A SOFTWARE FOR SMOOTHING MAGNET TRACK IN PARTICLE ACCELERATOR*

Q. Zhang†, W. Wang, X.Y. He, G.W. Liu
University of Science and Technology of China, Hefei, China

Abstract

This article describes a software for smoothing magnet track in particle accelerator. This paper introduces the development process of the software from the aspects of interface design, algorithm analysis, parameter meaning and so on. Magnet track smoothing means that under the conditions of meeting absolute accuracy, if the relative position error of the adjacent magnet is too large, it will cause the loss of beam, we call the track curve is not smooth enough. Smooth analysis can find these magnet components, the curve is smooth after a reasonable adjustment. The software is based on the least square method. The software is tested by using the data of HLS storage ring, the results meet the needs of the work.

INTRODUCTION

From June of 2012, Hefei Light Source（HLS）had a major renovation, it is named Hefei light source-II (HLSII). The HLSII is a new state-of-the-art, low-energy electron storage ring (800 MeV) that delivers world-leading intensity and brightness beam with a 40 nm-rad minimum horizontal emittance. The HLS-II includes a 73.435 meters linac accelerator and a 66.13 meters storage ring [1].

During the renovation, the smoothing technology was used. Magnet track smoothing means that under the conditions of meeting absolute accuracy, if the relative position error of the adjacent magnet is too large, it will cause the loss of beam, we call the track curve is not smooth enough. Smooth analysis can find these magnet components, the curve is smooth after a reasonable adjustment [2]. Relying on artificial smooth analysis is less efficient, therefore, the alignment group of National Synchrotron Radiation Laboratory (NSRL) decided to develop the smoothing software. By comparing the advantages of a variety of smoothing methods, we used the least squares method as the core algorithm of the software. We used java programming language because it was a cross-platform, object-oriented, simple language and the advantage of java is that it can break down complex tasks into a simple subtask.

METHODOLOGY

Interface

The interface is shown as Figure 1, it was designed by the JAVA SWT plug-in. The interface includes three modules: data input, parameter setting and calculation. Program running process was shown in Figure 2, the user selects the data file to be processed and sets the parameters according to the accuracy requirements, then, finishes the smoothing by clicking the ‘calculate’ button.

There are two kinds of smoothing methods and they have same algorithm that is the least squares method, the difference lies in the sliding window selection mode and the moving step (described in detail below), the user selects the appropriate method according to the shape of the magnet track. The user sets the parameters according to the accuracy requirements. The meaning of the window length is processed the account of data every time. The step size is the length of the window that moves each time. The process iterates by sliding the window.

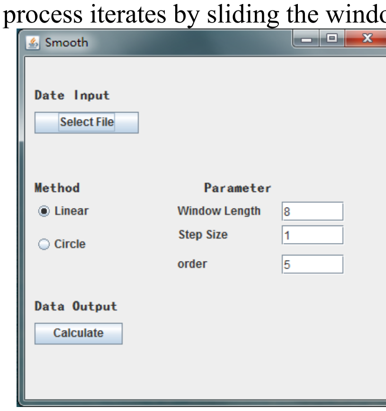


Figure 1: GUI.

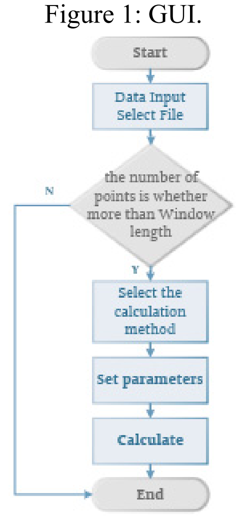


Figure 2: Smoothing Flowchart.

To simplify the file read program, the software currently only supports txt file, the data format is shown in Figure 3. The magnet coordinate data includes the point name, X coordinate value, Y coordinate value, Z coordinate value and two values are tab-separated. After the program completes processing, two txt files are generated in the path where the original data is located: The adjust-

†email address: nsrlzq@mail.ustc.edu.cn

ed magnet coordinate value file and the adjustment amount file required for each point, the data format is the same as the original data format.

