UNIVERSITY OF NOTRE DAME
Department of Civil Engineering
and Geological Sciences

CE 60130 Finite Elements in Engineering
J.J. Westerink

Homework Set #1 KEY

Problem 1

Consider the following ordinary differential equation (ODE) of $u(x)$:

$$\frac{d^2u}{dx^2} + x\frac{du}{dx} + \frac{u}{6} = -3 \qquad \text{for the domain } 0 \leq x \leq 10$$

with boundary conditions

$$u|_{x=0} = 5$$
$$\left.\frac{du}{dx}\right|_{x=10} = -\frac{\pi}{5}$$

Solve this ODE using the method of weigthed residuals on the domain $0 \leq x \leq 10$ using polynomial basis functions. Recall that the approximate solution follows

$$u_{app} = u_B + \sum_{i=1}^{N} \alpha_i \phi_i$$

where u_B is the boundary function that satisfies the boundary conditions, α_i are the unknown coefficients which must be determined and ϕ_i are the known basis functions that form a complete set.

a) *Formulate the boundary function, u_B, such that the boundary conditions are satisfied.*

Solution:

To begin we find the appropriate boundary function u_B using the polynomial expansion x^n. We have that $u_B(0) = 5$ and $u'_B(10) = -\pi/5$. Considering the boundary function $u_B = a_1 + a_2x^n$ we have

$$u_b(0) = a_1$$
$$\Longrightarrow$$
$$a_1 = 5$$

$$u'_B(10) = a_2 n 10^{n-1}$$
$$\Longrightarrow$$
$$a_2 n 10^{n-1} = -\frac{\pi}{5}$$

Let $n = 1$ and $a_2 = -\pi/5$

$$-\frac{\pi}{5} = -\frac{\pi}{5}$$

Thus the boundary function is

$$\boxed{u_B = 5 - \frac{\pi}{5}x}$$

b) *Determine the form of the basis functions, ϕ_i, such that the admissibility and completeness criteria are strictly satisfied.*

Solution:

Now we focus on generating approximating functions that are homogeneous on the boundaries, $\phi_i(0) = 0$ and $\phi'_i(10) = 0$, such that $\phi_i(x) = x^n$, (note: n in this formulation is not necessarily an integer). At $x = 0$ we have $0 = 0$, already satisfied. However at $x = 10$ we have

$$\phi'_i(10) = n10^{n-1} = 0$$
$$\Longrightarrow$$
$$\cancel{n = 0}$$

This leads to a contradiction. As a result we must reconsider our choice for basis functions. Now consider a set of basis functions of the form $\phi_i(x) = (n - x)x^i$, note that given that the boundary function u_B is linear in order to have linear independence amongst the approximation functions the lowest order approximation function must be quadratic. At $x = 0$ we have $(n - 0) \cdot 0 = 0$, already satisfied. However at $x = 10$ we have

$$\phi'_i(10) = (i)(n - 10)10^{i-1} - 10^i = 0$$
$$\Longrightarrow$$
$$i(n - 10)10^{i-1} = 10^i$$
$$i(n - 10) = 10$$
$$n = \frac{10}{i} + 10$$
$$n = \frac{10(i + 1)}{i}$$

Thus the set of approximation functions are

$$\boxed{\phi_i(x) = \left(\frac{10(i + 1)}{i} - x\right) x^i}$$

c) *Develop formulae for the entries in the system of simultaneous equations and the right hand side vector to solve using the Collocation, Least Squares and Galerkin methods.*

Solution:

We have the approximation function

$$u_{app} = 5 - \frac{\pi}{5}x + \sum_{i=1}^{N} \alpha_i \left(\frac{10(i+1)}{i} - x \right) x^i$$

For each method we consider the interior error, $\epsilon_I = \langle L(u_{app} + 3, w_j \rangle$ and set that to zero

$$\begin{aligned}
\langle L(u_{app}) + 3, w_j \rangle =& \int_0^{10} \left[\frac{d^2 u_{app}}{dx^2} + x\frac{du_{app}}{dx} + \frac{u_{app}}{6} + 3 \right] w_j dx \\
=& \int_0^{10} \left\{ \sum_{i=1}^{N} \alpha_i \left[i(i-1) \left(\frac{10(i+1)}{i} - x \right) x^{i-2} - 2ix^{i-1} \right. \right. \\
& \left. \left. + x \left(i \left(\frac{10(i+1)}{i} - x \right) x^{i-1} - x^i \right) + \frac{1}{6} \left(\frac{10(i+1)}{i} - x \right) x^i \right] - \frac{7\pi}{30}x + \frac{23}{6} \right\} w_j dx \\
=& 0
\end{aligned}$$

