上机题第五题实验报告

王伟任　　计 33　　2013011333

一、题目要求及分析

第五章上机题 1： 用幂法求下列矩阵按模最大的特征值λ_1及其对应的特征向量$\boldsymbol{x_1}$，使$|(\lambda_1)_{k+1} - (\lambda_1)_k| < 10^{-5}$。

（1）$\boldsymbol{A}$ = [5, -4, 1; -4, 6, -4; 1, -4, 7]

（2）$\boldsymbol{B}$ = [25, -41, 10, -6; -41, 68, -17, 10; 10, -17, 5, -3; -6, 10, -3, 2]

二、实验结果及分析

对两个矩阵的求解结果分别为：

$\boldsymbol{A}$ 矩阵的最大特征值为λ_1 = 12.2543111057，

对应的特征向量$\boldsymbol{x_1}$ = [-0.6740214065, 1.0000000000, -0.8895570705]；

$\boldsymbol{B}$ 矩阵的最大特征值为λ_1 = 98.5216977084，

对应的特征向量$\boldsymbol{x_1}$ = [-0.6039723423, 1.0000000000, -0.2511351305, 0.1489534456]。

结果分别储存在 out1.txt 和 out2.txt 中。

三、实验代码

采用 C++语言实现。

```
#include <cstdio>
#include <cmath>

void multiply(int n, double** A, double* bl, double* ans){
    for (int i = 0; i < n; i++){
        ans[i] = 0;
        for (int j = 0; j < n; j++)
            ans[i] += A[i][j] * bl[j];
    }
}

double mmax(int n, double* v){
    double ans = 0;
    double nmax = 0;
    for (int i = 0; i < n; i++)
        if (fabs(v[i]) > nmax){
            ans = v[i];
            nmax = fabs(v[i]);
        }
    return ans;
}

int main(int argc, char** argv){
```

```
    FILE* fp = fopen(argv[1], "r");
    int n;
    fscanf(fp, "%d", &n);
    fgetc(fp);
    double** array = new double*[n];
    for (int i = 0; i < n; i++){
        array[i] = new double[n];
        for (int j = 0; j < n; j++){
            fscanf(fp, "%lf", &array[i][j]);
            fgetc(fp);
        }
    }
    fclose(fp);

    double* v = new double[n];
    double* u = new double[n];
    v[0] = u[0] = 1.0;
    for (int i = 1; i < n; i++)
        v[i] = u[i] = 0.0;
    double l1, l2;
    l1 = 0;
    do{
        l2 = l1;
        multiply(n, array, u, v);
        l1 = mmax(n, v);
        for (int i = 0; i < n; i++)
            u[i] = v[i] / l1;
    } while(fabs(l1 - l2) >= 0.00001);

    fp = fopen(argv[2], "w");
    fprintf(fp, "l = %.10f\n", l1);
    fprintf(fp, "x = [");
    for (int i = 0; i < n - 1; i++)
        fprintf(fp, "%.10f\t", u[i]);
    fprintf(fp, "%.10f]\n", u[n - 1]);

    fclose(fp);
    for (int i = 0; i < n; i++)
        delete[] array[i];
    delete[] array;
    delete[] u;
    delete[] v;
    return 0;
}
```