《面向领域的并行数值方法》作业1

陳日康　信息与计算科学　22336049

一、题目描述

已知下列非齐次两点边值问题：$\begin{cases} Lu = -\dfrac{d}{dx}(p\dfrac{du}{dx}) + qu = f, \quad a < x < b \\ u(a) = \alpha, \qquad u'(b) = \beta, \end{cases}$

设 $[a, b] = [-1, 1]$，$p(x) \equiv -(-\pi^2 - 1)^{-1}$，$q(x) \equiv 1$，$\alpha = 0$，$\beta = -\pi e$，以及
$f(x) = \dfrac{2\pi}{\pi^2 - 1}. \cos(\pi x). e^x$。

任务 1：分别取 $h = 0.20,\ 0.10,\ 0.05,\ 0.02$ 时，将求解域 $[a, b]$ 等分为长度为 h 的单元或子区间；基于上述剖分，参考教材 3.2.1 与 3.2.4 等章节内容，编程构建针对边值问题的相应差分方程；分别使用高斯消去法和雅可比迭代法（迭代 30 次），求解上述差分方程获得数值解 u_h，绘制数值解 u_h 的函数图像。

任务 2：已知 $u(x) = \sin(\pi x) \cdot e^x$ 是上述边值问题的解析解，针对不同的步长 h 和不同计算方法得到的数值解 u，绘制误差函数 $u_h - u$ 的函数图像，且进行观察分析。

二、实现过程

2.1 推导过程

本次作业采用中心差分方法进行离散化，并分别实现了高斯消去法与雅可比迭代法两种求解方式。在差分离散方面，依据教材的内容，在网格等距且系数函数 $p(x)$ 为常数的前提下，使用中心差分格式对微分算子 $-\dfrac{d}{dx}(p(x)u') + q(x)u$ 进行离散。设网格步长为 h，节点为 $x_i = a + ih$，则在内部节点处，微分算子可近似为 $\dfrac{p}{h^2}(u_{i-1} - 2u_i + u_{i+1}) + qu_i = f(x_i)$，从而形成一个对称的三对角线性方程组。

而在边界处理方面，左端点采用 Dirichlet 条件 $u(-1) = \alpha$，右端点采用 Neumann 条件 $u'(1) = \beta$。Neumann 条件使用向后差分公式逼近导数，即 $u'(x_N) \approx \dfrac{u_N - u_{N-1}}{h} = \beta$，由此构造出右端点的差分格式。

在求解方法方面，高斯消去法具有较高的计算精度，适合中小规模线性系统的直接求解；雅可比迭代法则基于将系数矩阵拆分为对角矩阵 D 和剩余部分 $R = A - D$，并使用迭代公式 $x^{(k+1)} = D^{-1}(b - Rx^{(k)})$ 在设定最大迭代次数 30 次内逐步逼近数值解。

2.2 关键代码解析

1. 差分系统构造 `construct_system(h)`：

```
def construct_system(h):
    ...
    N = int((b - a) / h)         # 网格节点数
    x = np.linspace(a, b, N + 1)  # 均匀划分的网格点
    ...
    A = np.zeros((N+1, N+1))     # 系数矩阵初始化
    F = np.zeros(N+1)            # 右端向量初始化

    # 内部点使用中心差分
    for i in range(1, N):
        A[i, i-1] = p / h**2
        A[i, i]   = -2 * p / h**2 + q
        A[i, i+1] = p / h**2
        F[i] = f_rhs(x[i])

    # 左端点 Dirichlet 边界条件
    A[0, 0] = 1
    F[0] = alpha

    # 右端点 Neumann 边界条件
    A[N, N] = 1 / h
    A[N, N-1] = -1 / h
    F[N] = beta

    return A, F, x
```

在差分方程的构建过程中，首先对内部节点使用中心差分格式对算子 $-pu'' + qu$ 进行离散处理，构成三对角矩阵；对于边界条件的处理，左端点采用 Dirichlet 边界条件 $u(a) = \alpha$，设置为 $A[0,0] = 1$；而右端点则根据 Neumann 条件 $u'(b) = \beta$ ，采用向后差分近似导数，即以 $(u_N - u_{N-1})/h$ 替代 $u'(b)$，将其转化为线性代数形式，从而构成完整的边值离散系统。最后返回构造好的系数矩阵 A、右端项向量 F 以及网格节点数组 x。

