

# 现代偏微分方程计算方法课程作业

## 问题介绍

对于非齐次两点边值问题:

$$\begin{aligned} Lu &= -p(x)\frac{du(x)}{dx} + q(x)u(x) = f, \\ u(a) &= \alpha, u'(b) = \beta, x \in [a, b] \end{aligned} \quad (1)$$

其与以下优化问题等价:

$$\begin{aligned} J(u_*) &= \min_{\substack{u \in H^1 \\ u(a)=\alpha}} J(u), \\ J(u) &= \frac{1}{2}a(u, u) - (f, u) - p(b)\beta u(b), \\ a(u, v) &= \int_a^b \left( p(x)\frac{du(x)}{dx} \frac{dv(x)}{dx} + q(x)u(x)v(x) \right) dx. \end{aligned} \quad (2)$$

其中  $(u, v)$  表示两个函数的内积  $\int uv dx$ . 现给定边值条件  $[a, b] = [-1, 1], p(x) \equiv -(\pi^2 - 1)^{-1}, q(x) \equiv 1, \alpha = 0, \beta = -\pi e$ , 以及

$$f(x) = \frac{2\pi}{\pi^2 - 1} \cdot \cos(\pi x) \cdot e^x, \quad (3)$$

使用 Galerkin 方法在  $[a, b]$  的等长划分进行对  $u(x)$  进行求解, 其中基函数  $\varphi_i(x)$  定义为:

$$\begin{aligned} \varphi_0(x) &= \begin{cases} 1 - \xi, & x_0 < x < x_1, \\ 0, & \text{otherwise} \end{cases} \\ \varphi_i(x) &= \begin{cases} \xi, & x_{i-1} < x < x_i \\ 1 - \xi, & x_i < x < x_{i+1}, \\ 0, & \text{otherwise} \end{cases} \\ \varphi_{n(x)} &= \begin{cases} \xi, & x_{n-1} < x < x_n. \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (4)$$

在  $[a, b]$  上与之对应的待求解序列为  $\{u_1, \dots, u_n\}$ , 通过这  $n$  个点的值进行插值即可得到近似解  $u_h(x)$ :

$$u_h(x) = \sum_{i=1}^n u_i \varphi_i(x). \quad (5)$$

对相应的优化问题进行求导, 可以得到需求解的方程:

$$\begin{cases} \sum_{i=1}^n a(\varphi_i, \varphi_j) u_i = \int_a^b f \varphi_j dx, & j = 1 \dots n-1 \\ a(\varphi_{n-1}, \varphi_n) u_{n-1} + a(\varphi_n, \varphi_n) u_n = \int_a^b f \varphi_n dx + p(b)\beta, & j = n \end{cases} \quad (6)$$

在等式 4 的基函数设置下, 当  $|i - j| \geq 2$  时,  $\varphi_i(x) \cdot \varphi_j(x) = 0$ , 且  $\frac{d\varphi_i}{dx} \frac{d\varphi_j}{dx} = 0$ , 因此对于等式 6 中的双线性泛函构成的系数矩阵而言, 其每一行只有对角线及其左右两个位置上三个非零元素 (首行和末行则只有两个非零元素), 即:

$$\mathbf{A} = \begin{bmatrix} a(\varphi_1, \varphi_1) & a(\varphi_1, \varphi_2) & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \ddots & 0 \\ \vdots & a(\varphi_{j-1}, \varphi_j) & a(\varphi_j, \varphi_j) & a(\varphi_{j+1}, \varphi_j) & \vdots \\ 0 & \cdots & \cdots & a(\varphi_{n-1}, \varphi_n) & a(\varphi_n, \varphi_n) \end{bmatrix} \quad (7)$$

其中  $a(\cdot, \cdot)$  的计算可分为三类, 分别为:

$$\begin{aligned} a(\varphi_{j-1}, \varphi_j) &= \int_0^1 [-h^{-1}p(x_{j-1} + h\xi) + hq(x_{j-1} + h\xi)(1 - \xi)\xi] d\xi, \\ a(\varphi_j, \varphi_j) &= \int_0^1 [h^{-1}p(x_{j-1} + h\xi) + hq(x_{j-1} + h\xi)\xi^2] d\xi \\ &\quad + \int_0^1 [h^{-1}p(x_j + h\xi) + hq(x_j + h\xi)(1 - \xi)^2] d\xi, \\ a(\varphi_{j+1}, \varphi_j) &= \int_0^1 [-h^{-1}p(x_j + h\xi) + hq(x_j + h\xi)(1 - \xi)\xi] d\xi, \end{aligned} \quad (8)$$

