上机题第六题实验报告

王伟任　　计 33　　2013011333

一、题目要求及分析

第六章上机题 3： 对物理实验中所得数据分别用公式$y = a + bt + ct^2$和指数函数$y = ae^{bt}$作曲线拟合，比较上述两条拟合曲线哪条更好。

由于用法方程法作曲线拟合只能用多项式函数拟合，不能直接用指数函数拟合，因此需要将指数函数做一次变型将其转化为多项式函数，方法是两边取自然对数，变为$\ln y = \ln a + bt$的多项式形式。

二、实验结果及分析

用公式$y = a + bt + ct^2$进行拟合的结果为：

a = -45.2942307692; b = 94.1942921784; c = -6.1268261151。

误差的 2-范数为 22.0137732933。

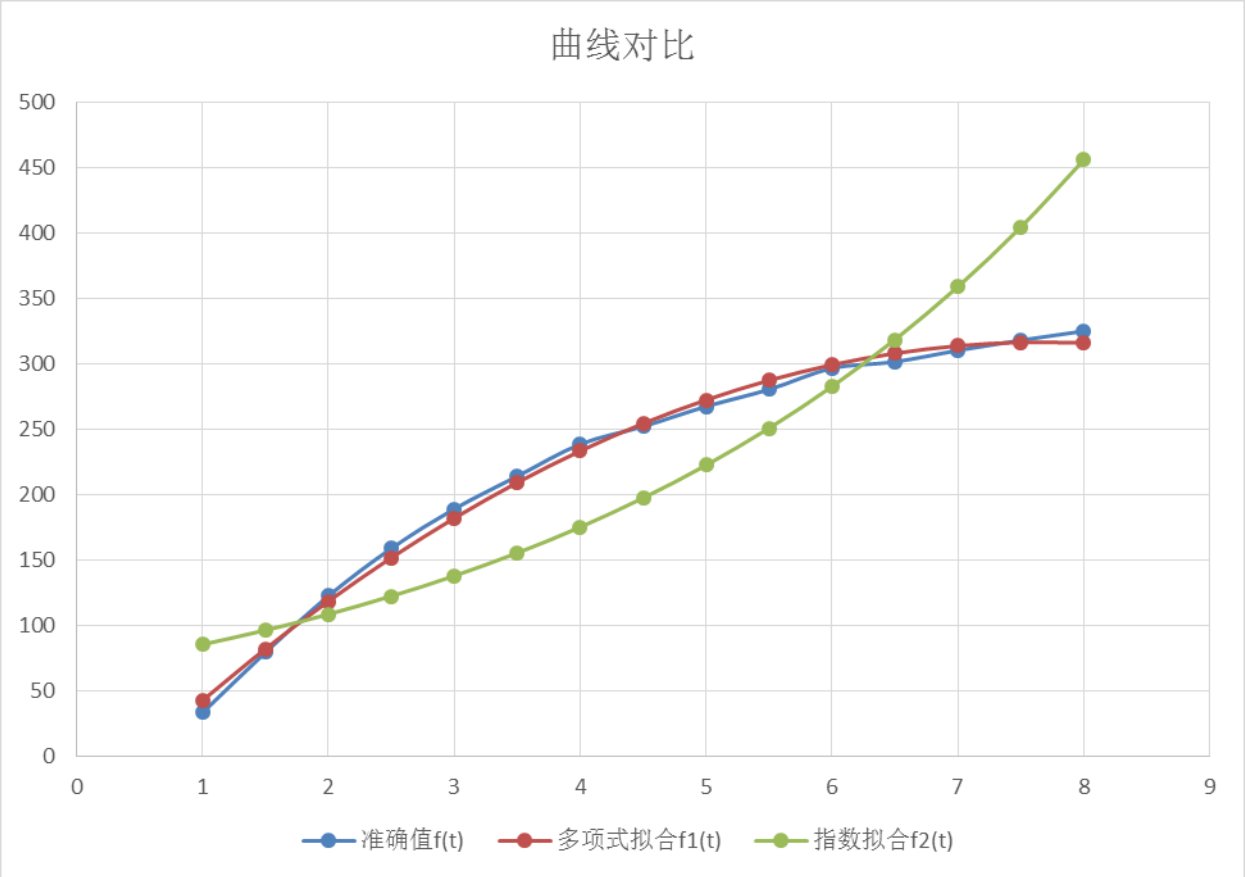
结果保存在 out1.txt 中。

用指数函数$y = ae^{bt}$进行拟合的结果为：

a = 67.3937925785; b = 0.2389834379。

误差的 2-范数为 218.9097098003。

结果保存在 out2.txt 中。



将表格函数，多项式函数，指数函数三条曲线绘制在同一张图中，可以很明显看出，用多项式函数作曲线拟合的效果更好，与表格函数的差距更小一些。用指数函数拟合效果较差，而且虽然指数函数拟合的曲线也是单调递增的，但是斜率却是增大的趋势，而原函数的斜率趋势是减小的。

