AE 370: HW7

Linyi Hou

April 16, 2020

Problem 1

a. Given the ODE

$$-\frac{d^2u}{dx^2} + 2u = f, \quad 0 < x < 3 \tag{1}$$

we know that the energy equation must be expressible as

$$-\int_a^b u''(x)\phi_j(x)dx + 2\int_a^b u(x)\phi_j(x)dx = (\hat{u}, \phi_j)_E \tag{2}$$

since it is, by definition of the ODE, equivalent to

$$\int_a^b f(x)\phi_j(x)dx. \tag{3}$$

Now, integrate by parts for the first term in Equation (2):

$$-\left[u'(x)\phi_j(x)\right]_a^b + \int_a^b u'(x)\phi_j'(x)dx + 2\int_a^b u(x)\phi_j(x)dx = (\hat{u}, \phi_j)_E \tag{4}$$

finally, get

$$\int_a^b u'(x)\phi_j'(x)dx + 2\int_a^b u(x)\phi_j(x)dx = (\hat{u}, \phi_j)_E \tag{5}$$

b. From the definition provided by class notes,

$$\begin{aligned} \mathcal{V}_n^L = \{g(x)\ &: g(a) = g(b) = 0, \\ &g(x) = a_i x + d_i \text{ for } a_i, d_i \in \mathbb{R},\ x \in [x_{i-1}, x_i],\ i = 2, \dots, n+1\} \end{aligned} \tag{6}$$

In words, this means that $\mathcal{V}_n^L$ is defined by a set of functions that satisfy the boundary conditions $g(a) = g(b) = 0$ (where $[a, b] = [0, 3]$ in the context of this problem), and each function could be written as a line over each sub-interval $[x_{i-1}, x_i]$.

c. Using the definition of the energy inner product, we have

$$(u, \phi_j)_E = (\hat{u}, \phi_j)_E \tag{7}$$

where $\hat{u}$ is the approximated solution and u is the true solution to the ODE.

Choice of basis: Let the basis functions $\{\phi_2, .., \phi_n\}$ satisfy the following:

$$q(x) = \sum_{j=2}^{n} q(x_j)\phi_j(x) \tag{8}$$

where q is a function in $\mathcal{V}_n^L$. It is convenient to define

$$\phi_j(x_i) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{9}$$

We find that the basis functions that satisfy Equation (9) are:

$$\phi_i(x) = \begin{cases} \frac{1}{\Delta x}[x - a - (i-1)\Delta x] & x \in [x_{i-1}, x_i] \\ -\frac{1}{\Delta x}[x - a - (i+1)\Delta x] & x \in [x_i, x_{i+1}] \\ 0 & else \end{cases} \tag{10}$$

and its derivatives

$$\phi_i'(x) = \begin{cases} \frac{1}{\Delta x} & x \in [x_{i-1}, x_i] \\ -\frac{1}{\Delta x} & x \in [x_i, x_{i+1}] \\ 0 & else \end{cases} \tag{11}$$

Energy Inner Product: It follows that the only non-zero energy inner products are the following (computed using MATLAB, see Appendix C):

$$(\phi_{i-1}, \phi_i)_E = \frac{\Delta x}{3} - \frac{1}{\Delta x} \tag{12}$$

$$(\phi_i, \phi_i)_E = \frac{4\Delta x}{3} + \frac{2}{\Delta x} \tag{13}$$

$$(\phi_i, \phi_{i+1})_E = \frac{\Delta x}{3} - \frac{1}{\Delta x} \tag{14}$$

Matrix form: From Equation (7), we have

$$\left(\sum_{i=2}^{n} u_i\phi_i, \phi_j\right)_E = (u, \phi_j)_E \tag{15}$$

using properties of the inner product:

$$\sum_{i=2}^{n} u_i(\phi_i, \phi_j)_E = (u, \phi_j)_E = (f, \phi_i)_s \tag{16}$$