特别地, 当  $j = n$  时,

$$a(\varphi_n, \varphi_n) = \int_0^1 [h^{-1}p(x_{n-1} + h\xi) + hq(x_{n-1} + h\xi)\xi^2] d\xi. \quad (9)$$

同样地, 由于基函数只在求解空间内某个小区间内取值, 因此等式 6 等号右边只在相应区间内具有非零有效值:

$$\begin{aligned} \mathbf{b}_j &= \int_a^b f\varphi_j dx = h \int_0^1 f(x_{j-1} + h\xi)\xi d\xi + h \int_a^b f(x_j + h\xi)(1 - \xi) d\xi, \\ \mathbf{b}_n &= \int_a^b f\varphi_n dx = h \int_0^1 f(x_{n-1} + h\xi)\xi d\xi + p(b)\beta, \end{aligned} \quad (10)$$

其中我们通过  $\xi$  将积分变量从  $[x_j, x_{j+1}]$  归一化到  $[0, 1]$  区间内. 而对于  $j = n$  处, 由于原始的两点边值问题具有非齐次条件, 因此还需要加上求导得到的非齐次项  $p(b)\beta$ .

通过对线性方程组  $\mathbf{Ax} = \mathbf{b}$  进行求解可得到每个基函数对应的系数  $u_j$ , 带入等式 5 中即可得到使用 Galerkin 法得到的对  $u(x)$  的近似估计.

## 数值求解实现

我们选择使用 Matlab 完成本次作业, 下面的代码展示了 Matlab 实现中用于求解系数矩阵  $\mathbf{A}$  和右端向量  $\mathbf{b}$  的主要部分: galerkin\_method 函数. 该函数接受 6 个参数, 含义分别为:

- `x`: 对求解区间  $[a, b]$  的均等划分, 通过 `a:h:b` 语句构造;
- `h`: 每个子区间的长度;
- `f`, `p`, `q`: 对应  $f(x)$ ,  $p(x)$ ,  $q(x)$  函数的实现;
- `beta`:  $\beta$  的值;

函数返回上述线性方程组  $Ax = b$  的系数矩阵  $A$  和右端向量  $b$ .

```
% Galerkin solution approach
function [coef, fphi] = galerkin_method(X, h, f, p, q, beta)
N = length(X) - 1;
% Calculate F
F = zeros(N, 1);
F(N) = h * integral(@(t) f(X(N)+h*t).*t, 0, 1) + p(X(N+1)) * beta;
for i = 1:N - 1
    F(i) = h * integral(@(t) f(X(i)+h*t).*t, 0, 1) + h * integral(@(t)
f(X(i+1)+h*t).*(1 - t), 0, 1);
end

% Calculate coefficient matrix
A = zeros(N, N);
for i = 1:N
    if i == 1
        A(1, 1) = integral(@(t) p(X(i)+h*t)/h+h*q(X(i)+h*t)*t.^2, 0, 1) + integral(@(t)
p(X(i+1)+h*t)/h+h*q(X(i+1)+h*t)*(1 - t).^2, 0, 1);
        A(1, 2) = integral(@(t) -p(X(i+1)+h*t)/h+h*q(X(i+1)+h*t)*t.*(1 - t), 0, 1);
    elseif i == N
        A(N, N-1) = integral(@(t) -p(X(i)/h+h*t)/h+h*q(X(i)+h*t)*(1 - t).*t, 0, 1);
        A(N, N) = integral(@(t) p(X(i)+h*t)/h+h*q(X(i)+h*t)*t.^2, 0, 1);
    else
        A(i, i-1) = integral(@(t) -p(X(i)/h+h*t)/h+h*q(X(i)+h*t)*(1 - t).*t, 0, 1);
        A(i, i) = integral(@(t) p(X(i)+h*t)/h+h*q(X(i)+h*t)*t.^2, 0, 1) + integral(@(t)
p(X(i+1)+h*t)/h+h*q(X(i+1)+h*t)*(1 - t).^2, 0, 1);
        A(i, i+1) = integral(@(t) -p(X(i+1)+h*t)/h+h*q(X(i+1)+h*t)*t.*(1 - t), 0,
1);
    end
end
coef = A;
fphi = F;
end
```

函数主要分为两部分: 第一个循环根据等式 10 通过 Matlab 内建的 `integral` 方法求出右端向量。第二个循环根据等式 8 和等式 9 依次对系数矩阵每一行中的非零元素进行计算, 其中的积分操作仍然通过 `integral` 方法进行计算。

