4 Spectral Methods

4.1 Legendre-Galerkin Spectral Methods

Example 4.1 Consider the two-point boundary value problem:

$$\begin{cases} -u''(x) + \alpha u(x) = f(x), \quad x \in I = (-1,1) \\ u(-1) = 0, u(1) = 0. \end{cases}$$

Weak formulation:

$$\begin{cases} \text{Find } u \in H^1(I) \text{ such that} \\ (u', v') + \alpha(u, v) = (f, v_N), \quad v \in H_0^1(I) \end{cases}$$

Let $\phi_k(x) = L_k(x) + a_k L_{k+1}(x) + b_k L_{k+2}(x)$ satisfies the boundary condition, we have $a_k = 0, b_k = -1$. Then,

$$\phi_k(x) = L_k(x) - L_{k+2}(x)$$

We denote

$$X_N = span\{\phi_k : k = 1, 2, \cdots, N-2\}$$

Spectral Scheme:

$$\begin{cases} \text{Find } u_N \in X_N \text{ such that} \\ (u_N', v_N') + (u_N, v_N) = (f, v_N), \quad v_N \in X_N \end{cases}$$

Given a set of basis functions $\{\phi_j\}_{j=0}^{N-2}$ of X_N

$$f_k = \int_I f_N \phi_k dx, \quad \mathbf{f} = (f_0, f_1, \ldots, f_{N-2})^T$$
$$u_N = \sum_{j=0}^{N-2} \hat{u}_j \phi_j, \quad \mathbf{u} = (\hat{u}_0, \hat{u}_1, \ldots, \hat{u}_{N-2})^T$$
$$s_{kj} = -\int_I \phi_j'' \phi_k dx, \quad m_{kj} = \int_I \phi_j \phi_k dx$$

and

$$S = (s_{kj})_{0 \le k,j \le N-2}, \quad M = (m_{kj})_{0 \le k,j \le N-2}$$

Taking $v_N = \phi_k$. The linear system

$$(S + \alpha M)\mathbf{u} = \mathbf{f}$$

The stiffness matrix $S = (s_{jk})$ is a diagonal matrix (P146-4.22):

$$s_{kk} = -(4k+6)b_k = 4k+6$$

The mass matrix $M = (m_{jk})$ is symmetric penta-diagonal (P146-4.23):

$$m_{jk} = m_{kj} = \begin{cases} \dfrac{2}{2k+1} + \dfrac{2}{(2k+5)}, & j = k \\ -\dfrac{2}{(2k+5)}, & j = k+2 \end{cases}$$

Note *An immediate consequence is that* $\{\phi_k\}_{k=0}^{N-2}$ *forms an orthogonal basis of* X_N *with respect to the inner product* $-(u_N'', v_N)$. *Furthermore, an orthonormal basis of* X_N *with respect to this inner product is*