We can now rewrite Equation (16) in matrix form:

$$\begin{bmatrix} (\phi_2,\phi_2)_E & (\phi_3,\phi_2)_E & \dots & (\phi_{n-1},\phi_2)_E & (\phi_n,\phi_2)_E \\ (\phi_2,\phi_3)_E & (\phi_3,\phi_3)_E & \dots & (\phi_{n-1},\phi_3)_E & (\phi_n,\phi_3)_E \\ \vdots & \vdots & \dots & \vdots & \vdots \\ (\phi_2,\phi_{n-1})_E & (\phi_3,\phi_{n-1})_E & \dots & (\phi_{n-1},\phi_{n-1})_E & (\phi_n,\phi_{n-1})_E \\ (\phi_2,\phi_n)_E & (\phi_3,\phi_n)_E & \dots & (\phi_{n-1},\phi_n)_E & (\phi_n,\phi_n)_E \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} -(f,\phi_2)_s \\ -(f,\phi_3)_s \\ \vdots \\ -(f,\phi_{n-1})_s \\ -(f,\phi_n)_s \end{bmatrix} \tag{17}$$

Substituting for actual values:

$$\left(\frac{\Delta x}{3} \begin{bmatrix} 4 & 1 & \dots & 0 & 0 \\ 1 & 4 & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 4 & 1 \\ 0 & 0 & \dots & 1 & 4 \end{bmatrix} + \frac{1}{\Delta x} \begin{bmatrix} 2 & -1 & \dots & 0 & 0 \\ -1 & 2 & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 2 & -1 \\ 0 & 0 & \dots & -1 & 2 \end{bmatrix} \right) \begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} -(f,\phi_2)_s \\ -(f,\phi_3)_s \\ \vdots \\ -(f,\phi_{n-1})_s \\ -(f,\phi_n)_s \end{bmatrix} \tag{18}$$

where

$$(f,\phi_i)_s = \int_{a+(i-2)\Delta x}^{a+i\Delta x} f(x)\phi_i(x)dx \tag{19}$$

d. Figures and code have been attached in Appendices A and B, respectively.

