**Отчет**

**по Методам численного анализа**

Исполнитель

студент 161 группы

специальности

Прикладная математика

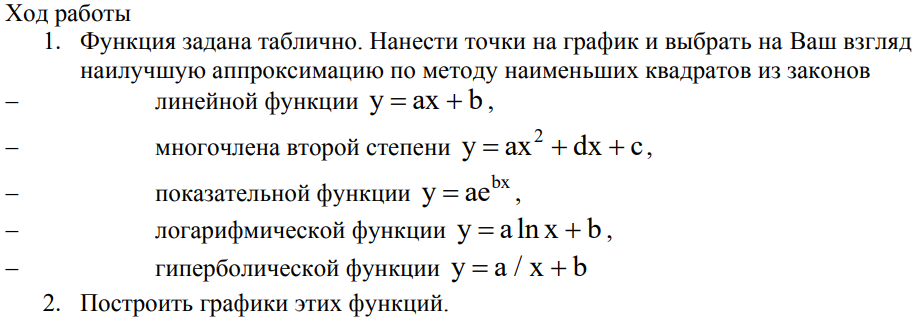
Борис Д. Ю.

«18» Мая 2018 г.

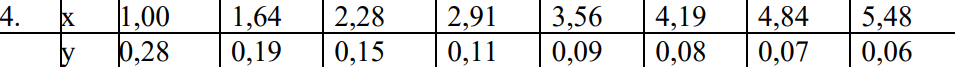
Гродно, 2018

**ЛАБОРАТОРНАЯ РАБОТА № 9**

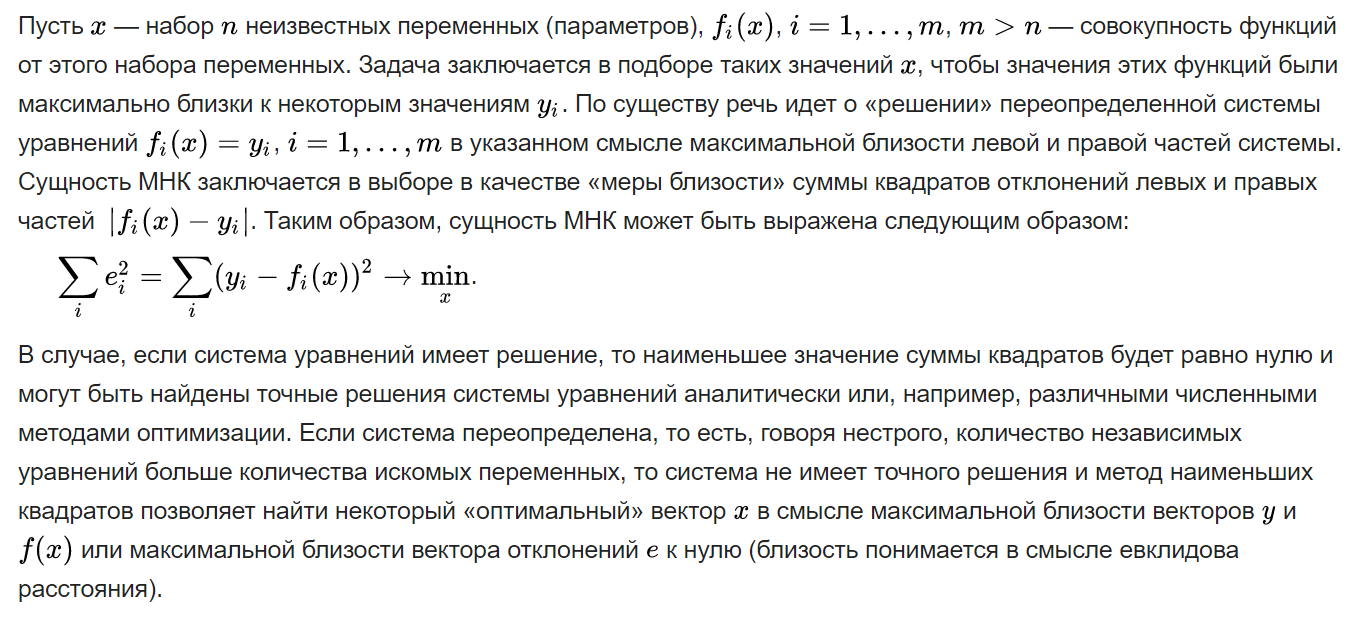
**Тема. Метод наименьших квадратов**



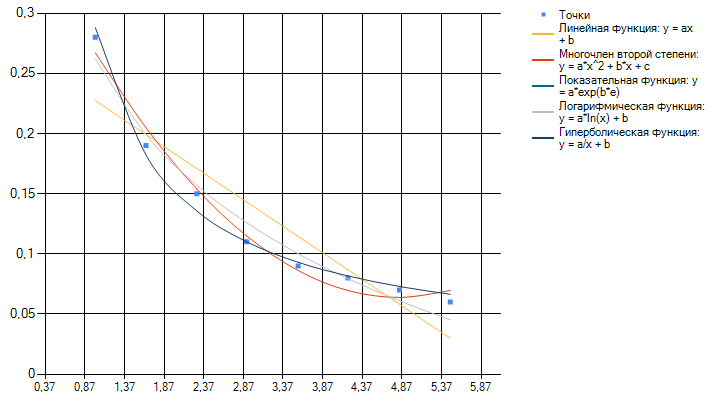
Условие:



1. **Теоретический материал**



1. **Результат вычислений**



P.S. График показательной функции был убран т.к. он вообще не сходится с точками

1. **Текст программы**

public List<double> linear(Sums s, double[] X, double[] Y)

{

double[] coefficient = new double [2];

double[,] sys\_matrix = {{s.sum2(X), s.sum(X) },

{s.sum(X), X.Length } };

double[] solution\_row = {s.sum\_x\_mult\_y(X, Y), s.sum(Y) };

alglib.rmatrixsolve(sys\_matrix, 2, solution\_row, out int t, out alglib.densesolverreport c, out coefficient);

List<double> cof = new List<double>();

cof.AddRange(coefficient);

return cof;

}

static List<double> quadratic(Sums s, double[] X, double[] Y)

{

double[] coefficient = new double[3];

double[,] sys\_matrix = {{s.sum4(X), s.sum3(X),s.sum2(X) },

{s.sum3(X),s.sum2(X), s.sum(X) },

{s.sum2(X),s.sum(X),X.Length } };

double[] solution\_row = {s.sum\_x\_x\_mult\_y(X, Y), s.sum\_x\_mult\_y(X,Y),s.sum(Y) };

alglib.rmatrixsolve(sys\_matrix, 3, solution\_row, out int t, out alglib.densesolverreport c, out coefficient);

List<double> cof = new List<double>();

cof.AddRange(coefficient);

return cof;

}