在得到系数矩阵  $A$  和右端向量  $b$  后, 作业要求使用高斯消元法和雅克比迭代法两种方法对线性方程组进行求解。在 Matlab 中对高斯消元法的实现如下:

```
function sol = gaussian_solve(coef, fphi)
A = coef(:, :);
b = fphi(:);
R = rref([A, b]);
sol = R(:, end);
end
```

该实现首先将  $A$  和  $b$  合成为一个增广矩阵  $R$  ( $R$ ), 再对  $R$  进行变换得到简化的行阶梯形矩阵 (`rref`), 该矩阵最右端的列向量即为最终的解向量  $x$ 。

```

function sol = jacobi_solve(A, b, n)
D = diag(diag(A));
R = A - D;
% Check the astringency of Jacobi iteration
rho = max(abs(eig(D \ R)));
if rho >= 1
    fprintf("rho = %f, Jacobi iteration is not converged\n", rho);
end
sol = zeros(length(b), 1);
for i = 1:n
    sol = D \ (b - R * sol);
end
end

```

使用 Matlab 实现雅克比迭代法的代码如上所示. 首先将系数矩阵  $A$  拆分为对角矩阵  $D$  以及剩余元素  $R = A - D$ . `if` 分支判断了雅克比迭代收敛的充要条件是否成立: 即计算  $D^{-1}R$  的谱半径并判断其是否小于 1. `for` 循环则通过迭代计算

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{R}\mathbf{x}^{(k)}) \quad (11)$$

来逐步逼近方程的精确解.

## 实验结果

本节将分别展示使用高斯消元法和雅克比迭代法两种方法求解上述偏微分方程的近似解的情况. 由于给定参数的设置, 通过等式 8 和等式 9 计算得到的矩阵为非对角占优矩阵, 并且进一步的计算得到  $\rho(A) \geq 1$  (表 1), 因此使用雅克比迭代法得到的数值解并不能收敛.

### 使用 Gaussian 消元法求解

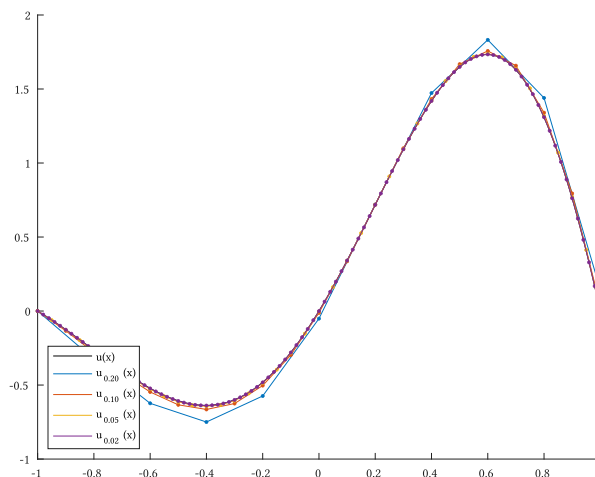


图 1:  $h = 0.2, 0.1, 0.05, 0.02$  情况下使用 Galerkin 方法求解出的近似  $u_h(x)$  图像.

图 1 展示了  $h$  在不同取值下对函数  $u(x) = \sin(\pi x) \cdot e^x$  的近似情况, 图中的离散点表示对应的基函数在采样点处的取值, 而连续的曲线则通过在采样点间对基函数  $\varphi_i(x)$  进行插值得到. 可以看到, 随着  $h$  的减小,  $u_h(x)$  逐渐变得光滑, 并且同解析解间的误差越小 (图 2).

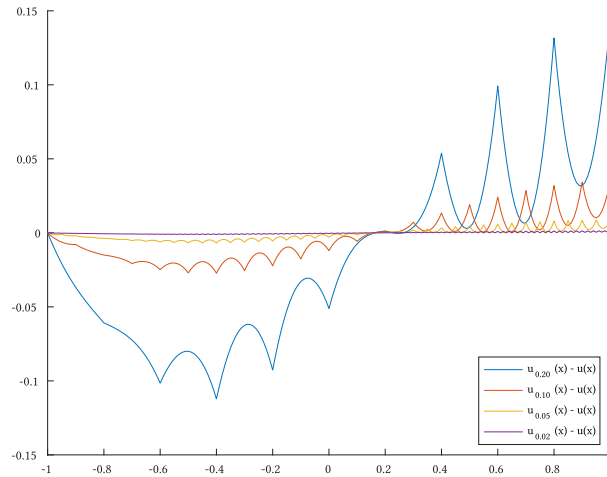


图 2:  $h = 0.2, 0.1, 0.05, 0.02$  时近似误差  $u_h(x) - u(x)$  的图像.

