

# Lecture notes: Difference Engine

## 课前问题

Q: 如何使用并行计算来优化? 以  $x^7+x$  为例。

A: 进行七次差分后可得到常数, 然后六七级差分、四五级差分、二三级差分、第一级差分 and  $f(1)$  同时进行并行计算, 运算完后将左边的数替换为结果, 然后五六级差分、三四级差分、一二级差分同时进行并行计算, 运算完后将左边的数替换为结果, 这样就完成一次计算, 之后继续循环。这样每轮计算只需要计算两次的时间, 速度变快且与变量阶数无关。

$f(x) = x^7 + x$

$x$	$f(x)$	$\Delta_1$	$\Delta_2$	$\Delta_3$	$\Delta_4$	$\Delta_5$	$\Delta_6$	$\Delta_7$
1	2	128	1932	10206	25200	31920	20160	5040
2	130	2060	12138	35406	57120	52080	25200	5040
3	2190	14198	47544	92526	109200			
4	16388	61742	140070					
5	78130							

Diagram illustrating the parallel calculation process for the difference engine. Arrows indicate the flow of data and the addition of differences to the function values. For example,  $\Delta_1$  is added to  $f(1)$  to get  $f(2)$ , and so on. The process is repeated for higher-order differences.

$f(x) = x^7 + x$

$x$	$f(x)$	$\Delta_1$	$\Delta_2$	$\Delta_3$	$\Delta_4$	$\Delta_5$	$\Delta_6$	$\Delta_7$
1	2	128	1932	10206	25200	31920	20160	5040
2	130	2060	12138	35406	57120	52080	25200	5040
3	2190	14198	47544	92526	109200			
4	16388	61742	140070					
5	78130							

Diagram illustrating the parallel calculation process for the difference engine. Arrows indicate the flow of data and the addition of differences to the function values. For example,  $\Delta_1$  is added to  $f(1)$  to get  $f(2)$ , and so on. The process is repeated for higher-order differences.

Q: 对于一般的连续函数比如  $f(x)=e^x$  (步长 0.01), 差分机如何解决?

A: 利用泰勒展开, 将  $e^x$  展开为  $1+x+1/2x^2+1/6x^3+\dots$ 。这样展开后连续函数即可用多项式粗略表示, 保留项数越多精确度越高。假设保留到  $x$  的  $n$  次方项, 然后即可用差分机进行  $n$  阶差分获得常数, 进行加法运算。

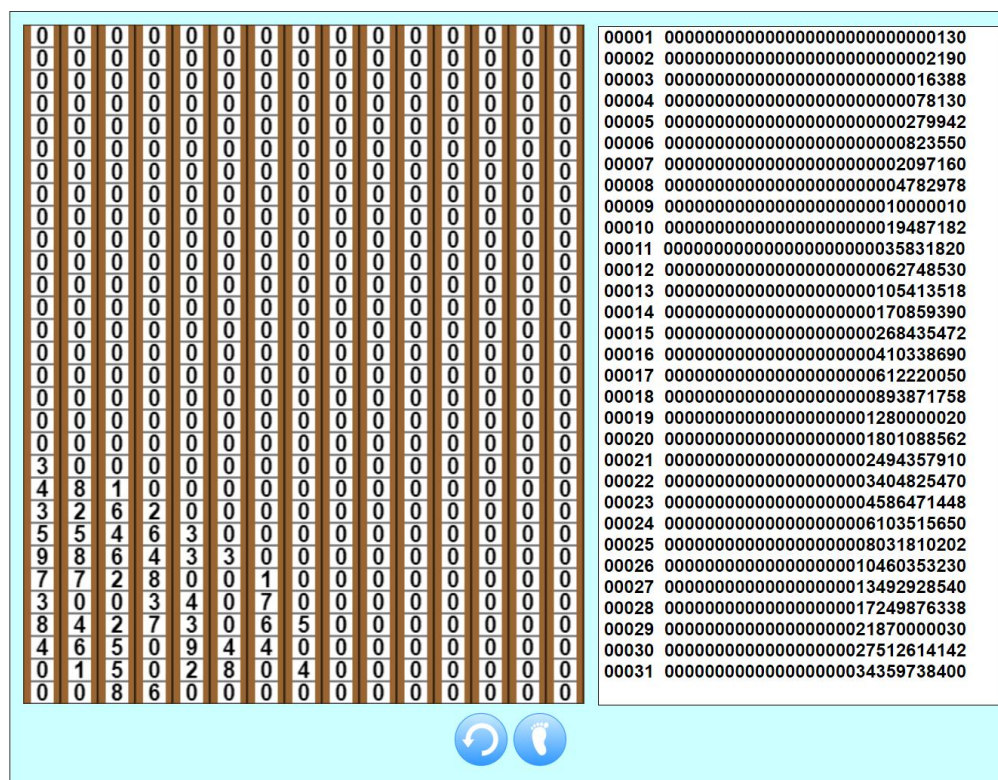
Q:  $f(x)=[x^2]$ (步长 0.01,  $[]$ 为取小数部分)适合用差分机打表吗? 思考一种较优的策略。

A: 可以。步长为 0.01 计算出来是四位小数, 则计算时可以忽略小数点, 用步长为 1 去计算  $x^2$  的值, 然后计算结果保留后四位数字就是小数部分。

## 差分机原理

对于  $n$  阶多项式,  $x$  每次步长相同, 则相邻两项进行  $n$  阶差分后即可得到常数差, 对常数差进行累加即可得到要求的多项式值。关键在于将非线性运算转换为线性运算。

以  $f(x) = x^7 + x$  为例，七阶差分后得到常数 5040，即可将前面部分全部累加得到不同  $x$  下  $f(x)$  的值。下图即是从  $x=2$  开始计算得到的  $f(x)$  结果。



## 串行计算与并行计算

串行计算是不将任务进行拆分，一个任务占用一块处理资源。

并行计算则不同。首先，并行计算可以划分成时间并行和空间并行。时间并行就是流水线技术，空间并行使用多个处理器执行并发计算。目前以研究空间并行为主。从空间并行的角度来说，并行计算将一个大任务分割成多个子任务，每个子任务占用一定处理资源。并行计算中不同子任务占用的不同的处理资源来源于同一块大的处理资源。换一个说法，就是将一块大的处理资源分为几块小的处理

资源，将一个大任务分割成多个子任务，用这些小的处理资源来单独处理这些子任务。并行计算中各个子任务之间是有很大的联系的，每个子任务都是必要的，其结果相互影响。

并行计算优点：

- 1、 计算速度快
- 2、 计算时间不取决于多项式的阶（都是计算两次）
- 3、 执行并行算法的硬件控制机关比串行算法的更易设计制造

## 巴贝奇与差分机简介

### Babbage's Machines



Charles Babbage  
(1791-1871)

Charles Babbage is one of the most fascinating figures in the history of computing.

Captivated by the idea that he could produce mathematical tables “by steam,” Babbage designed two early computing machines—the *Difference Engine* and the vastly more powerful *Analytical Engine*—that anticipated many of the features found in modern computers.

Neither machine was completed within Babbage’s lifetime. The Science Museum in London made a full-scale replica of the Difference Engine for the bicentennial of Babbage’s birth in 1991.

