|  |
| --- |
|  |
| SeesharpTools.JXI.  Mathematics.Curvefitting |
| user manual |
|  |
|  |
|  |

|  |
| --- |
|  |

**文档版本**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 修订说明 | 修订人 |
| 2017/08/29 | 0.1 | 初版，SeesharpTool.JXI.Mathematics.Curvefitting | **谢晓姣** |
|  |  |  |  |

**目录**

[1. CurveFitting 1](#_Toc9700)

[1.1. CurveFitting介绍 1](#_Toc11840)

[1.2. 线性拟合 1](#_Toc6899)

[1.2.1. 拟合原型 1](#_Toc21486)

[1.2.2. 拟合函数 1](#_Toc14962)

[1.2.3. 范例 1](#_Toc26235)

[1.3. 指数拟合 3](#_Toc18389)

[1.3.1. 拟合原型 3](#_Toc9940)

[1.3.2. 拟合函数 3](#_Toc9822)

[1.3.3. 范例 3](#_Toc8476)

[1.4. 多项式拟合 5](#_Toc6145)

[1.4.1. 拟合原型 5](#_Toc17511)

[1.4.2. 拟合函数 5](#_Toc30890)

[1.4.3. 范例 5](#_Toc10606)

## 

# CurveFitting

## CurveFitting介绍

CurveFitting包括线性拟合、指数拟合、多项式拟合。

## 线性拟合

### 拟合原型

线性拟合的拟合原型如下：



### 拟合函数

线性拟合类的多态方法：

void LinearFit(double[] x, double[] y, double interestX, out double fittedY)

void LinearFit(double[] x, double[] y, double interestX, out double fittedY, out double slope, out double intercept)

void LinearFit(double[] x, double[] y, double[] interestX, ref double[] fittedY)

void LinearFit(double[] x, double[] y, double[] interestX, ref double[] fittedY, out double slope, out double intercept)

其中，x为输入数据的横坐标，y为输入数据的纵坐标，interestX为用户感兴趣的x值，fittedY为interestX对应的拟合值，slope为斜率，intercept为截距。

**注意：x，y的数组长度必须一致，interestX，fittedY的数组长度必须一致!**

### 范例

double[] x = { 1, 2, 3, 4, 5 };

double[] y = { 3, 5, 7, 9, 11 };

double slop = 0;

double interception = 0;

double[] data = new double[5];

EasyCurveFitting.LinearFit(x, y, x, ref data, out slop, out interception);

结果：

data={3, 5, 7, 9, 11 }；

slop=2;

Interception=1;

## 指数拟合

### 拟合原型

指数拟合的拟合原型如下：



### 拟合函数

指数拟合类的多态方法：

void ExponentialFit(double[] x, double[] y, double interestX, out double fittedY)

void ExponentialFit(double[] x, double[] y, double interestX, out double fittedY, out double amplitude, out double damping, out double offset)

void ExponentialFit(double[] x, double[] y, double[] interestX, ref double[] fittedY)

void ExponentialFit(double[] x, double[] y, double[] interestX, ref double[] fittedY, out double amplitude, out double damping, out double offset)

其中，x为输入数据的横坐标，y为输入数据的纵坐标，interestX为用户感兴趣的x值，fittedY为interestX对应的拟合值，amplitude为幅值，damping为衰减，offset为偏移量（目前为0，即没有偏移量）。

**注意：x，y的数组长度必须一致，interestX，fittedY的数组长度必须一致!**

### 范例

double[] x = { 1, 2, 3, 4, 5 };

double[] y = new double[5];

for (int i = 0; i < 5; i++)

{

y[i] = System.Math.Exp(1.7 \* x[i]) \* (2.6);

}

double amplitude = 0;

double damping = 0;

double offset = 0;

double[] data = new double[5];

EasyCurveFitting.ExponentialFit(x, y, x, ref data, out amplitude, out damping, out offset);

结果：

data={14.232263218490726, 77.906660123232257, 426.45695897974457, 2334.4029582910866, 12778.398984777754 }；

amplitude=2.600000000000001;

damping=1.7;

Offset=0；

## 多项式拟合

### 拟合原型

多项式拟合的拟合原型如下：



### 拟合函数

多项式拟合类的多态方法：

void PolynomialFit(double[] x, double[] y, int order, double interestX, out double fittedY)

void PolynomialFit(double[] x, double[] y, int order, double interestX, out double fittedY, ref double[] coefficients)

void PolynomialFit(double[] x, double[] y, int order, double[] interestX, ref double[] fittedY)

void PolynomialFit(double[] x, double[] y, int order, double[] interestX, ref double[] fittedY, ref double[] coefficients)

其中，x为输入数据的横坐标，y为输入数据的纵坐标，order为拟合阶数，interestX为用户感兴趣的x值，fittedY为interestX对应的拟合值，coefficients为多项式拟合系数。

**注意：x，y的数组长度必须一致，interestX，fittedY的数组长度必须一致!**

### 范例

double[] x = { 1, 2, 3, 4, 5 };

double[] y = new double[5];

for (int i = 0; i < 5; i++)

{

y[i] = 10 + -10 \* x[i] -5 \* System.Math.Pow(x[i], 2) -4 \* System.Math.Pow(x[i], 3);

}

double[] coe = new double[5];

double[] data = new double[5];

int order = 3;

EasyCurveFitting.PolynomialFit(x, y, order, x, ref data, ref coe);

结果：

data={-9.0000000000094, -61.999999999993193, -172.99999999999784, -5.00000000002835, -664.99999999999807}；

coe={9.99999999993561，-9.9999999999196483，-5.00000000002835，-3.9999999999970135}；