奇异摄动理论入门学习报告

赵智阳

January 15, 2026

1 核心思想理解

奇异摄动理论是研究含小参数系统中、因参数趋于零导致解的性质发生突变（如边界层、内部层等急剧变化）的多尺度分析方法。

1.1 正则摄动与奇异摄动

正则摄动与奇异摄动是处理含小参数问题的两种不同方法，核心区别在于解的渐近行为是否一致收敛于退化问题的解。

正则摄动的核心特征是小参数 ϵ 的影响是温和、全局一致的。即当 $\epsilon \to 0$ 时，解 $y(x;\epsilon)$ 在整个区域上一致收敛于退化问题 $(\epsilon = 0)$ 的解 $y_0(x)$。求解时可直接将解展开为 ϵ 的幂级数 $y = y_0 + \epsilon y_1 + \epsilon^2 y_2 + \dots$，再代回问题逐项求解。常见问题如有弹簧刚度微调后的振动频率计算。

而奇异摄动的核心特征：小参数 ϵ 导致问题的定性性质发生剧烈、局部突变。当 $\epsilon \to 0$ 时，解 $y(x;\epsilon)$ 非一致收敛。在退化问题 $(\epsilon = 0)$ 中解的某些特征（如边界条件、方程阶数/类型）会丢失，从而在局部区域产生薄层，层内解发生剧烈变化。求解时需要采用匹配渐近展开、多重尺度等特殊技巧，针对不同尺度区域分别展开并匹配。物理实例有小粘性流体在高雷诺数下绕物体流动，即普朗特边界层理论。

1.2 边界层理论

面对无粘理论失效、全粘性理论难以求解的现状，德国力学家路德维希·普朗特于 1904 年开创边界层理论，系统解决了高雷诺数下流体绕物体流动时，无粘理论与实验观测间的根本矛盾。

1. 无粘理论失效：欧拉方程（$v = 0$）允许流体在物面“滑移”，无法解释壁面无滑移条件及由此产生的摩擦阻力。
2. 全粘性理论难以求解：纳维-斯托克斯方程（N-S 方程）在全域求解极端困难（高雷诺数时数值刚性极大）。

普朗特发现，小粘性（$\nu \to 0$ 或 $Re \to \infty$）的影响并非全局，而是高度局域化在物体表面附近一个极薄的区域内，该区域被称为边界层。基于此，他提出了两个关键思想：

1. 尺度分离：外层区域下，惯性力主导，粘性力可忽略，可用简化后的欧拉方程描述。内层区域下，粘性力与惯性力同等重要，必须同时保留。
2. 尺度变换（坐标拉伸）引入边界层坐标变换以解析内层行为。对于平板流动，法向坐标 y 被拉伸为 $Y = \frac{y}{\delta(Re)}$， 其中边界层厚度 $\delta \sim Re^{-1/2}$（对于层流）。通过系统的量级分析，纳维-斯托克斯方程在边界层内可简化为更易求解的普朗特边界层方程。

2 案例分析

考察二维常微分方程

$$\varepsilon y''(x) + y'(x) = 0, \quad 0 < x < 1$$

$$y(0) = 0, \quad y(1) = 1$$

其中 $0 < \varepsilon \ll 1$ 为小参数, 且有精确解

$$y_{\text{exact}}(x) = \frac{1 - e^{-x/\varepsilon}}{1 - e^{-1/\varepsilon}}$$

2.1 匹配渐进展开法

方程中一阶项系数 $a(x) = 1 > 0$，根据边界层理论，当 $a(x) > 0$ 时，边界层出现在左端点 $x = 0$ 附近。这意味着：

- 外解（远离边界层的区域）应满足右边界条件 $y(1) = 1$
- 内解（边界层内）负责满足左边界条件 $y(0) = 0$

设外解展开为：

$$y_{\text{out}}(x) = y_0(x) + \varepsilon y_1(x) + O(\varepsilon^2)$$

代入 $\varepsilon = 0$，有退化方程

$$y_0'(x) = 0$$

解得：

$$y_0(x) = C$$

外解匹配右边界条件 $y(1) = 1$，得

$$y_{\text{out}}(x) = 1 + O(\varepsilon)$$

在左边界层内，引入拉伸坐标（内变量）

$$X = \frac{x}{\varepsilon^p}$$

其中 $p > 0$ 为待定尺度指数。令 $y(x) = Y(X)$，则有

$$\frac{d}{dx} = \varepsilon^{-p}\frac{d}{dX}, \quad \frac{d^2}{dx^2} = \varepsilon^{-2p}\frac{d^2}{dX^2}$$