$$\tilde{\phi}_k(x) := \frac{1}{\sqrt{-b_k(4k+6)}}\phi_k(x)$$

In the following Matlab codes, we choose $\tilde{\phi}_k(x)$ *as basis function.*

```matlab
% LegenSM1.m
% Legendre-Galerkin Method for for the model equation
% -u_xx+u=f in (-1,1) with boundary condition u(-1)=u(1)=0;
% exact solution: u=sin(kw*pi*x);
% RHS: f=kw*kw*pi^2*sin(kw*pi*x)+sin(kw*pi*x);
% Rmk: Use routines lepoly(); legs(); lepolym();
clear all
kw=10;
Nvec=[32:2:68];     % kw=10
%Nvec=[4:2:22]      % kw=1
Errv=[];            % Initialization for error
for N=Nvec
    [xv,wv]=legs(N);      % Legendre-Gauss points and weights
    Lm=lepolym(N+1,xv);   % Lm is a Legendre polynomal matrix
    u=sin(kw*pi*xv);              % test function
    f=kw*kw*pi^2*sin(kw*pi*xv)+sin(kw*pi*xv);      % Right-hand-side(RHS)
    % Calculting coefficient matrix
    S=eye(N);             % stiff matrix
    M=diag(1./(4*[0:N-1]+6))*diag(2./(2*[0:N-1]+1)+2./(2*[0:N-1]+5))
        -diag(2./(sqrt(4*[0:N-3]+6).*sqrt(4*[0:N-3]+14).*(2*[0:N-3]+5)),2)
        -diag(2./(sqrt(4*[2:N-1]-2).*sqrt(4*[2:N-1]+6).*(2*[2:N-1]+1)),-2);
        % mass matrix
    A=S+M;
    % Solving the linear system
    B=diag(1./sqrt(4*[0:N-1]+6))*(Lm(1:end-2,:)-Lm(3:end,:));
    b=B*diag(wv)*f;       % Solving RHS
    uh=A\b;               % expansion coefficients of u_N
    un=B'*uh;             % compositing the numerical solution

    error=norm(abs(un-u),inf);  % maximum pointwise error
    Errv=[Errv;error];
end
% Plot the maximum pointwise error
plot(Nvec,log10(Errv),'ro-','MarkerFaceColor','w','LineWidth',1.5)
grid on,
xlabel('N','fontsize', 14), ylabel('log10(Error)','fontsize',14)
title('Round-off error of Legendre-Galerkin methods','fontsize',12)
set(gca,'fontsize',12)
```

```
% LegenSM2.m
% Legendre-Galerkin Method for the model equation
% -u''(x)+u'(x)+u(x)=f(x), x in (-1,1),
% boundary condition: u(-1)=u(1)=0;
% exact solution: u=sin(kw*pi*xv);
% RHS: f=kw*kw*pi^2*sin(kw*pi*xv)+sin(kw*pi*xv);
% Rmk: Use routines lepoly(); legs(); lepolym();
clear all
kw=10;
Nvec=[32:2:68];
Errv=[];
for N=Nvec
    [xv,wv]=legs(N);        % Legendre-Gauss points and weights
    Lm=lepolym(N+1,xv);     % Lm is a Legendre polynomal matrix
    u=sin(kw*pi*xv);        % test function
    f=kw*kw*pi^2*sin(kw*pi*xv)+sin(kw*pi*xv)+kw*pi*cos(kw*pi*xv);  % RHS
    % Calculating coefficients matrix
    S=eye(N);       % stiffness matrix
    M=diag(1./(4*[0:N-1]+6))*diag(2./(2*[0:N-1]+1)+2./(2*[0:N-1]+5))
        -diag(2./(sqrt(4*[0:N-3]+6).*sqrt(4*[0:N-3]+14).*(2*[0:N-3]+5)),2)
        -diag(2./(sqrt(4*[2:N-1]-2).*sqrt(4*[2:N-1]+6).*(2*[2:N-1]+1)),-2);
        % mass matrix
    D=diag(1./(sqrt(2.*[0:N-2]+3).*sqrt(2.*[0:N-2]+5)),1)...
        +diag(-1./(sqrt(2.*[0:N-2]+3).*sqrt(2.*[0:N-2]+5)),-1);
        % matrix derived from u'(x)
    A=S+M+D;        % Coefficient matrix
    % Solving the linear system
    B=diag(1./sqrt(4*[0:N-1]+6))*(Lm(1:end-2,:)-Lm(3:end,:));
    b=B*diag(wv)*f;
    uh=A\b;         % expansion coefficients of u_N
    un=B'*uh;       % Coefficiets to points
    error=norm(abs(un-u),inf);  % maximum pointwise error
    Errv=[Errv;error];
end
% Plot the maximum pointwise error
plot(Nvec,log10(Errv),'mo-','MarkerFaceColor','w','LineWidth',1.5)
grid on, xlabel('N','fontsize', 14), ylabel('log10(Error)','fontsize',14)
title('Round-off error of Legendre-Galerkin methods','fontsize',12)
set(gca,'fontsize',12)
```

Example 4.2 Consider the two-point boundary value problem:

$$\begin{cases} -u''(y) + u(y) = f(y), \quad y \in \Lambda = [0,1] \\ u(0) = 1, u'(1) = 0. \end{cases}$$

Let $x \in I = [-1,1]$, $y = \frac{x}{2} + \frac{1}{2}$ and $U(x) = u(y) - 1$, the converted problem:

$$\begin{cases} -4U''(x) + U(x) = F(x), \quad x \in I = [-1,1] \\ U(-1) = 0, U'(1) = 0. \end{cases}$$

where $F(x) = f(2x-1) - 1$.

Weak formulation:

$$\begin{cases} \text{Find } U \in H^1(I) \text{ such that} \\ 4(U', v_N') + (U, v_N) = (f, v_N), \quad v_N \in H^1(I) \end{cases}$$

Let $\phi_k(x) = L_k(x) + a_k L_{k+1}(x) + b_k L_{k+2}(x)$ satisfies the boundary condition, we have

$$a_k = \frac{2k+3}{(k+2)^2}, \quad b_k = -\frac{(k+1)^2}{(k+2)^2}.$$

Let us denote

$$X_N = span\{\phi_k, k = 0, 1, \cdots, N-2\}$$

Spectral Scheme:

$$\begin{cases} \text{Find } U_N \in X_N \text{ such that} \\ 4(U_N', \phi') + (U_N, \phi_N) = (f, \phi), \quad \phi \in X_N \end{cases}$$

