# Development Document

LinearInterpolation function

线性插值函数，使用两组（每组两个，共四个）迭代器进行求线性插值。

**Syntax**

double LinearInterpolation(double d,

VecDouIt it1Beg,

VecDouIt it1End,

VecDouIt it2Beg,

VecDouIt it2Edn

)；

**参数：**

*d*

类型：double

插值公式中的已知数值。

*it1Beg*

类型：VecDouit。

与d同种的容器的起始迭代器。

*it1End*

类型：VecDouit

与d同种的容器的结束迭代器。

*it2Beg*

类型：VecDouit

与所求值同种的容器的起始迭代器。

*it2End*

类型：VecDouit

与所求值同种的容器的结束迭代器。

**返回值：**

类型：double

如果函数函数执行成功，返回要求的值。

如果函数返回-1，表示d小于it1所指向的容器中的最小值。

如果函数返回-2，表示d大于it2所指向的容器中的最大值。

**附注：**

第一个容器应为递增容器，两个容器所含元素的个数必须相等。

WaterBlance function

水量平衡方程所对应的函数，通过已知q或V中的一个来求另一个。

△t\*(Q1+Q2)/(2\*10000) - △t\*(q1+q2)/(2\*10000) = V2 – V1;

**Syntax**

double WaterBalance(

double para,

double q1,

double V1,

VecDouIt ItQ,

int i

);

**参数:**

*Para*

类型：double

已知的q或V。

*q1*

类型：double

公式中的q1。

*V1*

类型：double

公式中的V1。

*ItQ*

类型：VecDouit

公式中Q的迭代器

*i*

类型：int

如果i的值为0，则为已知V求q。

若果i的值为1，则为已知q求V。

**返回值：**

如果i的值为0，返回所求q。

如果i的值为1，返回所求V。

**附注：**

无。

Discharge function

根据已知库容，由下泄流量公式求得下泄流量。

Syntax

Discharge(

double V,

int i

)；

参数:

*V*

类型：double

已知库容。

*I*

类型：int

如果i的值为0，为设计情况。

如果i的值为1，为校核情况。

返回值：

如果i的值为0，返回设计情况下库容对应的下泄流量。

如果i的值为1，返回校核情况下库容对应的下泄流量。

附注：

无。