首先推导出各阶导数的差分公式：

	f_i	f_{i+1}	f_{i+2}	f_{i+3}	f_{i+4}
$hf'(x_i)=$	-1	1			
$h^2f''(x_i)=$	1	-2	1		
$h^3f'''(x_i)=$	-1	3	-3	1	
$h^4f^{iv}(x_i)=$	1	-4	6	-4	1

$+O(h)$

图表 1 $O(h)$的差分公式

	f_{i-2}	f_{i-1}	f_i	f_{i+1}	f_{i+2}
$2hf'(x_i)=$		-1	0	1	
$h^2f''(x_i)=$		1	-2	1	
$2h^3f'''(x_i)=$	-1	2	0	-2	1
$h^4f^{iv}(x_i)=$	1	-4	6	-4	1

$+O(h^2)$

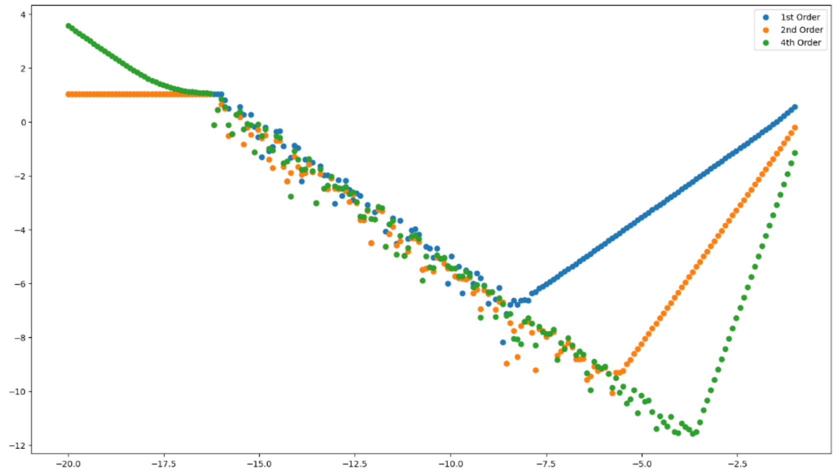
图表 2 $O(h^2)$的差分公式

	f_{i-3}	f_{i-2}	f_{i-1}	f_i	f_{i+1}	f_{i+2}	f_{i+3}
$12hf'(x_i)=$		1	-8	0	8	-1	
$12h^2f''(x_i)=$		-1	16	-30	16	-1	
$8h^3f'''(x_i)=$	1	-8	13	0	-13	8	-1
$6h^4f^{iv}(x_i)=$	-1	12	-39	56	-39	12	-1

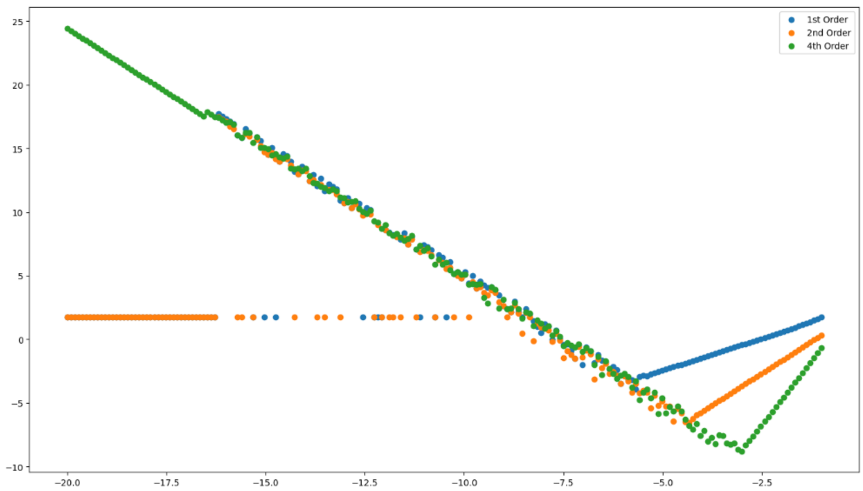
$+O(h^4)$

图表 3 $O(h^4)$的差分公式

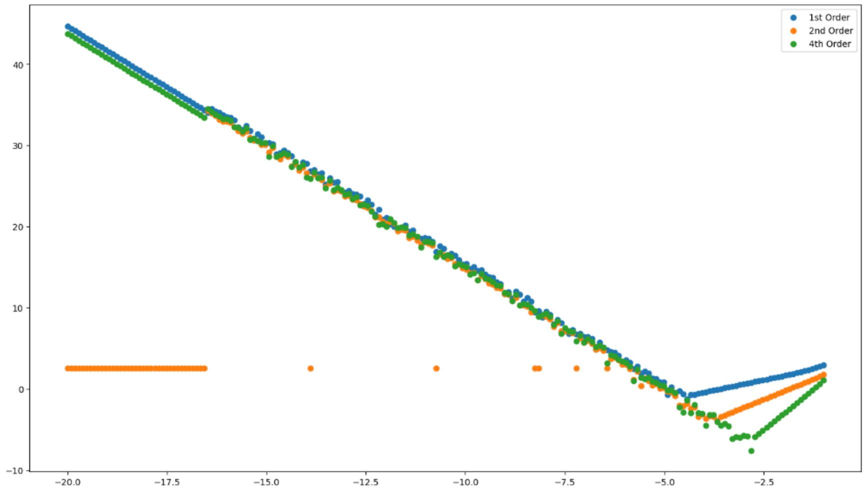
利用 python 画出$h-\delta$的对数值函数图（δ为误差绝对值）



图表 4 一阶导函数的差分误差对数图像



图表 5 二阶导函数的差分误差对数图像



图表 6 三阶导函数的差分误差对数图像

通过观察上图我们可以发现：

①不同的$\mathcal{O}(h)$的图像都会有一个拐点，在这个拐点右侧，图像大致为一条直线，而拐点左侧则为不连续的散点，比较离散。

②更高阶的$\mathcal{O}(h)$的差分公式对导函数值的拟合更加精确,图像上反应出的就是更低的曲线点。

③同时当 h 小到一定值时，图像变得平直。

现对上述异常现象做出解释。

利用计算机并通过公式$\frac{df(x)}{dx}=\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$估计近似值时，其同时会产生两种误差，一类是截断误差（truncation error），其与h的值成正相关；另一类是由于算法问题以及机器精度导致的舍入误差（round-off error），其与机器精度成正相关，与h成负相关。这两类误差的和组成了这一计算公式导致的总误差，并且由于两类误差与h的相关性不同，这就导致当h减小时，截断误差在减小，而舍入误差在增大。可以写出两类误差具体的表达式：

截断误差：

$$e_{tru}=\frac{h}{2f''(x)}$$

舍入误差：

$$e_{rou}=\frac{2cf(x)}{h}$$

其中c为机器精度

于是总误差为：

$$e=e_{tru}+e_{rou}=\frac{h}{2f''(x)}+\frac{2cf(x)}{h}$$

当h在$10^{-2},10^{-8}$中时，截断误差的影响占主导地位，所以此时总误差会随着h的减小而减小，而当$h>10^{-8}$时，截断误差趋近于 0 ，舍入误差的影响占主导地位了，所以此时总误差会随着h的减小而增大。而当$h>10^{-16}$时，此时，h接近于甚至小于机器精度 c 值，于是$f(x+h)-f(x)$在计算机的运算过程被直接舍入为 0 了，所以此时近似值为 0 。