The stiffness matrix $S = (s_{jk})$ is a diagonal matrix (P146-4.22):

$$s_{kk} = -(4k+6)b_k = \frac{(4k+6)(k+1)^2}{(k+2)^2}$$

The mass matrix $M = (m_{jk})$ is symmetric penta-diagonal (P146-4.23):

$$m_{jk} = m_{kj} = \begin{cases} \frac{2}{2k+1} + \frac{2(2k+3)}{(k+2)^4} + \frac{2(k+1)^4}{(k+2)^4(2k+5)}, & j = k \\ \frac{2}{(k+2)^2} - \frac{2(k+1)^2}{(k+2)^2(k+3)^2}, & j = k+1 \\ -\frac{2(k+1)^2}{(k+2)^2(2k+5)}, & j = k+2 \end{cases}$$

```
% LegenSM3.m
% Legendre-Spectral Method for 1D elliptic problem
% -u_yy+u=f in [0,1] with boundary condition: u(0)=1,u'(1)=0;
% exact solution: u=(1-y)^2*exp(y);  RHS: f=(2-4*y)*exp(y);
% Converted : -4U_xx+U=F in [-1,1]
% boundary condition: U(-1)=0, U'(1)=0;
% exact solution: U=(1/2-1/2*x)^2*exp(1/2*x+1/2)-1;
% RHS:  F=-2*x*exp(1/2*x+1/2)-1.
clear all
Nvec=2:16;
Errv=[];  condnv=[];      % Initialization for error and condition number
for N=Nvec
    [xv,wv]=legs(N);      % xv and wv are Legendre-Gauss points and weights
    Lm=lepolym(N+1,xv);   % Lm is a Legendre polynomal matrix
    yv=1/2*(xv+1);                 % variable substitution
    U=(1-yv).^2.*exp(yv)-1;  % test function
    F=(2-4*yv).*exp(yv)-1;   % RHS in [0,1]
    % Calculting coefficient matrix
    e1=0:N-1;  e2=0:N-2;  e3=0:N-3;
    S=diag( (4*e1+6).*(e1+1).^2./(e1+2).^2 );     % stiff matrix
    M=diag(2./(2*e1+1)+2*(2*e1+3)./(e1+2).^4+2*((e1+1)./(e1+2)).^4./(2*e1+5))
        +diag( 2./(e2+2).^2-2*(e2+1).^2./((e2+2).^2.*(e2+3).^2) , 1 )
        +diag(2./(e2+2).^2-2*(e2+1).^2./((e2+2).^2.*(e2+3).^2),-1)
        +diag( -2*(e3+1).^2./((2*e3+5).*(e3+2).^2) , 2 )
        +diag(-2*(e3+1).^2./((2*e3+5).*(e3+2).^2),-2);  % mass matrix
    A=4*S+M;
    % Solving the linear system
    B=(Lm(1:end-2,:)+diag((2*e1+3)./(e1+2).^2)*Lm(2:end-1,:)...
        -diag((e1+1).^2./(e1+2).^2)*Lm(3:end,:));
    b=B*diag(wv)*F;        % Solving RHS
    Uh=A\b;                % expansion coefficients of u_N
    Un=B'*Uh;              % compositing the numerical solution
    error=norm(abs(Un-U),2);  % L^2 error
    Errv=[Errv;error];
    condnv=[condnv,cond(A)];  % condition number of A
end
% Plot the maximum pointwise error
plot(Nvec,log10(Errv),'go-','MarkerFaceColor','w','LineWidth',2)
grid on,
xlabel('N','fontsize', 14), ylabel('log_{10}(Error)','fontsize',14)
title('L^2 error of Legendre-Galerkin method','fontsize',12)
set(gca,'fontsize',12)
```

4.2 Collocation Methods

Example 4.3 The two-point boundary value problem:

$$\begin{cases} -u''(x) + \alpha u(x) = f(x), \quad x \in I = (-1, 1) \\ u(-1) = 0, u(1) = 0. \end{cases}$$