static List<double> exponential(Sums s,double[] X, double[] Y)

{

double[] coefficient = new double[2];

double[,] sys\_matrix = {{s.sum2(X), s.sum(X)},

{s.sum(X),X.Length }};

double[] solution\_row = { s.sum\_y\_on\_log\_x(X, Y), s.sum\_log\_x(Y)};

alglib.rmatrixsolve(sys\_matrix, 2, solution\_row, out int t, out alglib.densesolverreport c, out coefficient);

coefficient[1] = Math.Exp(coefficient[1]);

List<double> cof = new List<double>();

cof.AddRange(coefficient);

return cof;

}

static List<double> log(Sums s, double[] X, double[] Y)

{

double[] coefficient = new double[2];

double[,] sys\_matrix = {{s.sum\_mult2\_log\_x(X), s.sum\_log\_x(X) },

{s.sum\_log\_x(X), X.Length } };

double[] solution\_row = { s.sum\_y\_on\_log\_x(X, Y), s.sum(Y) };

alglib.rmatrixsolve(sys\_matrix, 2, solution\_row, out int t, out alglib.densesolverreport c, out coefficient);

List<double> cof = new List<double>();

cof.AddRange(coefficient);

return cof;

}

static List<double> hyperbolic(Sums s, double[] X, double[] Y)

{

double[] coefficient = new double[2];

double[,] sys\_matrix = {{s.sum\_dev\_1\_on\_x\_x(X), s.sum\_dev\_1\_on\_x(X)},

{s.sum\_dev\_1\_on\_x(X),X.Length }};

double[] solution\_row = { s.sum\_dev\_y\_on\_x(X, Y), s.sum(Y) };

alglib.rmatrixsolve(sys\_matrix, 2, solution\_row, out int t, out alglib.densesolverreport c, out coefficient);

List<double> cof = new List<double>();

cof.AddRange(coefficient);

return cof;

}

**Соответствующие вызовы методов:**

chart1.Series[0].Points.DataBindXY(MasX, MasY);

Sums s = new Sums();

List<double> X = new List<double>();

