

Inversion

by lhy1024

email: admin@liudos.us

Complexity

According to *the introduction to the algorithm, page 827*, we can know that the inversion of the matrix through the LUP decomposition is $O(N^3)$ complexity.

Optimization

Taking into account the accuracy, think that if the difference is less than 10^{-9} is equal.

In order to avoid duplicate operations, the memory is specifically opened in the object to use and copy the matrix.

We first determine whether it is a triangular matrix, if not, then perform LUP decomposition and calculate the determinant value through the decomposed matrix to determine whether it is a singular matrix. Although the singular matrix is very few in the random test.

Request

Add `<xtl>` and `<xtensor>` to the include path.

Compiled

```
g++ -o test test.cpp -std=c++14
```

Run

Time

```
./test -t 1000
```

Randomly generate N matrices, compute their inverse matrix, and record time.
Break out if the calculation is wrong, skip if there is a singular matrix.

Only the time to compute the inverse matrix is recorded, and the time when the judgment is correct and the time when the random matrix is generated are not recorded

Run from file

```
./test -i in.txt -o out.txt
```

Input and output a test matrix from file.
e.g.

```
1 2 3 4 5
2 3 4 0 6
3 4 2 5 6 7
4 5 6 3 8
9 1 2 3 14
```

Evaluation

Environment

OS: Windows Subsystem for Linux
CPU: i7-4720HQ
Ram: 16.0 GB DDR3
SSD: 256GB

Result

Cycles	1,000	10,000	100,000	1,000,000
Time(s)	0.0625	0.6406	7.3438	78.2363