Analytical Geometry and Linear Algebra II

Joint Assignment 02

Teacher assistant: Eugene Marchuk
Done by: Erokhin Evgenii
Group: DSAI-03
Email: e.erokhin@innopolis.university

Innopolis
2023

https://github.com/EvgenyVO/AglaJoint-Assignment-02

Main for Task1:

```
int main(int argc,  char * argv[]) {
   int lengthOfDataSet;
   cin >> lengthOfDataSet;                    // read length of data set
   vector<double> tI;  //vector of t(i)
   vector<double> bI; //vector of b(i)
   for (int i=0;i<lengthOfDataSet;i++){  //fill vectors
      double temp1;
      cin>>temp1;
      tI.push_back(temp1);
      double temp2;
      cin>>temp2;
      bI.push_back(temp2);
   }
   int degree;
   cin >> degree;
   degree = degree+1;
   Matrix A(lengthOfDataSet,degree);
   for (int i=0;i<lengthOfDataSet;i++){
      for (int j=0;j<degree;j++){
         A.matrixNew[i][j] = pow(tI[i],j); //fill matrix A
      }
   }
   cout<<"A:\n";
   cout<<A;
   cout<<"A_T*A:\n";
   Matrix A_TA = A.transpose()*A; // A(transpose)*A
   cout<<A_TA;

   cout<<"(A_T*A)^-1:\n";  // (A(transpose)*A) Inverse
   SquareMatrix *res1 = (SquareMatrix *) &A_TA;
   SquareMatrix A_TAInverse = Inverse(*res1);
   cout<<A_TAInverse;

   cout<<"A_T*b:\n"; // A(transpose)*A * b
   Matrix b(lengthOfDataSet,1);
   for (int i=0;i<lengthOfDataSet;i++){
      b.matrixNew[i][0] = bI[i];
   }
   Matrix A_Tb = A.transpose()*b;
   cout<<A_Tb;

   cout<<"x~:\n";
   Matrix res2 = A_TAInverse;
   Matrix x = res2*A_Tb;
   cout<<x;
}
```

Main for Task1Representation:

```
int main(int argc,  char * argv[]) {
   FILE* pipe = _popen(GNUPLOT_NAME, "w");

   int lengthOfDataSet;
   cin >> lengthOfDataSet;   // read length of data set
   vector<double> tI;  //vector of t(i)
   vector<double> bI; //vector of b(i)
   for (int i=0;i<lengthOfDataSet;i++){  //fill vectors
      double temp1;
      cin>>temp1;
      tI.push_back(temp1);
      double temp2;
      cin>>temp2;
      bI.push_back(temp2);
   }
   int degree;
   cin >> degree;
   degree = degree+1;
   Matrix A(lengthOfDataSet,degree);
   for (int i=0;i<lengthOfDataSet;i++){
      for (int j=0;j<degree;j++){
         A.matrixNew[i][j] = pow(tI[i],j); //fill matrix A
      }
   }

   Matrix A_TA = A.transpose()*A; // A(transpose)*A

   // (A(transpose)*A) Inverse
   SquareMatrix *res1 = (SquareMatrix *) &A_TA;
   SquareMatrix A_TAInverse = Inverse(*res1);

   // A(transpose)*A * b
   Matrix b(lengthOfDataSet,1);
   for (int i=0;i<lengthOfDataSet;i++){
      b.matrixNew[i][0] = bI[i];
   }
   Matrix A_Tb = A.transpose()*b;

   Matrix res2 = A_TAInverse;
   Matrix x = res2*A_Tb;

   fprintf(pipe, "plot [-30 : 30] [-30 : 30] %lf*x**3 + %lf*x**2 + %lf*x**1 + %lf*x**0 , '-'
using 1:2 with points\n",
         x.matrixNew[3][0], x.matrixNew[2][0], x.matrixNew[1][0], x.matrixNew[0][0]);
   fprintf(pipe, "plot '-' w p ls 1\n");
   for (int i = 0; i < lengthOfDataSet; i++) {
      fprintf(pipe, "%f\t%f\n", tI[i], bI[i]);
   }
}
```

Table of dots

	1	2	3	4	5	6	7	8	9	10
ti	19	-11	-9	-6	10	-13	-7	11	-15	-17
bi	13	-18	5	5	-11	11	15	-12	2	-9

Representation