说明文档

1 问题描述

题目要求用公式 $x_{n+1} = \frac{1}{2}(x_n + \frac{y}{x_n})$ 从一个初始数值开始不断迭代，去逼近 y 的平方根，并将每一步运算结果 x_{n+1} 与 y 的真实平方根的差的绝对值 ϵ_n 与给定的 ϵ_0 比较，直到 $\epsilon_n < \epsilon_0$ 时，精度达到要求，或经过足够多次循环仍达不到精度要求（$\epsilon_n > \epsilon_0$），停止循环，返回结果。

2 解答思路

2.1 第一问

按照题目要求定义一个 `babylonian` 函数，通过 `for` 循环不断迭代，同时设置最大循环次数 `max_iterations = 1000` 解决部分情况下精度始终达不到要求的情况。

2.2 第二问

题目要求研究每次迭代后误差 ϵ_n 的变化,因此我们再定义一个 `error_analysis` 函数,记录每一步迭代后误差的变化,同样设置最大循环次数 `max_iterations = 1000` 解决部分情况下精度始终达不到要求的情况。

2.3 第三问

对不同 x_0，分别使用 `error_analysis` 记录每一步迭代后误差的变化，分别画出收敛曲线。

结论：x_0 越接近 $\sqrt{y}$，收敛越快

2.4 第四问

同样再定义 `babylonian2` 函数，将误差由 $|x_n - \sqrt{y}|$ 改为 $|x_n - x_{n-1}|$ 即可。

结论：ϵ_n 随 n 的增大而减小，不断趋近于 0

3 如何使用代码

3.1 第一部分

在 Jupyter Notebook 文件中按顺序运行代码单元，在 (1) 中第二个代码单元中可调整 y, ϵ, x_0 的值，使用 `babylonian` 函数。

3.2 第二部分

在 (2) 中第一个代码单元中可调整 y, ϵ_0, x_0 的值，然后依次运行代码单元，即可得到图表。

3.3 第三部分

在 (3) 中第一个代码单元中可调整 y, ϵ_0 和不同 x_0 的值，然后依次运行代码单元，即可得到图表。

3.4 第四部分

在 (3) 中第一个代码单元中可调整 y 的值（如需改变 ϵ, x_0 直接增加 `epsilon = _` 等即可），然后依次运行代码单元，可在第三个代码单元中调整 y, ϵ, x_0 的值，使用 `babylonian2` 函数，也可运行第四个代码单元得到图表（数据以第一个代码单元为准）。