1.数理算法原理

$f(x)$的泰勒展开:

$$f(x+\Delta x)=f(x)+\Delta x f'(x)+\Delta x^2\frac{f''(x)}{2!}+\ldots+\Delta x^n\frac{f^{(n)}(x)}{n!} \qquad (1)$$

将(1)式保留到三阶项，可得:

$$f(x+\Delta x)=f(x)+\Delta x f'(x)+\Delta x^2\frac{f''(x)}{2!}+\Delta x^3\frac{f'''(x)}{3!}+O(\Delta x^4) \qquad (2)$$

$$f(x-\Delta x)=f(x)-\Delta x f'(x)+\Delta x^2\frac{f''(x)}{2!}-\Delta x^3\frac{f'''(x)}{3!}+O(\Delta x^4)$$

两式相减得到一阶导数的中心差分格式:

$$f'(x)=\frac{f(x+\Delta x)-f(x-\Delta x)}{2\Delta x}+\Delta x^2\frac{f'''(x)}{3!}+O(\Delta x^3) \qquad (1.c)$$

同理对(1)式保留到四阶项，两式相加得到二阶导数的中心差分格式:

$$f''(x)=\frac{f(x+\Delta x)-2f(x)+f(x-\Delta x)}{2\Delta x^2}+\Delta x^2\frac{f''''(x)}{4!}+O(\Delta x^3) \qquad (2.c)$$

从推导中可以看到二者均为二阶精度

将(1)式保留到二阶项并移项得到一阶导数的前向差分格式:

$$f'(x)=\frac{f(x+\Delta x)-f(x)}{\Delta x}+\Delta x\frac{f''(x)}{2!}+O(\Delta x^2) \qquad (1.f)$$

将(1)式做替换并保留到三阶项可得到:

$$f(x+2\Delta x)=f(x)+2\Delta x f'(x)+4\Delta x^2\frac{f''(x)}{2!}+8\Delta x^3\frac{f'''(x)}{3!}+O(\Delta x^4) \qquad (3)$$

$(3)-2(2)$得到二阶导数的前向差分格式：

$$f''(x)=\frac{f(x+2\Delta x)-2f(x+\Delta x)+f(x)}{\Delta x^2}+\Delta x f'''(x)+O(x^2) \qquad (2.f)$$

从推导中可以看到二者均为一阶精度

2.代码生成和调试

打开./src/HW2.cpp，输入步长和自变量取值，可以得到以上差分与精确值的误差（以$sin(x)$为例）

3.结果讨论和解释

在程序输入中改变步长，控制自变量取值不变，可以发现误差受到步长和精度的影响。具体来说，误差与步长成正比，与精度成反比。

从1.中公式我们可以对原因进行分析：

将所有公式中误差的高阶小量略去，可以得到截断误差的最主要部分如下：

$$err_{1f}\approx\Delta x\frac{f''(x)}{2!}$$

$$err_{1c}\approx\Delta x^2\frac{f'''(x)}{3!}$$

$$err_{2f}\approx\Delta x f'''(x)$$

$$err_{2c} \approx \Delta x^2 \frac{f''''(x)}{4!}$$

从公式中可以看出，步长（即Δx）越大，截断误差越大，而精度为n阶时，截断误差近似为Δx^n项，因此截断误差与精度成反比。而舍入误差是因为浮点数存储位数有限产生的，因此与步长和精度均无关。两者相加得到误差，因此误差与步长成正比，与精度成反比。实际上，截断误差$err \propto \Delta x^n$，这里n为近似值精度的阶数。

控制其他变量不变，单精度存储的误差要高于双精度，这是因为舍入在单精度数发生的位数早于双精度数，因此单精度的舍入误差大于双精度。

附录1：AI工具使用声明表

AI工具名称	生成代码	功能
Deepseek R1	第26-30行	用于为模板功能提供示例，用模板功能可以使函数返回值类型根据输入变化

核心代码生成行数占比：83.333%

附录2：本次commit截图