More Boundary-value Problems and Eigenvalue Problems in ODEs

Larry Caretto
Mechanical Engineering 501A

Seminar in Engineering Analysis

November 29, 2017

Outline

- Review boundary-value problems
  - Shoot and try
  - Finite Differences
  - Thomas Algorithm
- Other boundary values
  - Gradient boundary values
  - Mixed boundary values
- Eigenvalue problems
  - Source of such problems
  - Solution methods

2

Review Last Lecture

- In a boundary-value problem, we have conditions set at two different locations
- A second-order ODE $d^2y/dx^2 = g(x, y, y')$, needs two boundary conditions
  - Simplest are $y(0) = a$ and $y(L) = b$
  - Can also have $a\,dy/dx+by = c$ at $x = 0, L$
- Two solution approaches:
  - Shoot-and-try
  - Finite differences

3

Shoot-and-Try Approach

- Take an initial guess of derivative boundary conditions at $x = 0$ and use an initial-value routine to get $y_{(comp)}(L)$ at the other boundary
- Compare the value of $y_{(comp)}(L)$ found from the previous step to the boundary condition on $y(L)$
- Use the difference between $y_{(comp)}(L)$ and $y(L)$ to iterate the initial value of $z = dy/dx|_{x=0}$ and continue until $y_{(comp)}(L) \approx y(L)$

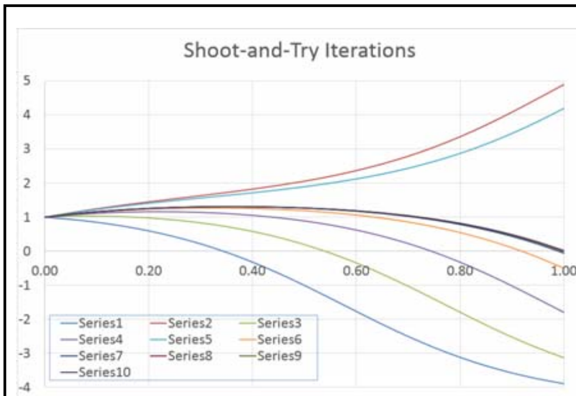
4

Shoot-and-Try Example

- Solve $d^2y/dx^2 + 16\sin(y^2) = 0$ with $y = 1$ at $x = 0$ and $x = L = 1$
- Must find pair of first order equations
  - Set $dy/dx = z$ as one ODE
  - Original ODE becomes $dz/dx = -16\sin(y^2)$
  - We know $y(0) = 1$, but we need $z(0)$ guess $z^{(0)}(0) = [y(L) - y(0)]/L = (0 - 1)/1 = -1$
    - This $z^{(0)}(0)$ gives $y(1) = -0.6824$ (RK4, $h = .01$)
    - Try $z^{(1)}(0) = [y(L) - y^{(0)}(L) - y(0)]/L = [1 - (-0.6824) - 0] = -0.3176$

$E^{(0)} = y^{(0)}(L) - y(L) = -0.6824 - 0 = -0.6824$

5



6

Shoot-and-Try Linear ODEs

- For a linear ODE the solution can be found on the third iteration
  - Complete two shoot-and-try solutions, $y^{(0)}(x)$ and $y^{(1)}(x)$ and for two initial guesses, $z^{(0)}(x_0)$ and $z^{(1)}(x_0)$

Finite Difference Approach

- Define uniform or non-uniform grid; uniform is easier and has higher order truncation error; note: $h = (x_N – x_0)/N$

●-----●--------●-------------●~ ~●-------●---●

x_0 x_1 x_2 x_3 x_{N-2} x_{N-1} x_N

- At each node write finite-difference equivalent to differential equation
- Handle boundary conditions at x_0 and x_N (simplest if $y_0 = y(0)$ and $y_N = y(L)$ given, but can have gradient boundaries)

Finite-Difference Example

- Solve $d^2T/dx^2 + a^2T = 0$
- Finite difference equation at node i
- $[d^2T/dx^2 + a^2T]_i = (T_{i+1} + T_{i-1} – 2T_i)/h^2 + a^2T_i +$ ▭ $= 0$ *Ignore truncation error*
- Ignore truncation error and get finite-difference equation system
- $T_{i+1} + T_{i-1} – 2T_i + h^2a^2T_i = 0$
- Have N+1 nodes numbered from 0 to N with boundary conditions at 0 and N

Tridiagonal Matrix Equations

- Finite-difference equations in matrix form with $\alpha = a^2h^2$ have tridiagonal form solved by Thomas Algorithm used with cubic spline

$$\begin{bmatrix} -2+\alpha & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -2+\alpha & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -2+\alpha & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \ddots & & \vdots & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & 0 & \cdots & -2+\alpha & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 & -2+\alpha \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \vdots \\ \vdots \\ T_{N-2} \\ T_{N-1} \end{bmatrix} = \begin{bmatrix} -T_A \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ -T_B \end{bmatrix}$$