2. 高斯消元法 `gauss_elimination(A, b)`：

```
def gauss_elimination(A, b):
    return np.linalg.solve(A, b)
```

使用 NumPy 的内置高斯消元器 `np.linalg.solve` 直接求解线性方程组 `Ax=b` （$Au = b$） 。

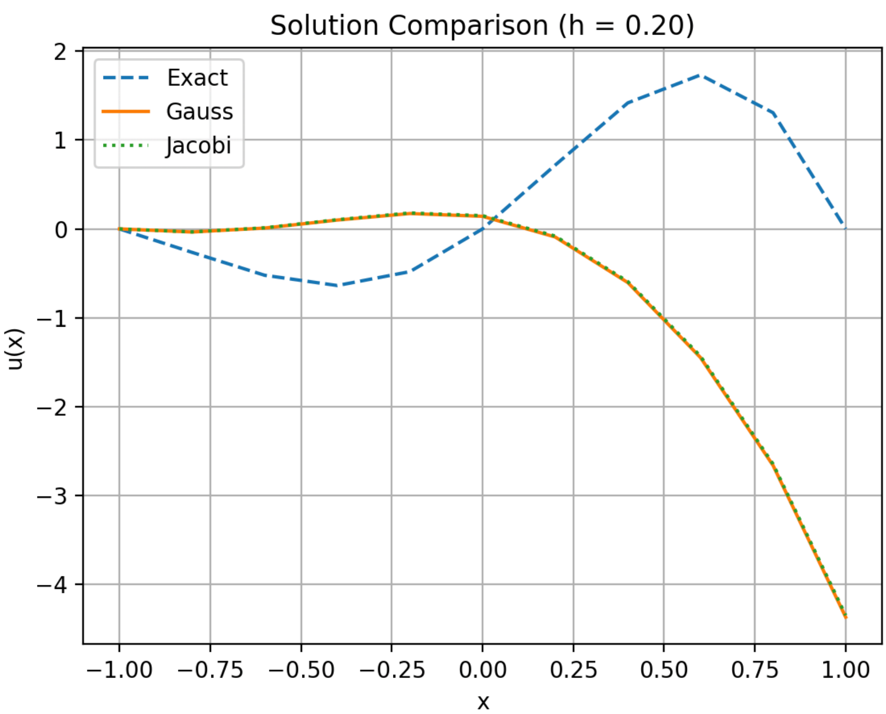
3. 雅可比迭代法 `jacobi(A, b, max_iter=30)`：

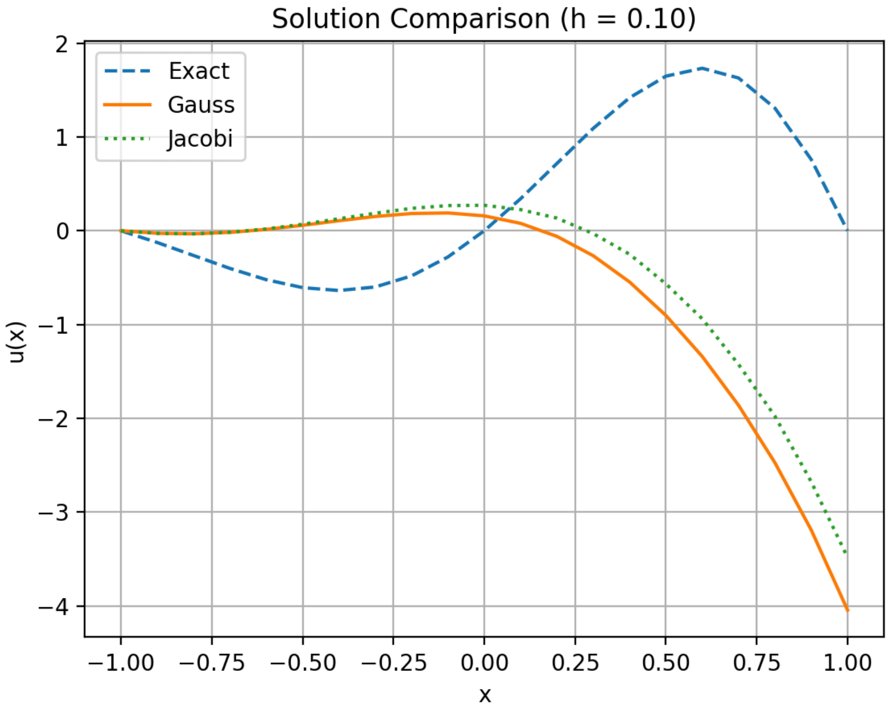
```
def jacobi(A, b, x0=None, max_iter=30):
    n = len(b)
    if x0 is None:
        x0 = np.zeros(n)
    D = np.diag(np.diag(A))       # 提取对角线矩阵 D
    R = A - D                     # 剩余部分 R = A - D
    D_inv = np.linalg.inv(D)      # 计算 D 的逆
    x = x0.copy()
    for _ in range(max_iter):
        x = np.dot(D_inv, b - np.dot(R, x))  # 迭代公式
    return x
```

