Write a MATLAB (or octave) code to implement either multiple shooting, the midpoint method, or the trapezoid method to solve the two-point boundary value problem

\[ x'(t) = A(t)x(t) + q(t), \quad a \leq t \leq b \]

(1)

\[ B_a x(a) + B_b x(b) = \beta \]

(2)

You should produce a MATLAB function of the form

\[
\text{function } [x,t]=\text{BVPsolve}(f,f0,a,b,Ba,Bb,beta,N)\text{ function}
\]

where \( f \) is a function handle (or string) denoting the function

\[ f(t,x) = A(t)x(t) + q(t), \]

\( f0 \) is a function handle or string denoting the function

\[ f_0(t,x) = A(t)x(t), \]

\( a \) and \( b \) are the boundary points, \( \beta \) is the vector of boundary values, and \( N \) is the number of shooting points. Thus the shooting points will be \( a = t_0 < t_1 < \ldots < t_N = b \) with \( t_j = a + jh \).

If you are doing multiple shooting use the Dormand–Prince method developed earlier this semester with \( tol = 1e-7 \) and \( hmin = 1e-5 \). This should be accurate enough for our purposes. None of the test problems are “stiff.”

You may assume that the boundary conditions are separable. That is,

\[
\begin{pmatrix}
B_a & B_b \\
\bar{B}_a & 0 \\
0 & \bar{B}_b
\end{pmatrix}
\]

If you elect to make this assumption, your program is better off to input \( \bar{B}_a \) and \( \bar{B}_b \). In creating the large linear system that you need to solve, use the command

\[ M = \text{sparsel}(M) \]

1
just do the commands
$\gg [L,U]=lu(M)$;
$\gg d = L\backslash r$
$\gg c = U\backslash d$

For this project, you should turn in your MATLAB code, the value of $x$ and $t$ at the shooting points, and graphs of the functions. That is turn in the plot of

\[
\text{plot}(t,x(1,:),t,x(2,:), \text{ etc, } t,x(n,:))
\]

where $n$ is the number of components of $x$. The value of $n$ is either 2 or 3 for the sample problems.

The test problems are as follows **Test Problem 1**

\[
u'' = 2u'' + u' - 2u + e^{-2t} \]

\[
u'(0) = 1, \quad u(1) - u'(1) = 0, \quad u(1) = 1.
\]

Choose $N = 10$.

**Test Problem 2** Same as test problem 1 except

\[
u'(0) = 1, \quad u(1) - u'(1) = 0, \quad u(0) = 1.
\]

The general solution to this differential equation is

\[
u(t) = \gamma_1 e^t + \gamma_2 e^{2t} + \gamma_3 e^{-t} - \frac{1}{12} e^{-2t}.
\]

**Test Problem 3**

\[-(p(t)u')' + r(t)u = 0\]

where

\[p(t) = 1 - t^2, \quad r(t) = -n(n + 1).\]

Test this problem with $n = 0, 1, 2, 3$ using the boundary conditions

\[
u(0) = 1, \quad u'(0.5) = 0, \quad n = 0,
\]

\[
u(0) = 0, \quad u'(0.5) = 1, \quad n = 1,
\]

\[
u(0) = -0.5, \quad u'(0.5) = 1.5, \quad n = 2,
\]

2
\[ u(0) = 0, \quad u'(0.5) = 3/8, \quad n = 3. \]

The solutions to this test problem are the first four Legendre polynomials.

\[ u(t) = P_0(t) = 1, \]

\[ u(t) = P_1(t) = t, \]

\[ u(t) = P_2(t) = 1.5t^2 - 0.5, \]

\[ u(t) = P_3(t) = 2.5t^3 - 1.5t. \]

For this test problem, use \( N = 5 \). I will discuss in class a way of writing this as a system other than the standard one.