三、实验代码

采用 C++语言实现。

用公式$y = a + bt + ct^2$进行拟合：

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

int n = 15;
int w = 3;

double** create(double* t){
    double** ans = new double*[n];
    for (int i = 0; i < n; i++){
        ans[i] = new double[w];
        for (int j = 0; j < w; j++){
            ans[i][j] = 1.0;
            for (int kk = 0; kk < j; kk++)
                ans[i][j] *= t[i];
        }
    }
    return ans;
}

double* multiply1(double** A, double* f){
    double* ans = new double[w];
    for (int i = 0; i < w; i++){
        ans[i] = 0;
        for (int j = 0; j < n; j++)
            ans[i] += A[j][i] * f[j];
    }
    return ans;
}

double** multiply2(double** A){
    double** ans = new double*[w];
    for (int i = 0; i < w; i++){
        ans[i] = new double[w];
        for (int j = 0; j < w; j++){
            ans[i][j] = 0.0;
            for (int k = 0; k < n; k++)
```

```
                    ans[i][j] += A[k][i] * A[k][j];
            }
        }
        return ans;
}

double** Cholesky(double** G){
        double** L = new double*[w];
        for (int i = 0; i < w; i++)
            L[i] = new double[w];
        for (int i = 0; i < w; i++)
            for (int j = i + 1; j < w; j++)
                L[i][j] = 0;

        L[0][0] = sqrt(G[0][0]);
        for (int i = 1; i < w; i++)
            L[i][0] = G[i][0] / L[0][0];
        for (int j = 1; j < w; j++){
            double ss = 0.0;
            for (int i = 0; i < j; i++)
                ss += L[j][i] * L[j][i];
            L[j][j] = sqrt(G[j][j] - ss);
            for (int i = j + 1; i < w; i++){
                double s = 0.0;
                for (int ii = 0; ii < j; ii++)
                    s += L[i][ii] * L[j][ii];
                L[i][j] = (G[i][j] - s) / L[j][j];
            }
        }
        return L;
}

double* solve(double** L, double* b){
        double* y = new double[w];
        for (int i = 0; i < w; i++){
            y[i] = b[i] / L[i][i];
            for (int j = 0; j < i; j++)
                y[i] = y[i] - y[j] * L[i][j] / L[i][i];
        }
        return y;
}

double* solve2(double** L, double* y){
        double* x = new double[w];
```

```
    for (int i = w - 1; i >= 0; i--){
        x[i] = y[i] / L[i][i];
        for (int j = i + 1; j < w; j++)
            x[i] = x[i] - x[j] * L[j][i] / L[i][i];
    }
    return x;
}


int main(){
    FILE* fp = fopen("test.txt", "r");
    double* t = new double[n];
    double* f = new double[n];
    for (int i = 0; i < n; i++)
        fscanf(fp, "%lf", &t[i]), fgetc(fp);
    for (int i = 0; i < n; i++)
        fscanf(fp, "%lf", &f[i]), fgetc(fp);
    fclose(fp);

    double** A = create(t);
    double* b = multiply1(A, f);
    double** G = multiply2(A);
    double** L = Cholesky(G);
    double* y = solve(L, b);
    double* x = solve2(L, y);

    fp = fopen("out1.txt", "w");
    for (int i = 0; i < w; i++){
        if (i == 0) fprintf(fp, "a = ");
        if (i == 1) fprintf(fp, "b = ");
        if (i == 2) fprintf(fp, "c = ");
        fprintf(fp, "%.10f;\t", x[i]);
    }
    fprintf(fp, "\n");
    fprintf(fp, "wu cha 2-norm = ");
    double* delta = new double[n];
    for (int i = 0; i < n; i++){
        delta[i] = 0;
        for (int j = 0; j < w; j++)
            delta[i] += A[i][j] * x[j];
    }
    double norm = 0;
    for (int i = 0; i < n; i++)
        norm += (delta[i] - f[i]) * (delta[i] - f[i]);
```