POINTNAME	X	Y	Z
SP02-1-P0	0	0.02	0.03
SP02-2-P0	317	0.31	1.23
SP02-3-P0	750	0.05	4.32
SP02-4-P0	1038	0.44	5.29
SP03-1-P0	3489	-0.14	12.38
SP03-2-P0	3776	-0.11	15.70
SP03-3-P0	4209	0.09	20.83
SP03-4-P0	4526	0	24.67

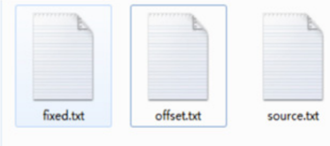


Figure 3: The file of input and output.

Development Process

The advantage of java is that it can break down complex tasks into a simple subtask. This program has three packages and several classes, each module has a different function.

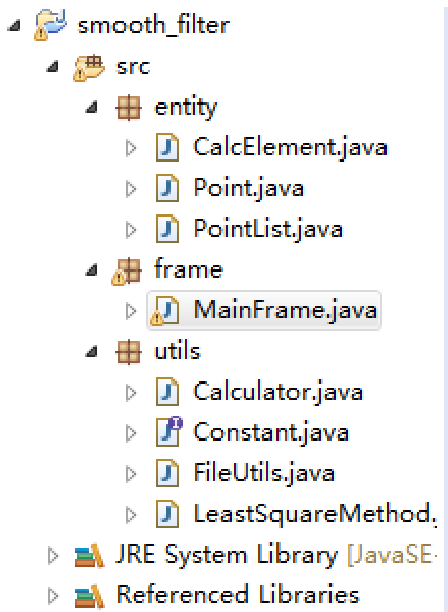


Figure 4: The compositions of Program.

The entity package is mainly used to store data. The utils package includes the least squares method, the file input and output algorithms, and the constants used in the program. The frame package uses the SWT plug-in to design a graphical interface that is the entrance to the entire smoothing program.

The core of the program is the least squares method. The mathematical expression is:

$$P_n(x) = \sum_{k=0}^{n} a_k x^k \in \Phi$$

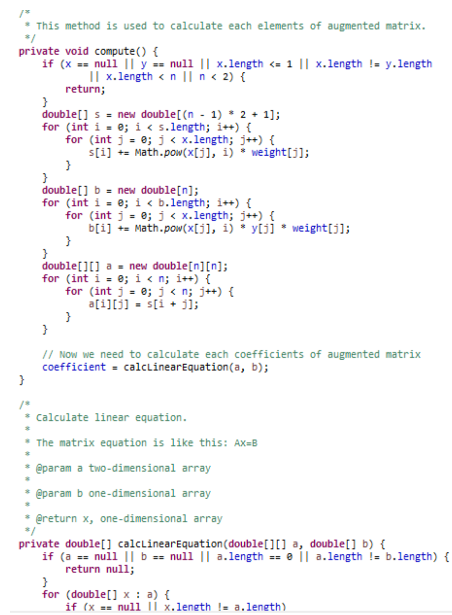
Calculate:

$$I = \sum_{I=0}^{m} [P_n(x_i - y_i)]^2 = \min$$

The problem can be solved by solving the solution of the system of linear equations: $Ga = B$,

$$\begin{bmatrix} m+1 & \sum_{i=0}^{m} x_i & \cdots & \sum_{i=0}^{m} x_i^n \\ \sum_{i=0}^{m} x_i & \sum_{i=0}^{m} x_i^2 & \cdots & \sum_{i=0}^{m} x_i^{n+1} \\ \vdots & \vdots & & \vdots \\ \sum_{i=0}^{m} x_i^n & \sum_{i=0}^{m} x_i^{n+1} & \cdots & \sum_{i=0}^{m} x_i^{2n} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{m} y_i \\ \sum_{i=0}^{m} x_i y_i \\ \vdots \\ \sum_{i=0}^{m} x_i^n y_i \end{bmatrix}$$

