

III. 步数递增下，不同精度不同方法的误差、CPU 时间和收敛速度的计算结果

III-a. 第一个初值问题

第一个初值问题因为是以 T_1 为周期的，所以可以将初始值 $u(0)$ 视为 u 在 T_1 的精确解，既 $u(T_1) = u(0)$ 。假设一个 p 阶精度的方法在步长为 h 所得到的数值解 U 在 T_1 的取值为 $U^{T_1/h}$ ，则可以定义误差 $E(h) = \|U^{T_1/h} - u(T_1)\|_\infty$ 。此时由之前所学的收敛性理论可知，若 h 足够的小，我们有

$$E(h) = Ch^p + o(h^p), \quad h \rightarrow 0 \quad (1)$$

$$\Rightarrow E(h) \approx Ch^p \quad (2)$$

加密网格，使得步长 h 缩小为 $\frac{1}{2}h$ ，此时同样有

$$E(h/2) \approx C(h/2)^p \quad (3)$$

结合 (2) 可得

$$p \approx \log_2(E(h)/E(h/2)) \quad (4)$$

所以根据 (4)，我们可以通过前后两次的误差计算收敛速度 $p' = \log_2(E(h)/E(h/2))$ 。

具体的计算结果见./result 中的文本文件”Test1_result.txt”。

III-b. 第二个初值问题

第二个初值问题采用的是 Richardson extrapolation 计算误差和收敛速度。首先定义网格的误差，设 U, u 分别为近似解和精确解，近似解的步长为 h ，而其中网格点设为 x_1, x_2, \dots, x_N ，其中 $U_i \approx u(x_i)$ ，定义一个点上的误差为 $e_i = U_i - u(x_i)$ ，进一步可以定义整个网格的误差为 $E(h) = U(h) - u = h \sum_{i=1}^N |e_i|$ ，又由 (2) 可知 $E(h) \approx hCh^p \frac{T}{h} = C'h^p$ 。

对于该初值问题，因为不知道精确解，所以我们先假设三个不同步长 $h, h/2, h/4$ 所得到的近似解分别为 $U(h), U(h/2), U(h/4)$ ，此时由上面网格的定义，分别将 $U(h/2), U(h/4)$ 作为精确解得到

