

Sep 25, 23 9:09

testLinearSystem.c

Page 1/3

```
// Copyright (c) Wenjun Ying, School of Mathematical Sciences and Institute
// of Natural Sciences, Shanghai Jiao Tong University, Shanghai 200240.

// This file is free software; as a special exception the author gives
// unlimited permission to copy and/or distribute it, with or without
// modifications, as long as this notice is preserved.

// This file is distributed in the hope that it will be useful, but
// WITHOUT ANY WARRANTY, to the extent permitted by law; without even the
// implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

#include <iostream>
#include <cstdlib>
#include <cstdio>
#include <cmath>

#include <limits.h>

const float EPSILON = 1.0E-6;

float computeMaxError(const float *x, const float *y, int n)
{
    float e = 0.0;
    for (unsigned int i = 0; i < n; i++) {
        float d = fabs(x[i] - y[i]);
        if (d > e) {
            e = d;
        }
    }
    return e;
}

void printVector(const char *info, const float *x, int n)
{
    std::cout << info << std::endl;
    for (unsigned int i = 0; i < n; i++) {
        std::cout << x[i] << " ";
    }
    std::cout << std::endl;
}

void printMatrix(const char *info, float **A, int n)
{
    for (unsigned int i = 0; i < n; i++) {
        for (unsigned int j = 0; j < n; j++) {
            if (fabs(A[i][j]) < EPSILON) {
                printf("    ");
            } else {
                printf("%7.3f", A[i][j]);
            }
        }
        std::cout << std::endl;
    }
    std::cout << std::endl;
}

int solveByGaussElimination(float **A, float *b,
                            int n, float *x)
{
    for (unsigned int k = 0; k < (n - 1); k++) {
        for (unsigned int i = k + 1; i < n; i++) {
            if (fabs(A[i][k]) < EPSILON) {
                return (-1);
            }
            float lik = A[i][k] / A[k][k];
            for (unsigned int j = k; j < n; j++) {
                A[i][j] = A[i][j] - lik * A[k][j];
            }
            b[i] = b[i] - lik * b[k];
        }
    }
}
```

Sep 25, 23 9:09

testLinearSystem.c

Page 2/3

```
}
}

printMatrix("A:", A, n);

for (int k = (n - 1); k >= 0; k--) {
    float sum = 0.0;
    for (unsigned int j = k + 1; j < n; j++) {
        sum += A[k][j] * x[j];
    }
    x[k] = (b[k] - sum) / A[k][k];
}

return 0;
}

float randfloat(void)
{
    float v = (rand()%INT_MAX) / (INT_MAX - 1.0);
    return v;
}

float randfloat(float a, float b)
{
    float v = a + (b - a) * randfloat();
    return v;
}

int main(int argc, char *argv[])
{
    int n = 0;
    if (argc > 1) {
        int n0 = atoi(argv[1]);
        if (n0 > 1) {
            n = n0;
        }
    }
    if (n <= 1) {
        std::cout << "Please input the matrix/vector dimension:" ;
        std::cin >> n;
        if (n <= 1) {
            std::cout << "invalid dimension found" << std::endl;
            exit(EXIT_FAILURE);
        }
    }

    float **A = new float*[n];
    for (unsigned int i = 0; i < n; i++) {
        A[i] = new float[n];
        for (unsigned int j = 0; j < n; j++) {
            A[i][j] = 0.0;
        }
    }

///////////////////////////////
srand(time(0));

for (unsigned int i = 0; i < n; i++) {
    for (unsigned int j = 0; j < n; j++) {
        A[i][j] = randfloat(-1.0, 1.0);
    }
}

printMatrix("A:", A, n);

float *z = new float[n];
for (unsigned int i = 0; i < n; i++) {
    z[i] = randfloat(-1.0, 1.0);
}
```

Sep 25, 23 9:09

testLinearSystem.c

Page 3/3

```
}
```

```
float *b = new float[n];
for (unsigned int i = 0; i < n; i++) {
    b[i] = 0.0;
    for (unsigned int j = 0; j < n; j++) {
        b[i] += A[i][j] * z[j];
    }
}

float *x = new float[n];
for (unsigned int i = 0; i < n; i++) {
    x[i] = 0.0;
}

int status = solveByGaussElimination(A, b, n, x);
if (status < 0) {
    std::cout << "Failed to solve the system" << std::endl;
    exit(EXIT_FAILURE);
}

printVector("x:", x, n);
printVector("z:", z, n);

float err = computeMaxError(x, z, n);
std::cout << "max error = " << err << std::endl;

// free pointers

for (unsigned int i = 0; i < n; i++) {
    delete[] A[i];
}
delete[] A;

delete[] b;
delete[] x;
delete[] z;

return EXIT_SUCCESS;
}
```