report

Saveliy Khlebnov CS-01 s.khlebnov@innopolis.university

April 2023

Main idea of this task to get formula from results of this function by least square approximation, so we will apply these formulas:

$A * x = b$;

$A^T A * x = A^T b$;

$x \approx (A^T A)^{-}1 A^T b$;

This will be done by given code:

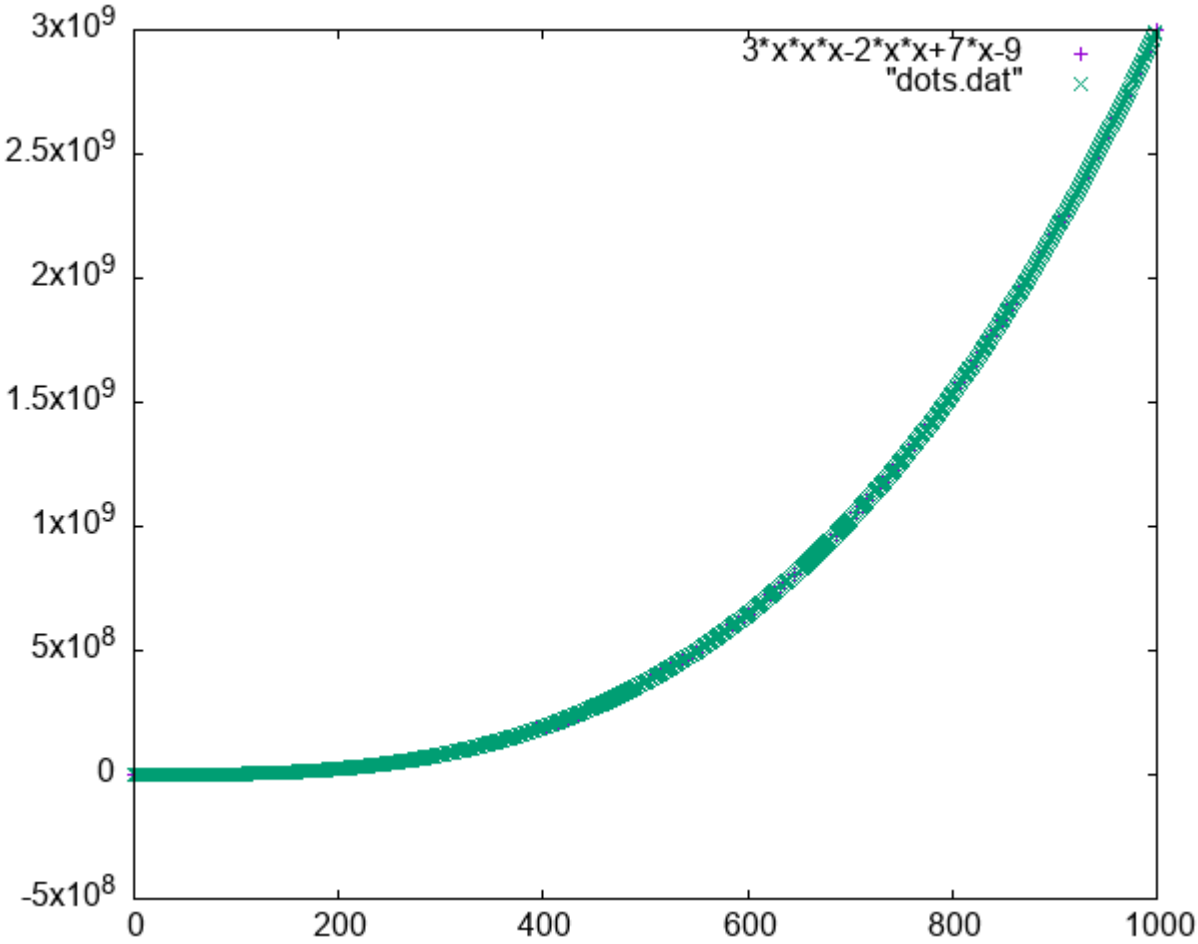
```
int main() {
    int m;
    std::cin >> m;

    ColumnVector<double> b{m};

    double t[m];
    for (int i = 0; i < m; i++) {
        std::cin >> t[i] >> b[i];
    }

    int degree;
    std::cin >> degree;
    Matrix<double> A{m, degree+1};
    for (int i = 0; i < m; ++i) {
        double temp = 1;
        for (int j = 0; j <= degree; j++) {
            A[i][j] = temp;
            temp *= t[i];
        }
    }

    std::cout << std::setprecision(4) << std::fixed;

    std::cout << "A:\n" << A;

    SquareMatrix<double> As{degree+1};
    As = A.transpose()*A;

    std::cout << "A_T*A:\n" << As;

    As = As.inverse();
    std::cout << "(A_T*A)^-1:\n" << As;

    b = A.transpose()*b;
    std::cout << "A_T*b:\n" << b;

    b = As*b;
    std::cout << "x~:\n" << b;
```

```
    return 0;
}
```

we can check correctness by applying this code for some dataset, it can be generated by next python code:

```
with open ("dots.dat", "w") as file:
for i in range (1000) :
    print (i , (3 * i**3 - 2 * i**2 + 7 * i - 9) + ( random () - 0.5) *10 , file= file )
```

after this we will get this result:
(A_T*A)^-1:
0.0159 -0.0001 0.0000 0.0000
-0.0001 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000
$A\_T * b$:
747838570721.8541
598004324926130.5000
498103992890072896.0000
426741988697711771648.0000
x :
-8.8687
7.0026
-2.0000
3.0000
graph of dots and original function by gnuplot:



link to source code `https://github.com/saveliyh/agla_assignment2.git`