

计算物理学作业 4

朱寅杰 1600017721

4.1 求解 Lotka-Volterra 方程的初值问题

对于微分方程组

$$(\dot{x}, \dot{y}) = (\alpha x - \beta xy, \delta xy - \gamma y) \quad (1)$$

作变量代换 $(X, Y, T) = (\delta x/\alpha, \beta y/\alpha, \alpha t)$, 得到

$$(\dot{X}, \dot{Y}) = (X(1 - Y), Y(X - \gamma/\alpha)), \gamma/\alpha > 0 \quad (2)$$

对于这个方程组, 我们首先考察其不含时的不动点解 (即 $\dot{X} = \dot{Y} = 0$ 的解): 其一为 $X(T) = Y(T) = 0$, 其一为 $X(T) = \gamma/\alpha, Y(T) = 1$ 。作一个微扰的话,

- a) 对于前者, 设 $X(T) = \epsilon_1(T)$, $Y(T) = \epsilon_2(T)$, 其中 ϵ 均为小量。从而有 $(\dot{\epsilon}_1, \dot{\epsilon}_2) = (\epsilon_1(1 - \epsilon_2), (\epsilon_1 - \gamma/\alpha)\epsilon_2)$, 再对 T 求导, 并保留至最低阶小量, 有

$$(\ddot{\epsilon}_1, \ddot{\epsilon}_2) = (\epsilon_1(1 - \epsilon_2)^2 - \epsilon_1\epsilon_2(1 - \gamma/\alpha), \epsilon_2(\epsilon_1 - \gamma/\alpha)^2 + \epsilon_1\epsilon_2(1 - \epsilon_2)) \approx (\epsilon_1, -(\gamma/\alpha)^2\epsilon_2)$$

也就是说对于 X 方向上的微扰这个不动点是不稳定的, 而对于 Y 方向上的微扰这个不动点是稳定的。

- b) 对于后者, 取 $X(T) = \gamma/\alpha + \epsilon_1(T)$, $Y(T) = 1 + \epsilon_2(T)$, 则有

$$(\dot{\epsilon}_1, \dot{\epsilon}_2) = (-(\epsilon_1 + \gamma/\alpha)\epsilon_2, (1 + \epsilon_2)\epsilon_1)$$

再对 T 求导, 保留至一阶小量, 得到

$$(\ddot{\epsilon}_1, \ddot{\epsilon}_2) = (-(\epsilon_1 + \gamma/\alpha)(1 + \epsilon_2)\epsilon_1, \epsilon_1^2(1 + \epsilon_2) - (1 + \epsilon_2)(\epsilon_1 + \gamma/\alpha)\epsilon_2) \approx (-\epsilon_1\gamma/\alpha, -\epsilon_2\gamma/\alpha)$$

故这是一个稳定不动点。

事实上, 这个微分方程组是可积的。将两式相除得到

$$\frac{dY}{dX} = \frac{Y(X - \gamma/\alpha)}{X(1 - Y)}$$

分离变量积分得到 $X + Y - \log(X^{\gamma/\alpha}Y) = \text{常数}$ 。写作变量代换之前的形式, 就是 $\delta x + \beta y - \gamma \log x - \alpha \log y = \text{常数}$ 。

最后, 我们使用 Runge-Kutta 法来数值求解这个问题中 $\alpha = 2/3, \beta = 4/3, \gamma = \delta = 1$ 的特殊情况。求解的代码见 RK.py。总共计算了 $(x, y) = (0.8, 0.8)$ 、 $(1.0, 1.0)$ 、 $(1.2, 1.2)$ 、 $(1.4, 1.4)$ 、 $(1.6, 1.6)$ 五组初值的解, 将五个初值下 (x, y) 的演化轨迹一并画在下图中。从图中可见, 这几个初值解出的都是围绕着不动点 $(x, y) = (1.0, 0.5)$ (对应于 $(X, Y) = (\gamma/\alpha, 1)$) 逆时针走行的闭合曲线, 而这闭合曲线正是上面求出的这个微分方程组的初积分的等值线。