Appendix A: Figures

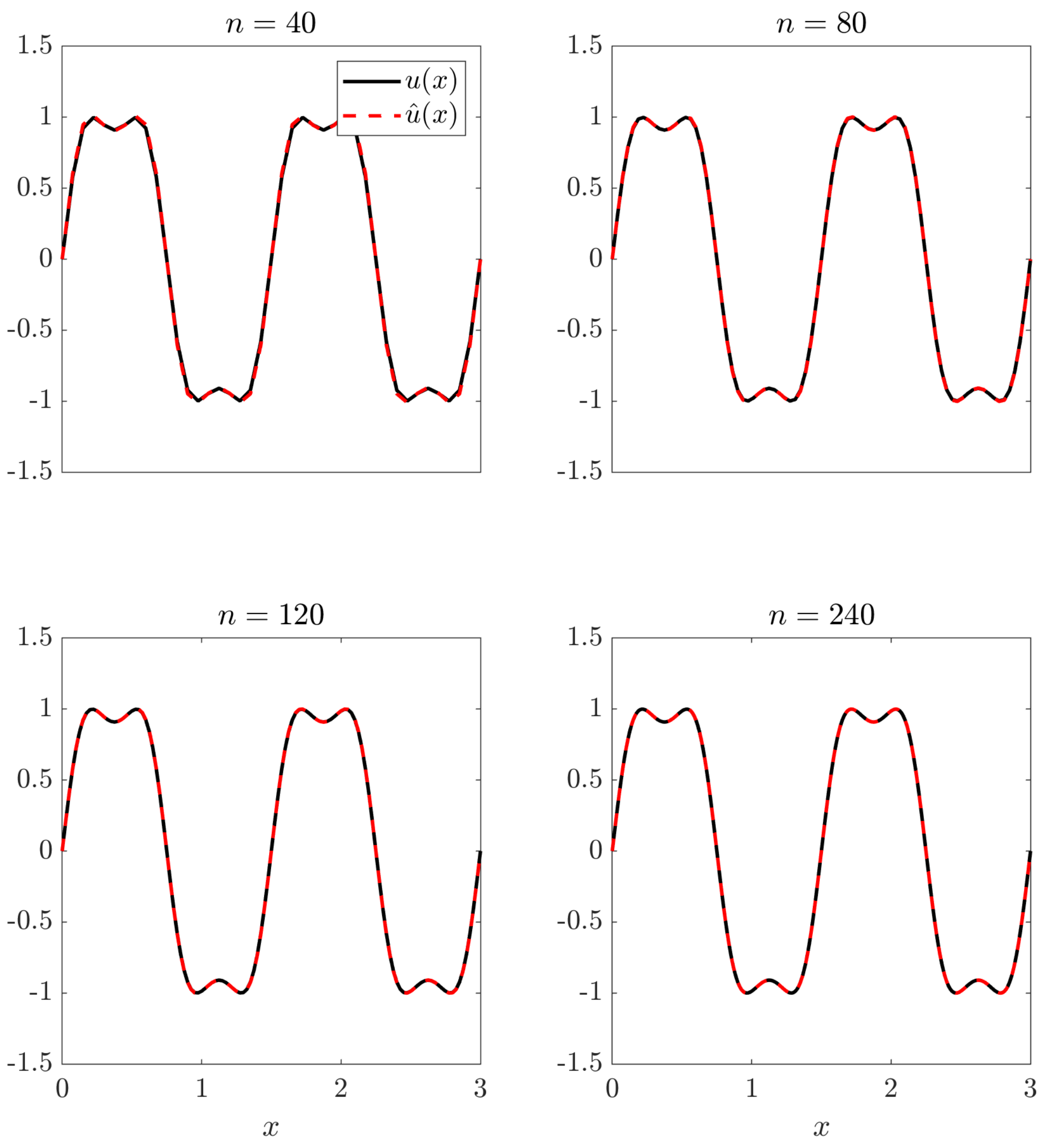

Figure 1: Problem 1 Function Plot

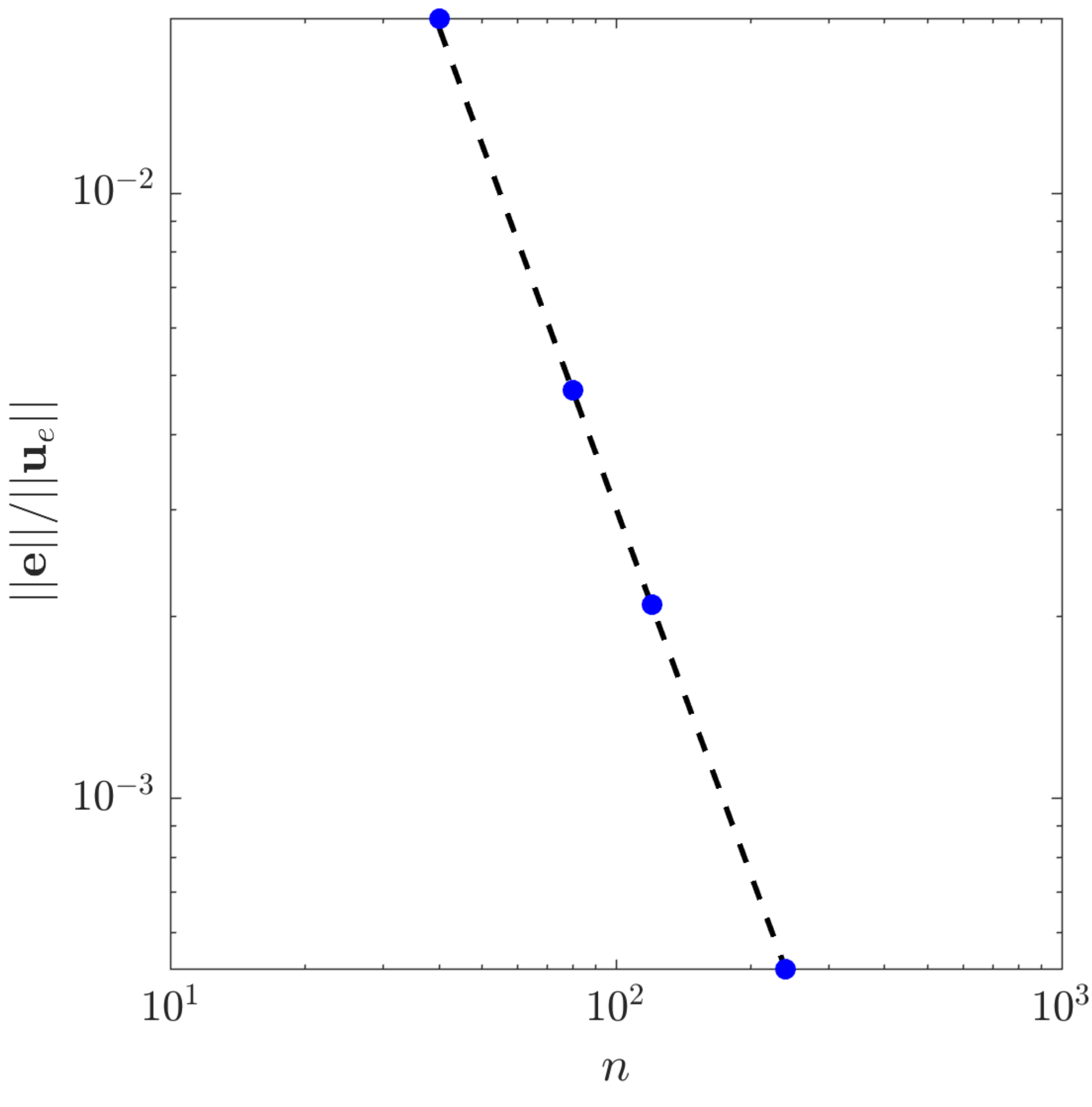

Figure 2: Problem 1 Error Plot

Appendix B: Code

```
%% Problem 1

close all
clear;clc

%solve -u'' + 2*u = f over 0 < x < 1
%with BCs u(0) = u(3) = 0;

%left and right bounds
xl = 0; xr = 3;
ln = xr - xl;

%exact sol
uex = @(x) sin( 2* sin(4*pi*(x-xl)/ln) );


%f(x) =
fcn = @(x) (32*pi^2*cos(2*sin((4*pi*(x-xl))/ln)).*sin((4*pi*(x-xl))/ln))/
    ln^2 ...
    +(64*pi^2*sin(2*sin((4*pi*(x-xl))/ln)).*cos((4*pi*(x-xl))/ln).^2)/ln
        ^2 + ...
    2*(sin( 2* sin(4*pi*(x-xl)/ln) ));


%# of n points to use
nvect = [40; 80; 120; 240];

%initialize error vect
err = zeros( size( nvect ) );

for j = 1 : length(nvect)

    n = nvect( j ) ;
    dx = ( xr - xl ) / n;

    %define grid (including boundary pts)
    xx = linspace(xl,xr,n+1)';

```

```matlab
%initialize soln vector and rhs vector
u = zeros( n-1,1 );
b = u;

%Build G (matrix associated with (10) from partial solutions)
