]);

for (int k = 0; k < m; k++)

std::swap(Ak[k][j], Ak[k][j + 1]);

}

}

}

}

void SNE::OrderSolution()

{

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

for (int j = 0; j < m - 1; j++) {

if (indexes[j] < indexes[j + 1])

{

std::swap(indexes[j], indexes[j + 1]);

std::swap(varsK[j], varsK[j + 1]);

}

}

}

}

void SNE::PrintSolutionVector()

{

printf\_s("\nSolution vector:\n");

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

{

printf\_s("%lf\n", varsK[i]);

}

}

void SNE::OutputDataForGraph(FILE\* solVector)

{

fprintf\_s(solVector, "%lf %lf\n", varsK[0], varsK[1]);

}

void SNE::OutputSolutionvector(FILE\* out)

{

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

{

fprintf\_s(out, "%lf\n", varsK[i]);

}

}

SNE::SNE()

{

funcs.push\_back(GetF1);

funcs.push\_back(GetF2);

funcs.push\_back(GetF3);

funcs.push\_back(GetF4);

funcs.push\_back(GetF5);

funcs.push\_back(GetF6);

derivatives.push\_back(GetFDerivative1);

derivatives.push\_back(GetFDerivative2);

derivatives.push\_back(GetFDerivative3);

derivatives.push\_back(GetFDerivative4);

derivatives.push\_back(GetFDerivative5);

derivatives.push\_back(GetFDerivative6);

}

**SNE.h**

#pragma once

#include <iostream>

#include <algorithm>

#include <vector>

#include <functional>

double GetF1(int i, double\* vars);

double GetF2(int i, double\* vars);

double GetF3(int i, double\* vars);

double GetF4(int i, double\* vars);

double GetF5(int i, double\* vars);

double GetF6(int i, double\* vars);

double GetFDerivative1(int i, int j, double\* vars);

double GetFDerivative2(int i, int j, double\* vars);

double GetFDerivative3(int i, int j, double\* vars);

double GetFDerivative4(int i, int j, double\* vars);

double GetFDerivative5(int i, int j, double\* vars);

double GetFDerivative6(int i, int j, double\* vars);

class SNE {

public:

double \*varsK, \*deltaVarsK, \*\*Ak, \*Fk, \*Fv;

double solutionEps, betaEps, numericStep;

int maxIter, n, m, \*indexes;

bool useNumericDerivative;

int testNum;

std::vector<std::function<double(int i, double\* x)>> funcs;

std::vector<std::function<double(int i, int j, double\* vars)>> derivatives;

void Input(FILE\* params, FILE\* x0);

void CalculateAk();

void CalculateFk(double\* x, double\* out);

int CalculateXkNext(double normPrev);

double CalculateNorm(double \*in);

double GetFDerivativeNumeric(int i, int j);

int SolveSNU();

void ResolveDimensions2();

void ResolveDimensions1();

void OrderSolution();

void PrintSolutionVector();

void OutputDataForGraph(FILE\* solVector);

void OutputSolutionvector(FILE\* out);

SNE();

private:

double \*temp;

void AllocateMemory();

};