### 使用 Jacobi 迭代法求解

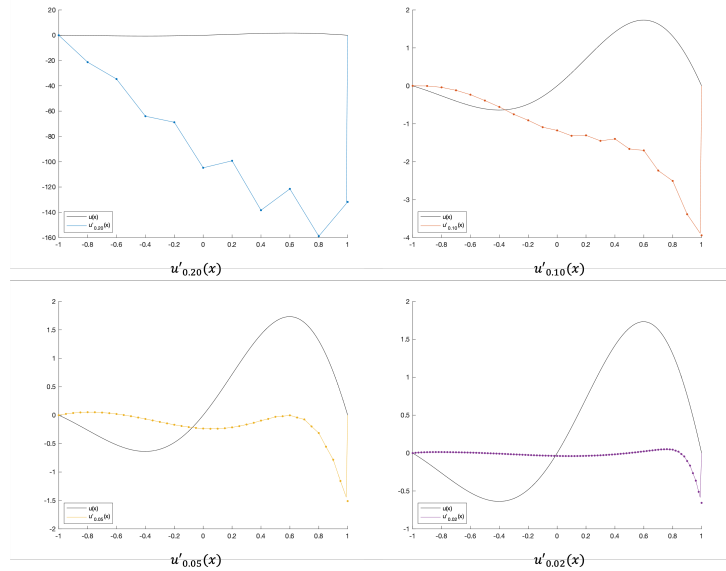


图 3:  $h = 0.2, 0.1, 0.05, 0.02$  时使用雅克比迭代法得到的  $u'_h(x)$  的图像.

图 3 展示了使用雅克比迭代法得到的  $u'_h(x)$  与解析解  $u'(x)$  的图像对比, 可以看出由于雅克比迭代法并不收敛, 所以得到的图像与解析解函数并不相似.

图 4 展示了使用雅克比迭代时得到的近似解  $u'_h(x)$  与解析解  $u'(x)$  之间的误差. 表 1 的第一行展示了系数矩阵  $A$  的谱半径, 可以看到  $h$  取不同值时  $\rho(A)$  均  $\geq 1$ , 因此可以判断雅克比迭代法无法收敛. 表 1 的 3, 4 行则展示了使用高斯消元法得到的  $u_h(x)$  与使用雅克比迭代法得到的  $u'_h(x)$  分别与解析解  $u'(x)$  之间的最大误差, 从中可以看出后者要明显大于前者.

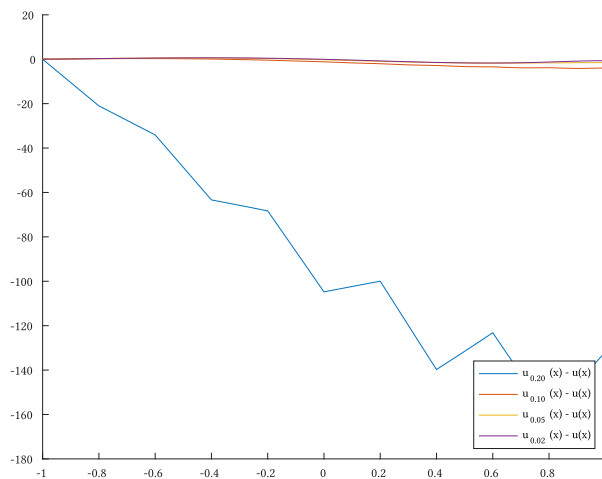


图 4:  $h = 0.2, 0.1, 0.05, 0.02$  时近似误差  $u'_h(x) - u(x)$  的图像.

$h$	0.2	0.1	0.05	0.02
$\rho(\mathbf{A})$	1.186396	1.042476	1.010390	1.001652
$\max( u_h(x) - u(x) )$	0.131650	0.034215	0.008646	0.001388
$\max( u'_h(x) - u(x) )$	160.204728	4.143427	1.749870	1.712174

表 1:  $h$  在不同取值下对应系数矩阵  $\mathbf{A}$  的谱半径, 以及高斯消元法和雅克比迭代法获得的近似解  $u_h(x), u'_h(x)$  与解析解  $u(x)$  之间的最大误差.