代入原方程

$$\varepsilon^{1-2p}Y''(X) + \varepsilon^{-p}Y'(X) = 0$$

令

$$1 - 2p = -p \quad \Rightarrow \quad p = 1$$

因此边界层厚度为 $O(\varepsilon)$，内变量为

$$X = \frac{x}{\varepsilon}$$

此时方程简化为

$$Y''(X) + Y'(X) = 0$$

设内解展开为

$$Y_{\text{in}}(X) = Y_0(X) + \varepsilon Y_1(X) + O(\varepsilon^2)$$

零阶内方程通解为

$$Y_0(X) = A + Be^{-X}$$

其中 A, B 为待定常数。

左边界条件 $y(0) = 0$ 转换为 $Y_0(0) = 0$，代入得：

$$A + B = 0 \quad \Rightarrow \quad B = -A$$

因此

$$Y_0(X) = A(1 - e^{-X})$$

下面应用匹配原理, 即内解的外极限应等于外解的内极限。

- 外解的内极限：当 $x \to 0^+$，由外层方程解得 $y_{\text{out}} \to 1$
- 内解的外极限：当 $X \to \infty$，由内层方程解得 $Y_0(X) \to A$

令二者相等 $A = 1$，代入内层解，得完整的零阶内解

$$Y_0(X) = 1 - e^{-X}$$

复合展开的一般形式为：

$$y_{\text{comp}}(x) = y_{\text{out}}(x) + Y_{\text{in}}(X) - y_{\text{match}}$$

其中 y_{match} 为内外解的共同部分（匹配值）。本例中：

$$\begin{aligned} y_{\text{out}}(x) &= 1 \\ Y_{\text{in}}(X) &= 1 - e^{-X} \\ y_{\text{match}} &= 1 \end{aligned}$$

因此零阶复合解为：

$$y_{\text{approx}}(x) = 1 + (1 - e^{-x/\varepsilon}) - 1 = 1 - e^{-x/\varepsilon}$$

2.2 结果分析

比较精确解 $y_{\text{exact}}(x)$ 和近似解 $y_{\text{approx}}(x)$：

- 当 $\varepsilon \to 0$ 时，$e^{-1/\varepsilon} \to 0$，精确解简化为 $1 - e^{-x/\varepsilon}$，与我们的渐近解一致
- 在边界层内 ($x \sim O(\varepsilon)$)，$e^{-x/\varepsilon} = O(1)$，解从 0 快速上升到 1
- 在边界层外 ($x \gg \varepsilon$)，$e^{-x/\varepsilon} \approx 0$，解近似为常数 1，与退化方程解一致