Rearranging terms produces the general expression

$$\begin{aligned}
\sum_{i=1}^{N} \alpha_i \int_0^{10} & \left[i(i-1) \left(\frac{10(i+1)}{i} - x \right) x^{i-2} - 2ix^{i-1} + x \left(i \left(\frac{10(i+1)}{i} - x \right) x^{i-1} - x^i \right) \right. \\
& \left. + \frac{1}{6} \left(\frac{10(i+1)}{i} - x \right) x^i \right] w_j dx = \int_0^{10} \left(\frac{7\pi}{30}x - \frac{23}{6} \right) w_j dx
\end{aligned}$$

This produces a system of equations that can be solved for the coefficients α_i

$$\underline{\underline{A}} \cdot \underline{\alpha} = \underline{b}$$

where

$$A_{j,i} = \int_0^{10} \left[(\Upsilon_i - x) \left(i(i-1) + \frac{1}{6}x^2 + ix^2 \right) - 2ix - x^3 \right] x^{i-2} w_j dx$$
$$b_j = \int_0^{10} \left(\frac{7\pi}{30}x - \frac{23}{6} \right) w_j dx$$
$$\Upsilon_i = \frac{10(i+1)}{i}$$

The weighting functions are determined by the method employed

Collocation:

$$w_j = \delta(x - x_j)$$

Least Squares:

$$\begin{aligned} w_j &= L(\phi_j) \\ &= L\left((\Upsilon_j - x)\, x^{j+1} \right) \\ &= \left[(\Upsilon_j - x) \left(j(j-1) + \frac{1}{6}x^2 + jx^2 \right) - 2jx - x^3 \right] x^{j-2} \end{aligned}$$

Galerkin:

$$\begin{aligned} w_j &= \phi_j \\ &= (\Upsilon_j - x)\, x^j \end{aligned}$$

Note, that for this formulation we need to take special consideration for the $i = 1$ case, where the second derivative of the basis function $\phi_1''(x) = -2$. For detailed solutions to the integrations see the provided Mathematica workbook.

d) *Using Matlab, or other suitable language, develop a code to solve each system (Collocation, Least Squares, Galerkin) for 2, 10, 20 and 100 term expansions. Plot the solution for each expansion set. Determine the error for each solution by comparing with the point based analytical solution provided on the course website (Problem_1_exact_soln.txt). Compute the L_2 error for each expansion and plot these errors on a log–log plot with the number of terms as the horizontal axis. In addition plot the pointwise error over the entire domain using a logarithmic scale for the y-axis..*

Solution:

Using Mathematica (or other analytic math program) it is possible to set up the entries for all the $A_{i,j}$'s and b_j's. See the Mathematica notebook file Problem_1_work.nb for more details about what these entries look like. Figures **??**–**??** show the results for each method for 2, 10, 20 and 100 term expansions.

Figure **??** shows the results for the log–log plot of the L_2 errors for each method and number of expansion terms.