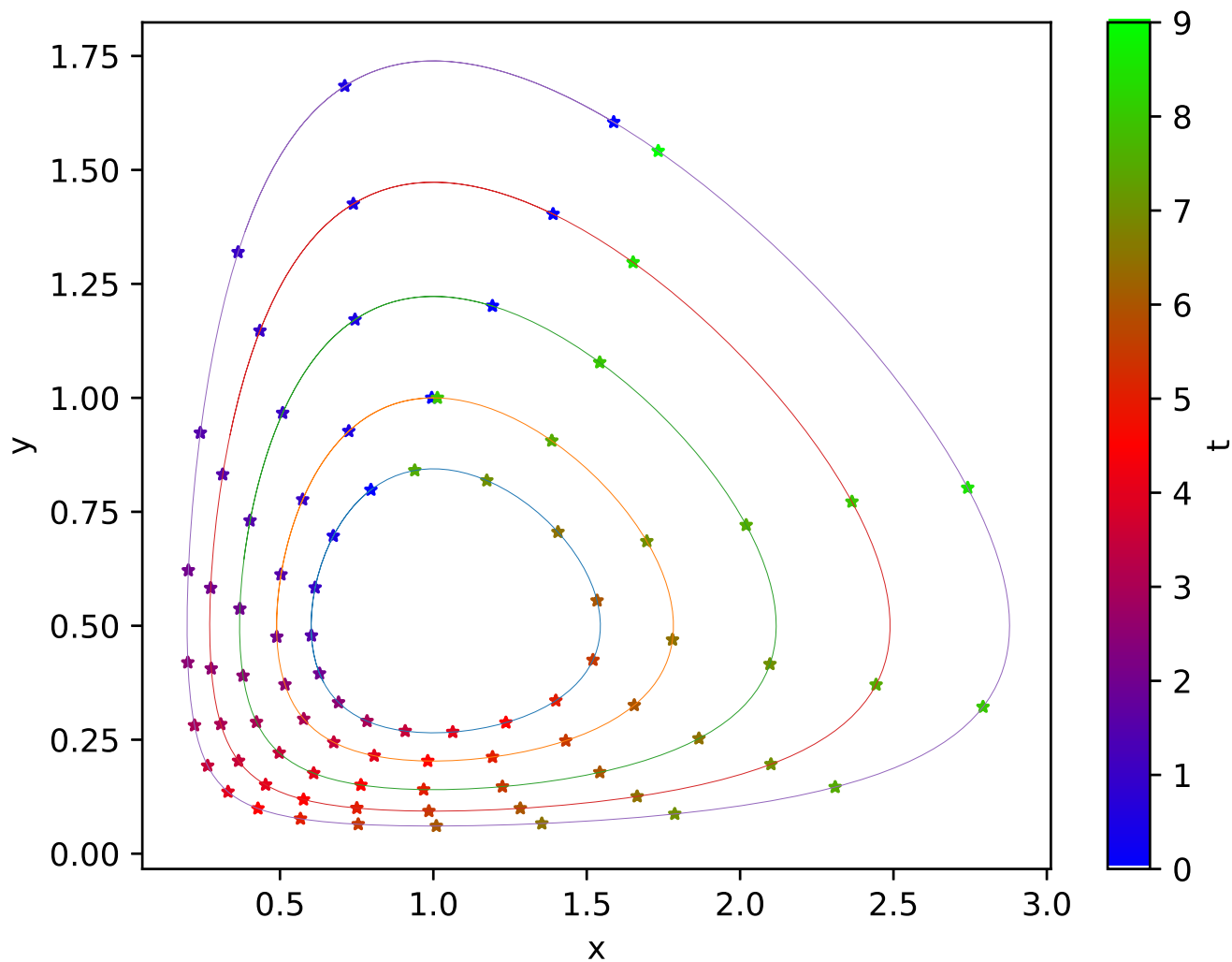


图 1: 图中由内至外分别是初值为 $(x, y) = (0.8, 0.8), (1.0, 1.0), (1.2, 1.2), (1.4, 1.4), (1.6, 1.6)$ 的解以参数方程的方式画出的曲线。为了标示 (x, y) 随 t 的演化规律, 从 $t = 0$ 开始, 每隔 0.5 在图线上的 $(x(t), y(t))$ 处打一个节点, 直至图线转过一周。不同的 t 对应的点的颜色是由蓝到红到绿渐变的, 所以如果你对色彩的敏感度足够高, 就可以直接通过颜色 (而不用从初始点开始一个一个数) 来确定在 t 为某值时各个解都演化到什么地方了。图片是矢量图, 因此如果想要观察细节可以尽情地 zoom in。

4.2 不同的稀疏矩阵迭代法求解泊松方程边值问题的速度比较

待解的方程是

$$-\nabla^2 u(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y), (x, y) \in (0, 1) \times (0, 1)$$

并且给出在边界上函数的值为零。将待解的区域分割成 $h = \frac{1}{N+1}$ 见方的方格，把算子 ∇^2 作分立化处理，求解在格点上的 $N \times N$ 个函数值。这实际上就变成了一个稀疏矩阵的线性方程组的求解问题，可以使用迭代进行求解。

我们分别使用了 Jacobi 法、Gauss-Seidel 法和经过 Chebyshev 加速的弛豫算法（感谢刘老师的讲义，弛豫算法与 Chebyshev 加速中需要计算的谱半径等参数都已经帮我们算好了，只需拿来使用即可）来进行求解，并统计其迭代次数。为便于比较，迭代停止的标准统一选作各格点处方程左右两边差的绝对值不超过 10^{-4} 。计算的源代码分别为 Jacobi.py、Gauss-Seidel.py 和 SOR.py。分别计算了 $N = 10, 20, \dots, 60$ 的情况（虽然题目里只要求算了 10、20 和 50），结果列于下表中：

N	Jacobi		Gauss-Seidel		SOR	
	迭代步数	r_{max}	迭代步数	r_{max}	迭代步数	r_{max}
10	295	9.745×10^{-5}	149	9.413×10^{-5}	33	7.310×10^{-5}
20	1086	9.894×10^{-5}	544	9.912×10^{-5}	63	9.410×10^{-5}
30	2370	9.994×10^{-5}	1187	9.900×10^{-5}	94	8.362×10^{-5}
40	4149	9.996×10^{-5}	2076	9.971×10^{-5}	124	9.051×10^{-5}
50	6423	9.981×10^{-5}	3213	9.965×10^{-5}	154	9.487×10^{-5}
60	9190	9.991×10^{-5}	4596	9.993×10^{-5}	184	9.793×10^{-5}

可以看到，Gauss-Seidel 迭代所用次数几乎等于 Jacobi 迭代的一半，二者基本都以 N^2 的方式增长。而优化后的弛豫算法的迭代次数是随着 N 线性增长的（而且线性十分好）。如果进一步拟合的话可以发现，三者的增长规律分别是 $2.4714(N + 0.99)^2 - 3$ 、 $1.235(N + 1.01)^2 - 1$ 、 $3.023(N + 1)$ ，与理论预期符合得相当好。