G = 1*dx/3 * ( 4*eye(n-1) + diag(ones(n-2,1),1) + diag(ones(n-2,1),-1) ) ...
    + 1/dx * ( 2*eye(n-1) - diag(ones(n-2,1),1) - diag(ones(n-2,1),-1) );

%Build RHS vector, b, by looping through each element
for jj = 1 : n-1


    %--RHS (be very careful with indexing here!)

        %define x from j-1 to j
        x_lft = (xl+(jj-1)*dx):dx:(xl+(jj)*dx);
        %define phij from j-1 to j (upward sloping part of phij)
        phij_lft = 1/dx * ( x_lft-xl-(jj-1)*dx );
        %inner product contribution from j-1 to j
            %can use Matlab's trapz for simplicity, but other quadrature
            %rules are OK too!
        b(jj) = trapz( x_lft, fcn( x_lft ).* phij_lft );

        %define x from j to j+1 (downward sloping part of phij)
        x_rgt = (xl+(jj)*dx):dx:(xl+(jj+1)*dx);
        %phij from j to j+1
        phij_rgt = -1/dx * ( x_rgt-xl-(jj+1)*dx );
        %inner product contribution from j-1 to j
        b(jj) = b(jj) + trapz( x_rgt, fcn( x_rgt ).* phij_rgt );

    %--

end

%Solve for uhat
uhat = G\b;