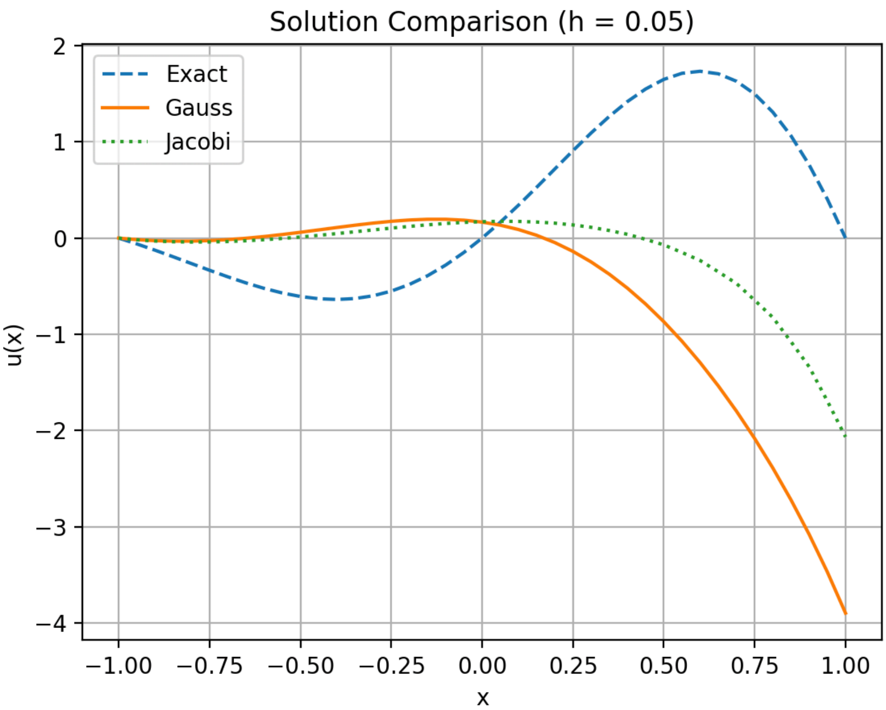
函数实现了雅可比迭代法用于求解线性方程组 `Ax=b`（$Au = b$），其中 `A` 为系数矩阵，`b` 为右端项向量，`x0` 为初始迭代向量，`max_iter` 表示最大迭代次数。将系数矩阵 A 拆分为对角部分 D 与剩余部分 $R = A - D$，利用迭代公式 $x^{(k+1)} = D^{-1}(b - Rx^{(k)})$ 不断更新解向量。函数开始时首先获取向量 b 的长度作为系统维度 `n`，并将初始迭代解设为全零向量初始解向量默认为全零向量，然后计算逆矩阵 D^{-1}，再通过差值计算得到非对角部分 `R`，函数进入住循环，执行最多 `max_iter` 次迭代（30 次）。每一步中，根据雅可比公式更新解向量 `x`，直到达到设定的迭代次数为止。最终返回当前近似解作为结果。