List<double> Y = new List<double>();

List<double> coeffish = linear(s, MasX, MasY);

for ( var i = 0 ; i < MasX.Length ; i++ )

{

X.Add(MasX[i]);

Y.Add(coeffish[0] \* MasX[i] + coeffish[1]);

}

chart1.Series[1].Points.DataBindXY(X, Y);

X.Clear();

Y.Clear();

coeffish.Clear();

coeffish = quadratic(s, MasX, MasY);

for ( var i = 0 ; i < MasX.Length ; i++ )

{

X.Add(MasX[i]);

Y.Add(coeffish[0] \* ( MasX[i] \* MasX[i] ) + coeffish[1] \* MasX[i] + coeffish[2]);

}

chart1.Series[2].Points.DataBindXY(X, Y);

X.Clear();

Y.Clear();

coeffish.Clear();

coeffish = exponential(s, MasX, MasY);

for ( var i = 0 ; i < MasX.Length ; i++ )

{

X.Add(MasX[i]);

Y.Add(coeffish[0] \* Math.Exp(coeffish[1] \* MasX[i]));

}

chart1.Series[3].Points.DataBindXY(X, Y);

X.Clear();

Y.Clear();

coeffish.Clear();

coeffish = log(s, MasX, MasY);

for ( var i = 0 ; i < MasX.Length ; i++ )

{

X.Add(MasX[i]);

Y.Add(coeffish[0] \* Math.Log(MasX[i]) + coeffish[1]);

}

chart1.Series[4].Points.DataBindXY(X, Y);

X.Clear();

Y.Clear();

coeffish.Clear();

coeffish = hyperbolic(s,MasX, MasY);

for ( var i = 0 ; i < MasX.Length ; i++ )

{

X.Add(MasX[i]);

Y.Add(coeffish[0] / MasX[i] + coeffish[1]);

}

chart1.Series[5].Points.DataBindXY(X, Y);

}

**И суммы:**

public double sum(double[] x)

{

double summal = 0;

for ( int i = 0 ; i < x.Length ; i++ )

{

summal += x[i];

}

return summal;

}

public double sum2(double[] x)

{

double summal = 0;

for ( int i = 0 ; i < x.Length ; i++ )

{

summal += x[i] \* x[i];

}

return summal;

}

public double sum3(double[] x)

{

double summal = 0;

for ( int i = 0 ; i < x.Length ; i++ )

{

summal += x[i] \* x[i] \* x[i];

}

return summal;

}

public double sum4(double[] x)

{

double summal = 0;

for ( int i = 0 ; i < x.Length ; i++ )

{

summal += x[i] \* x[i] \* x[i] \* x[i];

}

return summal;

}

public double sum\_x\_mult\_y(double[] x, double[] y)

{

double summal = 0;

for ( int i = 0 ; i < x.Length ; i++ )

{

summal += x[i] \* y[i];

}

return summal;

}

public double sum\_x\_x\_mult\_y(double[] x, double[] y)

{

double summal = 0;

for ( int i = 0 ; i < x.Length ; i++ )

{

summal += x[i] \* x[i] \* y[i];

}

return summal;

}

public double sum\_mult2\_log\_x(double[] x)

{

double summal = 0;

for ( int i = 0 ; i < x.Length ; i++ )

{

summal += Math.Log(x[i]) \* Math.Log(x[i]);

}

return summal;

}

public double sum\_log\_x(double[] x)

{

double summal = 0;

for ( int i = 0 ; i < x.Length ; i++ )

{

summal += Math.Log(x[i]);

}

return summal;

}

public double sum\_y\_on\_log\_x(double[] x, double[] y)

{

double summal = 0;

for ( int i = 0 ; i < x.Length ; i++ )

{

summal +=(y[i] \* Math.Log(x[i]));

}

return summal;

}

public double sum\_dev\_1\_on\_x(double[] x)

{

double summal = 0;

for ( int i = 0 ; i < x.Length ; i++ )

{

summal += 1.0 /x[i];

}

return summal;

}

public double sum\_dev\_1\_on\_x\_x(double[] x)

{

double summal = 0;

for ( int i = 0 ; i < x.Length ; i++ )

{

summal += 1.0/ (x[i]\*x[i]);

}

return summal;

}

public double sum\_dev\_y\_on\_x(double[] x, double[] y)

{

double summal = 0;

for ( int i = 0 ; i < x.Length ; i++ )

{

summal += y[i]/ x[i] ;

}

return summal;

}