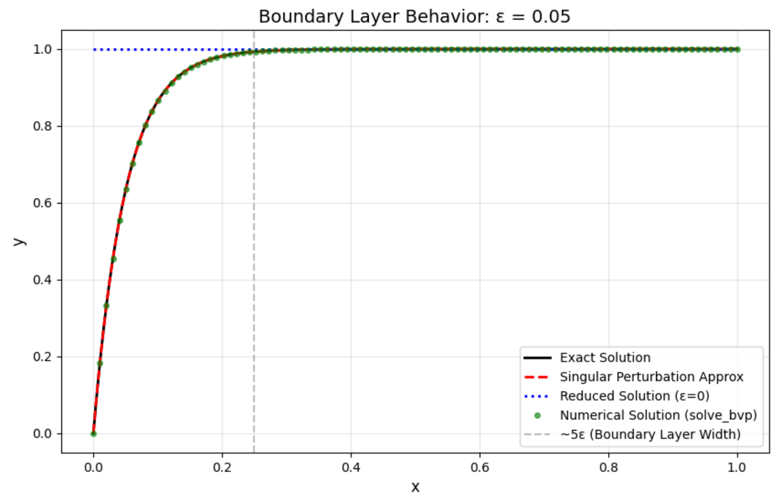

Figure 1: python 模拟下的解图像，$\epsilon = 0.05$

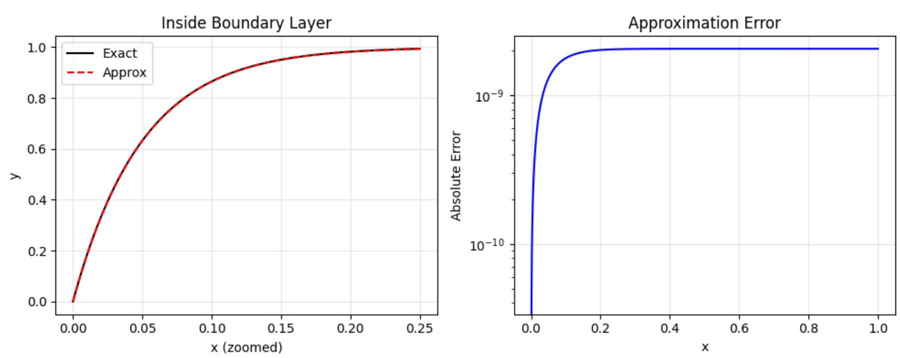

Figure 2: 边界层内部与近似误差对数图

3 附录:Python 代码

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_bvp

#参数
epsilon = 0.05

#精确解
x_exact = np.linspace(0, 1, 1000)
y_exact = (1 - np.exp(-x_exact/epsilon)) / (1 - np.exp(-1/epsilon))

#奇异摄动近似解
y_approx = 1 - np.exp(-x_exact/epsilon)

#退化方程解
y_reduced = np.ones_like(x_exact)

#数值求解原方程
def ode(x, y):
    return np.vstack((y[1], -y[1]/epsilon))

def bc(ya, yb):
    return np.array([ya[0], yb[0]-1])

x_num = np.linspace(0, 1, 100)
y_num = np.zeros((2, x_num.size))
y_num[0] = x_num  # 初始猜测

sol = solve_bvp(ode, bc, x_num, y_num, max_nodes=10000)
x_sol = sol.x
y_sol = sol.y[0]

# 绘图
plt.figure(figsize=(10, 6))
plt.plot(x_exact, y_exact, 'k-', linewidth=2, label='Exact Solution')
plt.plot(x_exact, y_approx, 'r--', linewidth=2, label='Singular Perturbation Approx')
plt.plot(x_exact, y_reduced, 'b:', linewidth=2, label='Reduced Solution (ε=0)')
plt.plot(x_sol, y_sol, 'go', markersize=4, label='Numerical Solution (solve_bvp)', alpha=0.6)

plt.xlabel('x', fontsize=12)
plt.ylabel('y', fontsize=12)
plt.title(f'Boundary Layer Behavior: ε = {epsilon}', fontsize=14)
plt.legend(loc='best', fontsize=10)
plt.grid(True, alpha=0.3)
```

```
plt.axvline(x=5*epsilon, color='gray', linestyle='--', alpha=0.5, label='~5ε (Boundary Layer Wi
plt.legend()

# 子图：边界层细节
plt.figure(figsize=(10, 4))
plt.subplot(1, 2, 1)
x_detail = np.linspace(0, 5*epsilon, 200)
y_exact_detail = (1 - np.exp(-x_detail/epsilon)) / (1 - np.exp(-1/epsilon))
y_approx_detail = 1 - np.exp(-x_detail/epsilon)
plt.plot(x_detail, y_exact_detail, 'k-', label='Exact')
plt.plot(x_detail, y_approx_detail, 'r--', label='Approx')
plt.xlabel('x (zoomed)')
plt.ylabel('y')
plt.title('Inside Boundary Layer')
plt.grid(True, alpha=0.3)
plt.legend()

plt.subplot(1, 2, 2)
error = np.abs(y_exact - y_approx)
plt.semilogy(x_exact, error, 'b-')
plt.xlabel('x')
plt.ylabel('Absolute Error')
plt.title('Approximation Error')
plt.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()
```