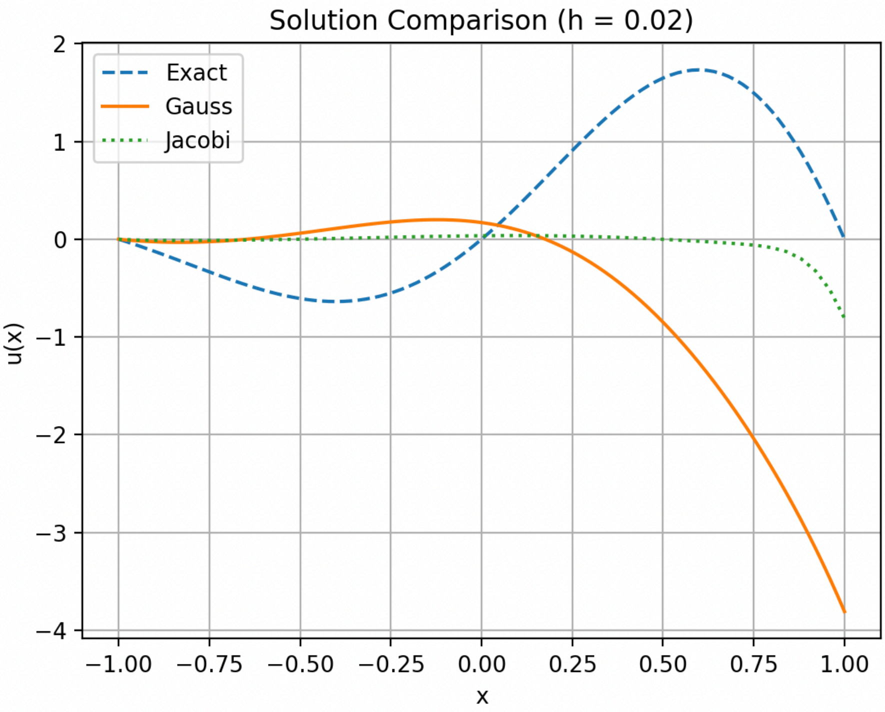
三、结果分析

任务1：



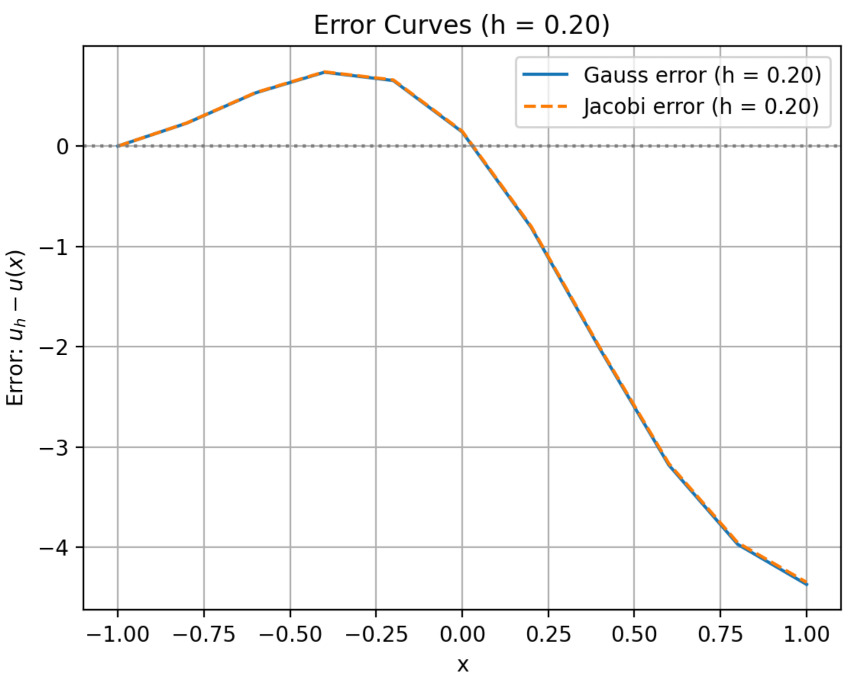

Solution Comparison (h = 0.10)
Exact
Gauss
Jacobi
u(x)
x
2
1
0
−1
−2
−3
−4
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

