

# 一、迭代法

迭代法是用于求方程或方程组近似根的一种常用的算法设计方法。设方程为  $f(x)=0$ ，用某种数学方法导出等价的形式  $x=g(x)$ ，然后按以下步骤执行：

- (1) 选一个方程的近似根，赋给变量  $x_0$ ；
- (2) 将  $x_0$  的值保存于变量  $x_1$ ，然后计算  $g(x_1)$ ，并将结果存于变量  $x_0$ ；
- (3) 当  $x_0$  与  $x_1$  的差的绝对值还小于指定的精度要求时，重复步骤 (2) 的计算。

若方程有根，并且用上述方法计算出来的近似根序列收敛，则按上述方法求得的  $x_0$  就认为是方程的根。上述算法用 C 程序的形式表示为：

**【算法】** 迭代法求方程的根

```
{ x0=初始近似根;
do {
x1=x0;
x0=g(x1); /*按特定的方程计算新的近似根*/
} while ( fabs(x0-x1)>Epsilon);
printf(“方程的近似根是%f\n”, x0);
}
```

迭代算法也常用于求方程组的根，令

$X = (x_0, x_1, \dots, x_{n-1})$

设方程组为：

$x_i = g_i(X) \ (i=0, 1, \dots, n-1)$

则求方程组根的迭代算法可描述如下：

**【算法】** 迭代法求方程组的根

```
{ for (i=0;i<n;i++) x[i]=初始近似根;
do {
for (i=0;i<n;i++) y[i]=x[i];
for (i=0;i<n;i++) x[i]=gi(X);
for (delta=0.0,i=0;i<n;i++) if (fabs(y[i]-x[i])>delta) delta=fabs(y[i]-x[i]);
} while (delta>Epsilon);
for (i=0;i<n;i++) printf(“变量 x[%d]的近似根是 %f”, I, x[i]);
printf(“\n” );
}
```

具体使用迭代法求根时应注意以下两种可能发生的情况：

- (1) 如果方程无解，算法求出的近似根序列就不会收敛，迭代过程会变成死循环，因此在使用迭代算法前应先考察方程是否有解，并在程序中对迭代的次数给予限制；
- (2) 方程虽然有解，但迭代公式选择不当，或迭代的初始近似根选择不合理，也会导致迭代失败。