From the programming point of view need to do two things: Solving the coefficients of the system of linear equations and Solving Linear Equations. The code is shown in Figure 5 (The part of code).



```
/*
 * This method is used to calculate each elements of augmented matrix.
 */
private void compute() {
    if (x == null || y == null || x.length <= 1 || x.length != y.length
            || x.length < n || n < 2) {
        return;
    }
    double[] s = new double[(n - 1) * 2 + 1];
    for (int i = 0; i < s.length; i++) {
        for (int j = 0; j < x.length; j++) {
            s[i] += Math.pow(x[j], i) * weight[j];
        }
    }
    double[] b = new double[n];
    for (int i = 0; i < b.length; i++) {
        for (int j = 0; j < x.length; j++) {
            b[i] += Math.pow(x[j], i) * y[j] * weight[j];
        }
    }
    double[][] a = new double[n][n];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            a[i][j] = s[i + j];
        }
    }

    // Now we need to calculate each coefficients of augmented matrix
    coefficient = calcLinearEquation(a, b);
}

/*
 * Calculate linear equation.
 *
 * The matrix equation is like this: Ax=B
 *
 * @param a two-dimensional array
 *
 * @param b one-dimensional array
 *
 * @return x, one-dimensional array
 */
private double[] calcLinearEquation(double[][] a, double[] b) {
    if (a == null || b == null || a.length == 0 || a.length != b.length) {
        return null;
    }
    for (double[] x : a) {
        if (x == null || x.length != a.length)
```

Figure 5: The code of least squares method.

Due to the large number of magnets, it is not possible to smooth all the magnets at once. The team of European Organization for Nuclear Research (CERN) developed a smoothing data processing software PLANE, only the magnets within the window were smoothed, and then through the movement of the window to ensure the continuity of smooth analysis [2]. According to the mode of data processing, the program developed two smooth methods. Linear smoothing method: Within this window all the points except the one in the window centre are fitted to a given order polynomial. The centre point is 'rejected' if its deviation from the polynomial exceeds a specified tolerance. The process iterates by sliding the window point-by-point [3]. Circle smoothing method: Within this window all the points are fitted to a given order polynomial, the points behind centre point are adjusted if their deviation from the polynomial exceeds a specified tolerance. The process is iterated by moving half the length of the window.

ISBN 978-3-95450-182-3

```
public static void linear(PointList pointList, int order, String type) {
    int segment = 7;
    int step = 1;
    int length = segment - 1;

    List<Point> points = pointList.getPoints();
    int range = points.size() - segment + 1;
    for (int i = 0; i < range; i += step) {
        double[] horizontal = new double[length];
        double[] vertical = new double[length];
        for (int j = 0; j < length; j++) {
            int anchor = i + j;
            if (j > 2) {
                anchor++;
            }
            Point currentPoint = pointList.getPoints().get(anchor);
            horizontal[j] = currentPoint.getZ();
            if (type.equals(Constant.TYPE_ZX)) {
                vertical[j] = currentPoint.getX();
            } else {
                vertical[j] = currentPoint.getY();
            }
        }
        LeastSquareMethod method = new LeastSquareMethod(horizontal, vertical, order);
        Point targetPoint = points.get(i + 3);
        if (type.equals(Constant.TYPE_ZX)) {
            targetPoint.setX(fixValue(method, targetPoint));
        } else {
            targetPoint.setY(fixValue(method, targetPoint));
        }
    }
}
```

Figure 6: The code of linear smoothing method.

According to the accuracy requirements, the user sets the order of the polynomial fitting equation. If the order is less than or equal to 1, the program will report an error. The order is set according to the amount of data, High order does not necessarily produce high-accuracy results, but it will inevitably lead to a decline in running speed. According to the test results, we generally choose the fifth order fitting equation.

CONCLUSION

In this paper, we introduced a smoothing software. The purpose of smoothing is to make the magnet track curve smoother to reduce beam loss and increase beam life. The article details the software development process and the core algorithm. The software can smooth different style accelerators by selecting different parameters. Depending on the accuracy requirements, the user can set the appropriate window length, step size and equation order. After the program debugging, smoothing software meets the basic needs of the work by testing the data of HLS storage ring. But this software still has many shortcomings. In the future, we will expand its functions and make it more user-friendly. Smoothing software will no longer be a specific data processing software, but also with data storage and query functions.

REFERENCES

[1] W. WANG and X. Y. He, "The surveying data processing of control network based on HLS upgrade", in *Proc. 4st Int. Particle Accelerator Conf. (IPAC'13)*, Shanghai, China, May 2013, paper WEPME025, pp. 2986-2988.

[2] F. Barry and B. Tom, "Surveying principles and application Ninth edition", Springer, New York, American, 2011, pp. 121-124.

[3] D. Missiaen *et al.*, "The alignment of the LHC", in *Proc. 2st Int. Particle Accelerator Conf. (IPAC'09)*, Vancouver, Canada, May 2009, paper WE4GRC04, pp. 1973-1975.