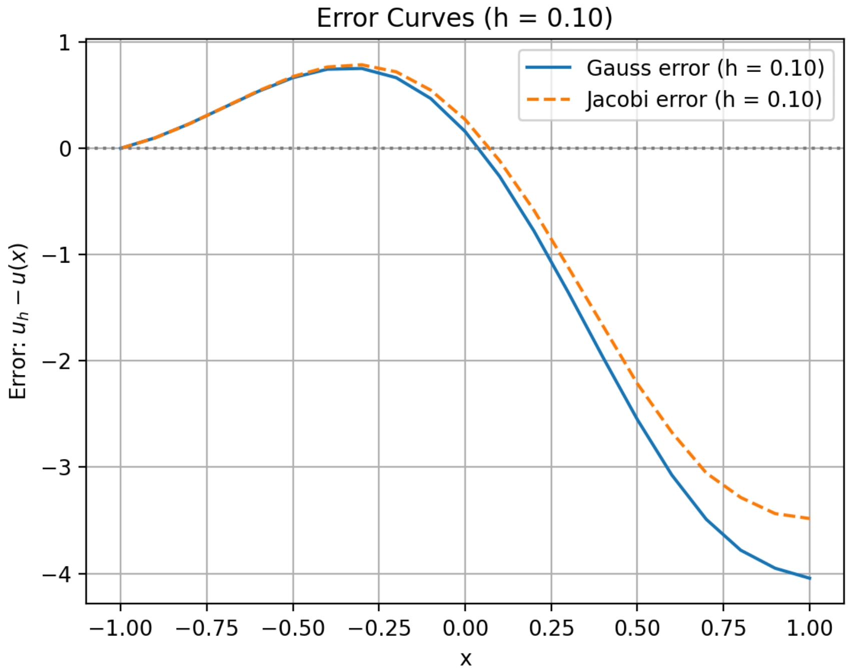

Solution Comparison (h = 0.05)
Exact
Gauss
Jacobi
u(x)
x
2
1
0
−1
−2
−3
−4
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