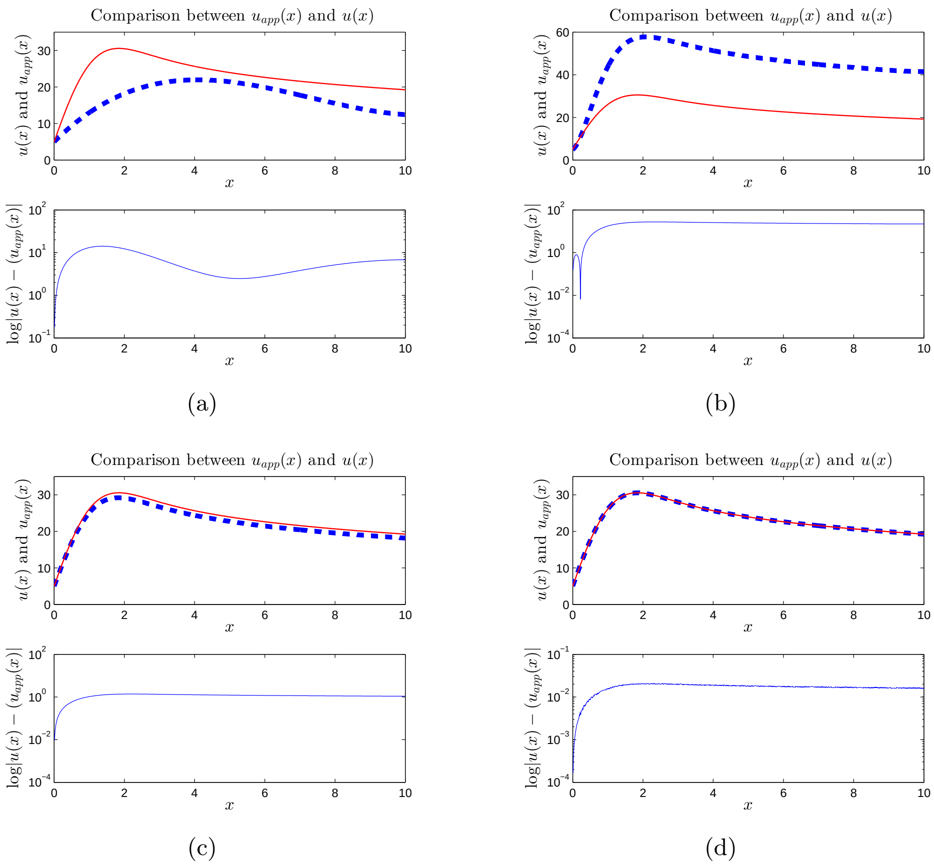


Figure 1: Collocation Method: (A) 2-Term, (B) 10-Term, (C) 20-Term, (D) 100-Term

e) *Comment on each method used. Discuss properties such as convergence, accuracy, computational cost. Inter-compare the solutions from each method.*

Solution:

All three methods appear to converge as the number of terms increases. The rate of convergence for the Galerkin and Collocation methods appears to be slightly higher then for the Least Squares method. However, the Least Squares method is consistently more accurate than the other two methods for expansions of 10 or more terms. For the $N = 2$ case all three methods produce solutions that appear more accurate in the L_2 measure than the $N = 10$ case, but in structure and shape the solutions are not accurate. In computational cost there is not much difference between the methods. However, when discussing the initial set-up the collocation and Galerkin methods are much simpler then the Least Squares method. The general $A_{i,j}$ entries for the collocation method are the most simple, followed by the Galerkin method, with the Least Squares method having the most complicated entries.

For higher order expansions (large N) the three methods produce very similar solutions. Although the Least Squares is the most accurate, the three methods produce solutions that capture the structure of the exact solution very well. Considering the rate of convergence from figure **??** we would expect the errors for the Collocation and Galerkin methods to eventually be comparable to

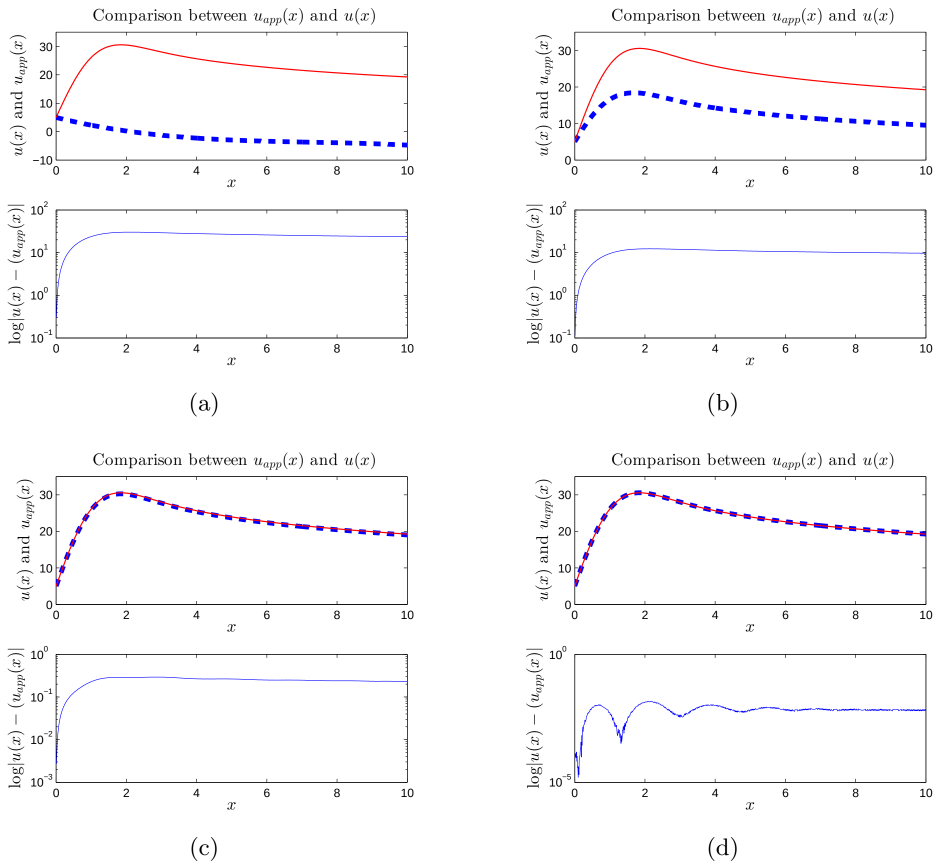
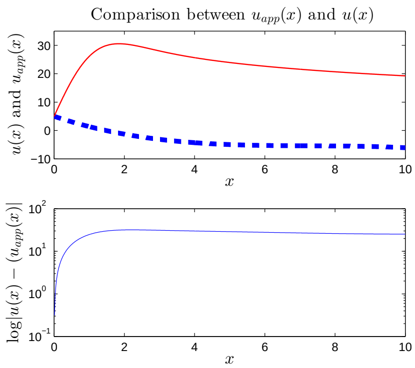
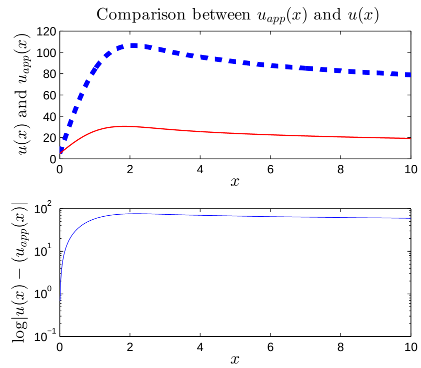


