

Calculating the determinant of a matrix by Gauss

Table of Contents

- [Algorithm](#)
- [Implementation](#)
- [Practice Problems](#)

Problem: Given a matrix A of size $N \times N$. Compute its determinant.

Algorithm

We use the ideas of [Gauss method for solving systems of linear equations](#)

We will perform the same steps as in the solution of systems of linear equations, excluding only the division of the current line to a_{ij} . These operations will not change the absolute value of the determinant of the matrix. When we exchange two lines of the matrix, however, the sign of the determinant can change.

After applying Gauss on the matrix, we receive a diagonal matrix, whose determinant is just the product of the elements on the diagonal. The sign, as previously mentioned, can be determined by the number of exchanged rows (if odd, then the sign of the determinant should be reversed). Thus, we can use the Gauss algorithm to compute the determinant of the matrix in complexity $O(N^3)$.

It should be noted that if at some point, we do not find non-zero cell in current column, the algorithm should stop and returns 0.


Implementation

```
const double EPS = 1E-9;
int n;
vector < vector<double> > a (n, vector<double>

double det = 1;
for (int i=0; i<n; ++i) {
    int k = i;
    for (int j=i+1; j<n; ++j)
        if (abs (a[j][i]) > abs (a[k][i]))
            k = j;
    if (abs (a[k][i]) < EPS) {
        det = 0;
        break;
    }
    swap (a[i], a[k]);
    if (i != k)
```

```
        det = -det;
    det *= a[i][i];
    for (int j=i+1; j<n; ++j)
        a[i][j] /= a[i][i];
    for (int j=0; j<n; ++j)
        if (j != i && abs (a[j][i]) > EPS)
            for (int k=i+1; k<n; ++k)
                a[j][k] -= a[i][k] * a[j][i];
}

cout << det;
```



Practice Problems

- [Codeforces - Wizards and Bets](#)

(c) 2014-2019 translation by <http://github.com/e-maxx-eng>

07:80/112