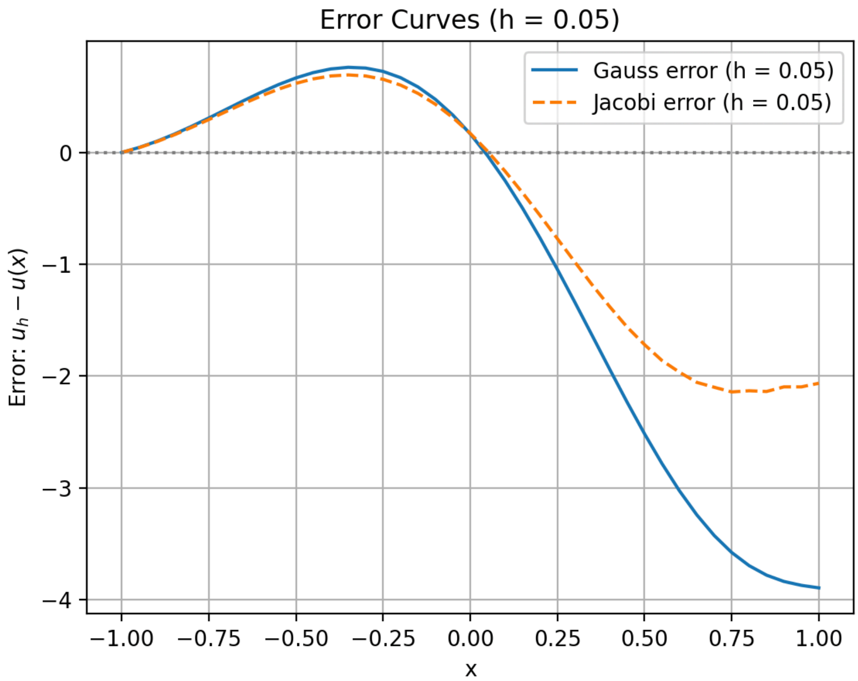


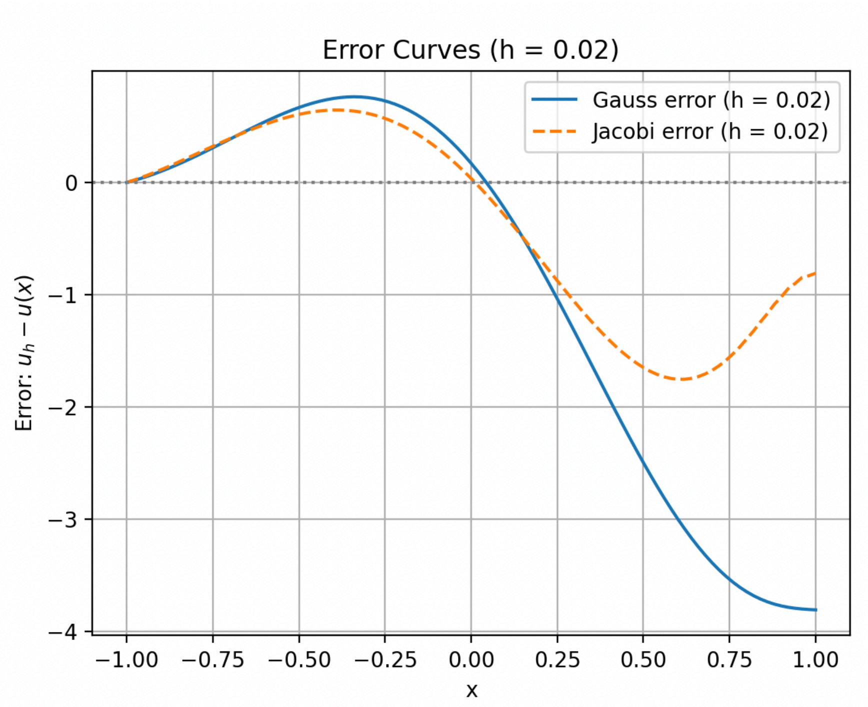
从图像中可以看出，随着步长 h 的减小，高斯消去法的数值解逐渐逼近解析解，表现出良好的收敛性。而雅可比迭代法在较大步长时（如 $h = 0.20$）效果尚可，但在步长减小后误差明显增大，尤其在 $h = 0.02$ 时几乎无法逼近真实解，可能是因为固定的迭代次数（30 次）不足以保证收敛，导致误差累积。此外，靠近区间右端（Neumann 边界）处误差更明显，也说明边界处理方式对精度有较大影响。可以看出，高斯消去法精度更稳定，而雅可比迭代法在迭代不足时会出现较大偏差。

任务2：




Error Curves (h = 0.10)
Gauss error (h = 0.10)
Jacobi error (h = 0.10)
Error: $u_h - u(x)$
x
1
0
−1
−2
−3
−4
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


Error Curves (h = 0.05)
Gauss error (h = 0.05)
Jacobi error (h = 0.05)
Error: $u_h - u(x)$
x
0
−1
−2
−3
−4
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00



从以上的图像中可以观察到，当步长为 $h = 0.20$ 时，两种方法的误差几乎完全重合，说明此时雅可比法在 30 次迭代内已经基本收敛，误差曲线也呈现出较好的对称和平滑性。随着步长减小至 $h = 0.10$，两种方法的误差逐渐分离，雅可比法的误差在右半区间开始偏离高斯消去法，当 h 继续减小至 0.05 和 0.02 时，偏差变得更加明显，雅可比迭代法误差曲线在靠近右端点区域出现了持续偏移甚至反向回升的震荡现象。这说明固定的迭代次数在较细网格下已经无法满足雅可比迭代法的收敛要求，导致残差未被有效压缩，误差逐点累积。与之相比，高斯消去法作为直接解法，在所有步长下均表现出良好的稳定性和精度，误差曲线光滑、收敛性良好，并随步长减小而持续接近真实解，符合中心差分格式的理论误差阶为 $O(h^2)$ 的预期。

特别的是，所有误差图都呈现出一定的非对称性，尤其在靠近区间右端 $x = 1$ 附近，误差幅度相对较大。这可能与边值问题中采用 Dirichlet–Neumann 混合边界条件有关，特别是 Neumann 条件的离散逼近会在边界处引入非对称误差传播机制，进而影响全局误差的分布形态。

四、总结

本次作业围绕非齐次两点边值问题，分别完成了差分方程的构建与两种数值方法的实现，并对不同步长下的解与误差行为进行了系统分析。在任务一中，采用了中心差分格式对微分算子进行离散，结合 Dirichlet–Neumann 边界条件，构建了三对角线性系统。通过高斯消去法与雅可比迭代法分别求解数值解，并与解析解进行对比，可以看出两种方法在较大步长下效果相近，而高斯消去法在整体精度与稳定性方面更具优势。

在任务二中进一步绘制了数值误差的变化曲线，清晰地展示了误差随步长减小而下降的收敛趋势。高斯消去法的误差始终较小且曲线平滑，而雅可比迭代法在步长减小时出现了误差积累和震荡，表明其在迭代次数不足时无法有效收敛。此外，边界条件处理对误差分布也有显著影响，尤其在 Neumann 边界附近误差较大。这次实验结果验证了差分格式的有效性，展示了不同数值方法在精度和收敛性方面的差异。