Figure 2: Least Squares Method: (A) 2-Term, (B) 10-Term, (C) 20-Term, (D) 100-Term

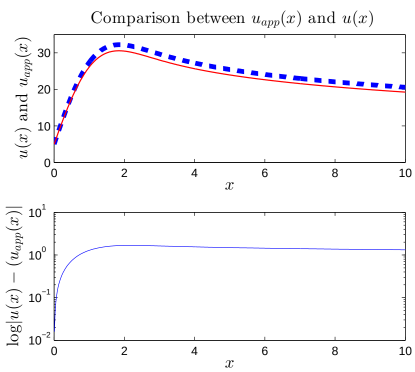
the Least Squares method. However, one factor that is not immediately clear from this study but would become important as the number of terms is further increased is the conditioning of the matrix A. The condition number of the matrix will effect the numerical solution to the matrix inversion problem and could potentially produce inaccurate solutions.



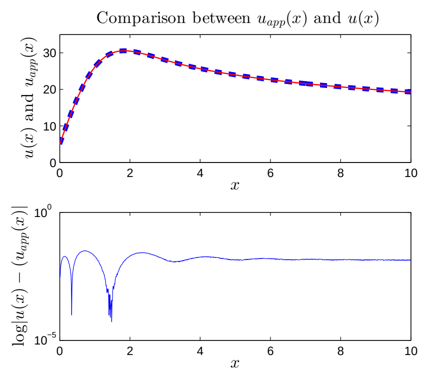
(a)



(b)



(c)



(d)

Figure 3: Galerkin Method: (A) 2-Term, (B) 10-Term, (C) 20-Term, (D) 100-Term

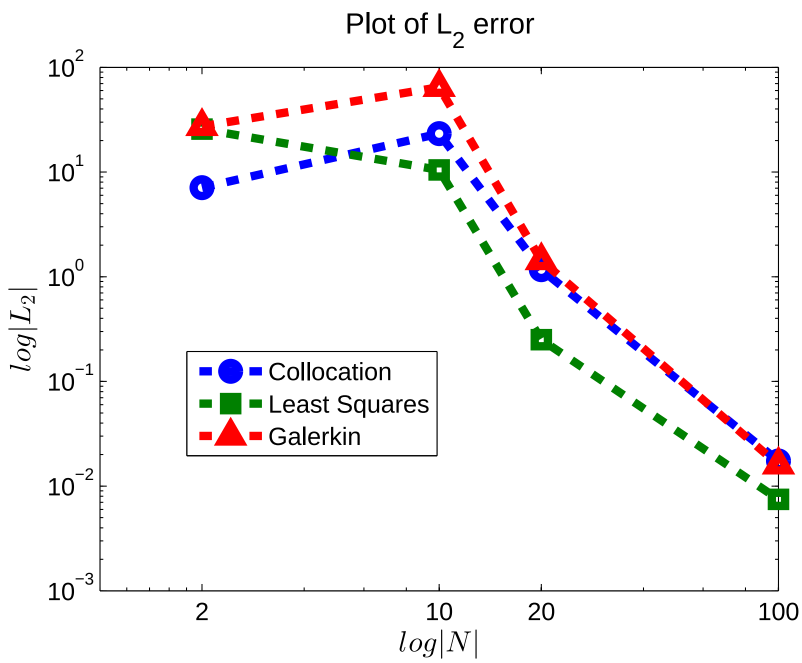


Figure 4: Log–Log plot of the error for each of the three methods