$$\hat{E}(h) = U(h) - U(h/2) \quad (5)$$

$$\hat{E}(h/2) = U(h/2) - U(h/4) \quad (6)$$

又因为 $E(h) \approx C'h^p$ 以及 $\hat{E}(h) = E(h) - E(h/2)$ ，从而有

$$\hat{E}(h) \approx C'(1 - \frac{1}{2^p})h^p \quad (7)$$

$$\hat{E}(h/2) \approx C'(1 - \frac{1}{2^p})\frac{h^p}{2^p} \quad (8)$$

从而有

$$\hat{E}(h)/\hat{E}(h/2) \approx 2^p \quad (9)$$

$$p \approx \log_2(\hat{E}(h)/\hat{E}(h/2)) \quad (10)$$

所以根据 (10)，我们可以计算收敛速度 $p' = \log_2(\hat{E}(h)/\hat{E}(h/2))$ 。

具体的计算结果见./result 中的文本文件”Test2_result.txt”。

III-c. 结果分析

从两个初值问题所得到的结果可以看出，收敛速度 p' 的确随着我们选择方法的精度 p 的增加而增加，并且大体上有 $p' \approx p$ 。

IV. 其他计算结果

IV-a. 不同方法达到其图像结果与精确图像肉眼区分不了所需要的最小步数

下面结果中使用的是四阶 Adams-Bashforth、五阶 Adams-Moulton、四阶 Bdfs 以及传统的四阶 Runge-Kutta

第一个初值问题

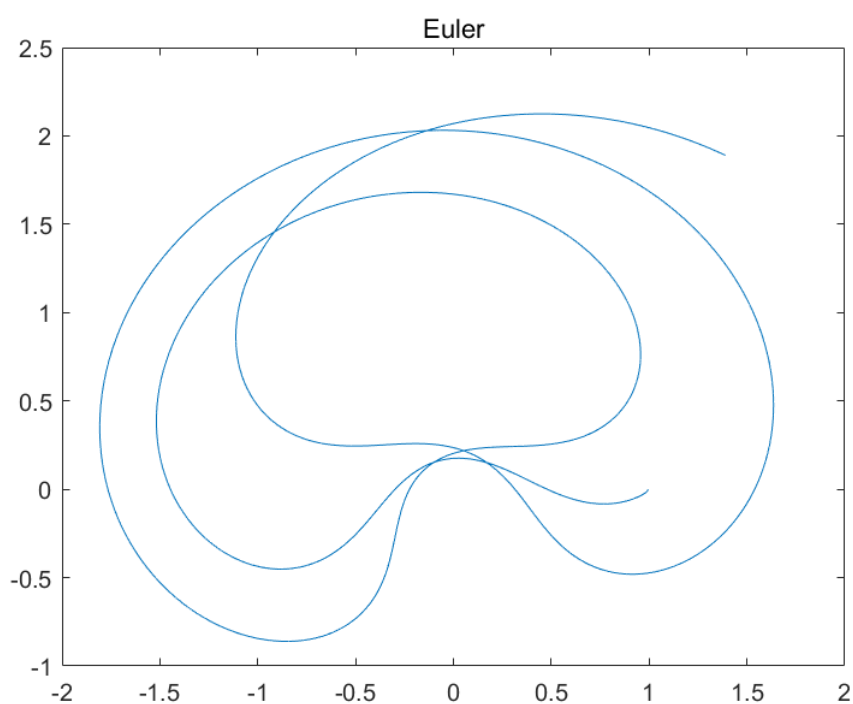
	Adams Bashforth	Adams Moulton	Bdfs	Runge Kutta
步数	50000	20000	35000	12000

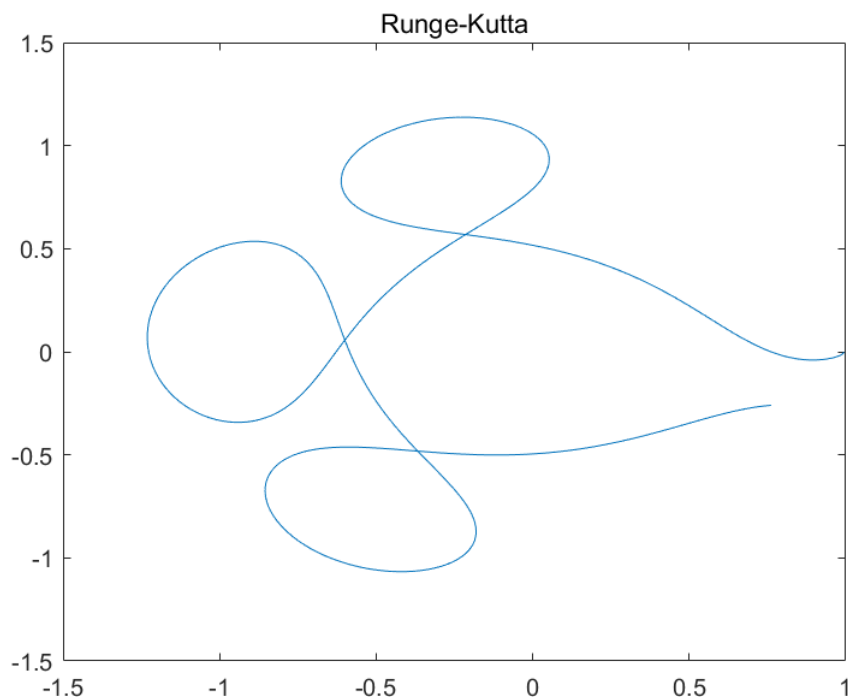
第二个初值问题

	Adams Bashforth	Adams Moulton	Bdfs	Runge Kutta
步数	2500	1100	2000	800

IV-b. Euler 方法和 Runge-Kutta 方法在第一个初值问题中的图像结果

下面两幅图分别是步数为 24000 的 Euler 方法和步数为 6000 的 Runge-Kutta 方法得到的图像结果：





从图中可以看到, Euler 方法得到的图像是不稳定的, 而 Runge-Kutta 方法得到的图像是稳定的, 且已经比较接近精确解。具体的原因从两种方法的稳定区域图可以看出, 传统四阶 Runge-Kutta 方法的稳定区域更大, 而 Euler 方法的稳定区域相对要小, 所以导致了 Euler 方法更容易不稳定。

IV-c. 对于第一个初值问题, 不同方法达到误差 $\leq 10^{-3}$ 所需要的 CPU 时间

下面结果中使用的是四阶 Adams-Bashforth、五阶 Adams-Moulton、四阶 Bdfs 以及传统的四阶 Runge-Kutta

	Adams Bashforth	Adams Moulton	Bdfs	Runge Kutta
步数	390000	90000	370000	90000
误差	0.00100787	0.000988293	0.00105583	0.000988999
CPU 时间	3.2193 s	2.51664 s	6.97739 s	2.24541 s

从上表可以看出传统四阶 Runge-Kutta 方法表现最为优异。