Exact solution: $u = \sin(k\pi x)$, $f = k^2\pi^2 \sin(k\pi x) + \alpha \sin(k\pi x)$.

```
% LegenCollo1.m
% Legendre-collocation method for the model equation:
% -u''(x)+\alpha u(x)=f(x), x in (-1,1);
% boundary condition: u(-1)=u(1)=0;
% exact solution: u=sin(kw*pi*x);
% RHS: f=kw*kw*pi^2*sin(kw*pi*x)+alpha*sin(kw*pi*x);
% Rmk: Use routines lepoly(); legslb(); legslbdm();
clear all
alpha=1;
kw=10;
N=32;
Nvec=[32:2:68];
Errv=[];
for N=Nvec
    [x,w]=legslb(N);            % compute LGL nodes and weights
    u=sin(kw*pi*x);             % test solution
    udprime=-kw*kw*pi*pi*sin(kw*pi*x);
    f=-udprime+alpha*u;     % RHS
    % Setup and solve the collocation system
    D1=legslbdm(N);         % 1st order differentiation matrices
    %D1=legslbdiff(N,x);    % 1st order differentiation matrices
    D2=D1*D1;                  % 2nd order differentiation matrices
    D=(-D2(2:N-1,2:N-1)+alpha*eye(N-2)); % coefficient matrix
    b=f(2:N-1);                  % RHS
    un=D\b;
    un=[0;un;0];                % Solve the system

    error=norm(abs(un-u),inf);  % maximum pointwise error
    Errv=[Errv;error];
end;
plot(Nvec,log10(Errv),'rd-','MarkerFaceColor','w','LineWidth',1.5)
grid on, xlabel('N','fontsize', 14), ylabel('log10(Error)','fontsize',14)
title('Convergence of Legendre-collocation method','fontsize',12)
set(gca,'fontsize',12)
```

```matlab
% LegenCollo2.m
% Legendre-collocation Method for the model equation:
% -u''(x)+u'(x)+u(x)=f(x),  x in (-1,1);
% % boundary condition: u(-1)=u(1)=0;
% exact solution: u=sin(kw*pi*x);
% RHS: f(x)=kw^2*pi^2*sin(kw*pi*x)+sin(kw*pi*x)+kw*pi*cos(kw*pi*x);
% Rmk: Use routines lepoly(); legslb(); legslbdm();
clear all
kw=10;
Nv=[32:2:68];
Errv=[];
for N=Nv
    [xv,wv]=legslb(N);           % compute LGL nodes and weights
    u=sin(kw*pi*xv);             % test function
    f=kw*kw*pi^2*sin(kw*pi*xv)+sin(kw*pi*xv)+kw*pi*cos(kw*pi*xv);  % RHS
    % Setup and solve the collocation system
    D1=legslbdm(N);     % 1st order differentiation matrix
    D2=D1*D1;           % 2nd order differentiation matrix
    D=-D2(2:N-1,2:N-1)+D1(2:N-1,2:N-1)+eye(N-2);    % coefficient matrix
    b=f(2:N-1);    % RHS
    un=D\b;
    un=[0;un;0];  % Solve the system

    error=norm(abs(un-u),inf);
    Errv=[Errv;error];
end
% Plot the maximum pointwise error
plot(Nv,log10(Errv),'md-','MarkerFaceColor','w','LineWidth',1.5)
grid on,
xlabel('N','fontsize', 14), ylabel('log10(Error)','fontsize',14)
title('Convergence of Legendre-collocation method','fontsize',12)
```

Example 4.4 The two-point boundary value problem:

$$\begin{cases} -u''(y) + u(y) = f(y), \quad y \in \Lambda = [0,1] \\ u(0) = 1, u'(1) = 0. \end{cases}$$

Exact solution: $u(y) = (1-y)^2 \exp(y)$, $f(y) = (2-4y)\exp(y)$.

```
% LegenCollo3.m
% Legendre-collocation Method for the model equation:
% -u''(y)+u(y)=f(y) in [0,1] with boundary condition: u(0)=1, u'(1)=0;
% test function : u(y)=(1-y)^2*exp(y);
% RHS : f(y)=(2-4*y)*exp(y);
% Rmk: Use routines legslb(); legslbdiff();
clear all
Nvec=4:18;
Errv=[]; condnv=[];
for N=Nvec
    xv=legslb(N);            % compute LGL nodes and weights
    yv=1/2*(xv+1);           % variable substitution
    u=(1-yv).^2.*exp(yv);    % test solution in [0,1]
    f=(2-4*yv).*exp(yv);     % RHS in [0,1]

    % Setup and solve the collocation system
    D1=legslbdiff(N,xv);     % 1st order differentiation matrices
    D2=D1*D1;                % 2nd order differentiation matrices
    D=-4*D2+eye(N);          % coefficient matrix
    D(1,:)=[1,zeros(1,N-1)];  D(N,:)=D1(N,:);
    b=[1; f(2:N-1); 0];        % RHS
    un=D\b;                    % Solve the system

    error=norm(abs(un-u),2);  % L^2 error
    Errv=[Errv;error];
    condnv=[condnv,cond(D)];
end
% Plot the L^2 error
plot(Nvec,log10(Errv),'gd-','MarkerFaceColor','w','LineWidth',1.5)
grid on, xlabel('N','fontsize', 14), ...
    ylabel('log_{10}(Error)','fontsize',14),
title('L^2 error of Legendre-collocation method','fontsize',12)
```