```

```
%Add BCs to ua
uhat = [uex(xl); uhat; uex(xr)];

%--plot soln
    figure(1),subplot(2,2,j)
    plot( xx, uex(xx), 'k-', 'linewidth', 2 ), hold on
    plot( xx, uhat, 'r--', 'linewidth', 2 ), hold on

    %make plot pretty
    title( ['$n = ', num2str( n ),'$'] ,'interpreter', 'latex',...
        'fontsize', 16)
    if j == 1
        h = legend( '$u(x)$', '$\hat{u}(x)$');
    end

    if j <= 2
        set( gca, 'XTick', [] )
    else
        xlabel( '$x$', 'interpreter', 'latex', 'fontsize', 16)

    end
    set(h, 'location', 'NorthEast', 'Interpreter', 'Latex', 'fontsize
       ', 16 )
    set(gca, 'TickLabelInterpreter','latex', 'fontsize', 16 )
    axis([0 3 -1.5 1.5])

    set(gcf, 'PaperPositionMode', 'manual')
    set(gcf, 'Color', [1 1 1])
    set(gca, 'Color', [1 1 1])
    set(gcf, 'PaperUnits', 'centimeters')
    set(gcf, 'PaperSize', [25 25])
    set(gcf, 'Units', 'centimeters' )
    set(gcf, 'Position', [0 0 25 25])
    set(gcf, 'PaperPosition', [0 0 25 25])
%--

%error
err( j ) = norm( uhat - uex(xx) ) / norm( uex(xx) );
```

```
end

svnm = 'q1_plot';
print( '-dpng', svnm, '-r200' )

%plot err
figure(100)

%plot dx^2 line to show err scales correctly
c = err(end)/(dx^2);
loglog( nvect, c*(ln./nvect).^2, 'k--', 'linewidth', 2 ), hold on

%plot err
loglog( nvect, err , 'b.', 'markersize', 26 )
xlim([10 1000])

%make pretty
xlabel( '$n$', 'interpreter', 'latex', 'fontsize', 16)
ylabel( '$||\textbf{e}||/||\textbf{u}_e||$ ', 'interpreter', 'latex', '
    fontsize', 16)

set(gca, 'TickLabelInterpreter','latex', 'fontsize', 16 )

set(gcf, 'PaperPositionMode', 'manual')
set(gcf, 'Color', [1 1 1])
set(gca, 'Color', [1 1 1])
set(gcf, 'PaperUnits', 'centimeters')
set(gcf, 'PaperSize', [15 15])
set(gcf, 'Units', 'centimeters' )
set(gcf, 'Position', [0 0 15 15])
set(gcf, 'PaperPosition', [0 0 15 15])

svnm = 'q1_error';
print( '-dpng', svnm, '-r200' )
```

Appendix C: Energy Inner Product Derivation Code

```
close all
clear;clc

syms a ii x dx real

%% phi(i-1), phi(i)

phi_m1 = -1/dx*(x-a-ii*dx);
phi =     1/dx*(x-a-(ii-1)*dx);

lb = a+(ii-1)*dx;
ub = a+ii*dx;

IP =     int( diff(phi_m1,x)*diff(phi,x), x, lb, ub ) ...
   + 2 * int( phi_m1*phi                , x, lb, ub )

%% phi(i), phi(i)

phi_l =  1/dx*(x-a-(ii-1)*dx);
phi_u = -1/dx*(x-a-(ii+1)*dx);

lb = a+(ii-1)*dx;
mb = a+ii*dx;
ub = a+(ii+1)*dx;

IP =     int( diff(phi_l,x)*diff(phi_l,x), x, lb, mb ) ...
   + 2 * int( phi_l*phi_l                , x, lb, mb ) ...
   +     int( diff(phi_u,x)*diff(phi_u,x), x, mb, ub ) ...
   + 2 * int( phi_u*phi_u                , x, mb, ub )

%% phi(i), phi(i+1)

phi =    -1/dx*(x-a-(ii+1)*dx);
phi_p1 =  1/dx*(x-a-ii*dx);

lb = a+ii*dx;
ub = a+(ii+1)*dx;

```

```
IP =     int( diff(phi,x)*diff(phi_p1,x), x, lb, ub ) ...
   + 2 * int( phi*phi_p1                , x, lb, ub )
```