```
    norm = sqrt(norm);
    fprintf(fp, "%.10f\n", norm);
    fclose(fp);

    delete[] t;
    delete[] f;
    delete[] b;
    delete[] y;
    delete[] x;
    delete[] delta;
    for (int i = 0; i < w; i++)
        delete[] G[i], delete[] L[i];
    delete[] G;
    delete[] L;
    for (int i = 0; i < n; i++)
        delete[] A[i];
    delete[] A;

    return 0;
}
```

用指数函数$y = ae^{bt}$进行拟合：

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

int n = 15;
int w = 2;

double** create(double* t){
    double** ans = new double*[n];
    for (int i = 0; i < n; i++){
        ans[i] = new double[w];
        for (int j = 0; j < w; j++){
            ans[i][j] = 1.0;
            for (int kk = 0; kk < j; kk++)
                ans[i][j] *= t[i];
        }
    }
    return ans;
}

double* multiply1(double** A, double* f){
    double* ans = new double[w];
    for (int i = 0; i < w; i++){
```

```
            ans[i] = 0;
            for (int j = 0; j < n; j++)
                ans[i] += A[j][i] * f[j];
        }
        return ans;
}

double** multiply2(double** A){
        double** ans = new double*[w];
        for (int i = 0; i < w; i++){
            ans[i] = new double[w];
            for (int j = 0; j < w; j++){
                ans[i][j] = 0.0;
                for (int k = 0; k < n; k++)
                    ans[i][j] += A[k][i] * A[k][j];
            }
        }
        return ans;
}

double** Cholesky(double** G){
        double** L = new double*[w];
        for (int i = 0; i < w; i++)
            L[i] = new double[w];
        for (int i = 0; i < w; i++)
            for (int j = i + 1; j < w; j++)
                L[i][j] = 0;

        L[0][0] = sqrt(G[0][0]);
        for (int i = 1; i < w; i++)
            L[i][0] = G[i][0] / L[0][0];
        for (int j = 1; j < w; j++){
            double ss = 0.0;
            for (int i = 0; i < j; i++)
                ss += L[j][i] * L[j][i];
            L[j][j] = sqrt(G[j][j] - ss);
            for (int i = j + 1; i < w; i++){
                double s = 0.0;
                for (int ii = 0; ii < j; ii++)
                    s += L[i][ii] * L[j][ii];
                L[i][j] = (G[i][j] - s) / L[j][j];
            }
        }
        return L;
```

```
}

double* solve(double** L, double* b){
	double* y = new double[w];
	for (int i = 0; i < w; i++){
		y[i] = b[i] / L[i][i];
		for (int j = 0; j < i; j++)
			y[i] = y[i] - y[j] * L[i][j] / L[i][i];
	}
	return y;
}

double* solve2(double** L, double* y){
	double* x = new double[w];
	for (int i = w - 1; i >= 0; i--){
		x[i] = y[i] / L[i][i];
		for (int j = i + 1; j < w; j++)
			x[i] = x[i] - x[j] * L[j][i] / L[i][i];
	}
	return x;
}


int main(){
	FILE* fp = fopen("test.txt", "r");
	double* t = new double[n];
	double* f = new double[n];
	for (int i = 0; i < n; i++){
		fscanf(fp, "%lf", &t[i]);
		fgetc(fp);
	}
	for (int i = 0; i < n; i++){
		fscanf(fp, "%lf", &f[i]);
		fgetc(fp);
		f[i] = log(f[i]);
	}
	fclose(fp);

	double** A = create(t);
	double* b = multiply1(A, f);
	double** G = multiply2(A);
	double** L = Cholesky(G);
	double* y = solve(L, b);
	double* x = solve2(L, y);
```

```
    fp = fopen("out2.txt", "w");
    fprintf(fp, "a = %.10f;\t", exp(x[0]));
    fprintf(fp, "b = %.10f;\n", x[1]);
    fprintf(fp, "wu cha 2-norm = ");
    double* delta = new double[n];
    for (int i = 0; i < n; i++){
        delta[i] = 0;
        for (int j = 0; j < w; j++)
            delta[i] += A[i][j] * x[j];
        delta[i] = exp(delta[i]);
    }
    double norm = 0;
    for (int i = 0; i < n; i++){
        f[i] = exp(f[i]);
        norm += (delta[i] - f[i]) * (delta[i] - f[i]);
    }
    norm = sqrt(norm);
    fprintf(fp, "%.10f\n", norm);
    fclose(fp);

    delete[] t;
    delete[] f;
    delete[] b;
    delete[] y;
    delete[] x;
    delete[] delta;
    for (int i = 0; i < w; i++)
        delete[] G[i], delete[] L[i];
    delete[] G;
    delete[] L;
    for (int i = 0; i < n; i++)
        delete[] A[i];
    delete[] A;

    return 0;
}
```