Error and Error Order

- Get overall measure of error (like norm of a vector)
- Typically use maximum error (in absolute value) or root-mean-squared (RMS) error
- N = 10 has $\varepsilon_{max} = 2.42\times10^{-3}$ and $\varepsilon_{RMS} = 1.83\times10^{-3}$. For N = 100, $\varepsilon_{max} = 2.41\times10^{-5}$ and $\varepsilon_{RMS} = 1.73\times10^{-5}$.
- Second-order error in solution

$$|\varepsilon|_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\varepsilon_i^2} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(T_{exact} - T_{numerical})_i^2}$$

Boundary Gradients

- Use second-order derivative expressions

$$q_0 = -k\left.\frac{dT}{dx}\right|_{x=x_0} = -k\frac{-3T_0 + 4T_1 - T_2}{2h}$$

$$q_N = -k\left.\frac{dT}{dx}\right|_{x=x_N} = -k\frac{3T_N - 4T_{N-1} + T_{N-2}}{2h}$$

x	$-q_{exact}/k$	h	-q/k	Error
0	2.1995	.1	2.2357	.03618
0	2.1995	.01	2.1999	.00036
1	-.9153	.1	-.9332	.01786
1	-.9153	.01	-.9155	.00021

Thomas Algorithm

- General format for tridiagonal equations

$$\begin{bmatrix} B_0 & C_0 & 0 & 0 & \cdots & 0 & 0 \\ A_1 & B_1 & C_1 & 0 & \cdots & 0 & 0 \\ 0 & A_2 & B_2 & C_2 & \cdots & 0 & 0 \\ 0 & 0 & A_3 & B_3 & & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & B_{N-1} & C_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & A_N & B_N \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ \vdots \\ \vdots \\ D_{N-1} \\ D_N \end{bmatrix}$$

California State University Northridge 13

Thomas Algorithm II

- Gauss elimination upper triangular form

$$\begin{bmatrix} 1 & -E_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -E_1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & -E_2 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & -E_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ \vdots \\ F_{N-1} \\ F_N \end{bmatrix}$$

Have to find E_i and F_i

California State University Northridge 14

Thomas Algorithm III

- Forward computations
  - Initial: $E_0 = -C_0 / B_0$ $F_0 = D_0 / B_0$
  - Apply equations below for i = 1,… N-1:

$$E_i = \frac{-C_i}{B_i + A_i E_{i-1}} \qquad F_i = \frac{D_i - A_i F_{i-1}}{B_i + A_i E_{i-1}}$$

  - At final point

$$x_N = F_N = \frac{D_N - A_N F_{N-1}}{B_N + A_N E_{N-1}}$$

- Back substitute: $x_i = F_i + E_i x_{i+1}$

California State University Northridge 15

Other Boundary Conditions

- General condition a dT/dx + bT = c
  - a = 1, b = 0 for Neumann (gradient given)
  - a = 0, b = 1 for Dirichlet (value given)
  - Write gradient using second order forward ($x = x_0$) or backward difference ($x = x_N$)
  - Combine with equation for first node in from the boundary to eliminate term with second node from boundary
  - Result conforms to tridiagonal system

California State University Northridge 16

General Boundary Example

- Look at $x = x_0$ boundary; results for $x = x_N$ follow similar derivation

$$a_0 \left.\frac{dy}{dx}\right|_{x=x_0} + b_0 y_0 = a_0 \frac{-3y_0 + 4y_1 - y_2}{2h} + b_0 y_0 = c_0$$

- Add these two equations eliminating y_2

$$\left(b_0 - \frac{3a_0}{2h}\right) y_0 + \frac{4a_0}{2h} y_1 - \frac{a_0}{2h} y_2 = c_0$$

$$\left(y_0 + (\alpha - 2) y_1 + y_2 = 0\right) \frac{a_0}{2h}$$

$$\left(b_0 - \frac{2a_0}{2h}\right) y_0 + \frac{(\alpha + 2) a_0}{2h} y_1 = c_0$$

California State University Northridge 17

General Boundary Example II

- Equation just derived is seen to give correct Dirichlet result for $a_0 = 0$, $b_0 = 1$

$$\left(b_0 - \frac{2a_0}{2h}\right) y_0 + \frac{(\alpha + 2) a_0}{2h} y_1 = c_0$$

- Similar derivation at $x = x_N$ gives

$$\left(b_N + \frac{2a_N}{2h}\right) y_N + \frac{(2 - \alpha) a_N}{2h} y_{N-1} = c_N$$

- Equations shown here will work for a = 0 or b = 0, but at least one must be nonzero

California State University Northridge 18

Nonlinear Problems

- Shoot-and-try requires no special procedures for nonlinear problems
- For finite difference or finite elements, solve a linearized equation
  - Example is pendulum equation $d^2\theta/dt^2 = (-g/l) \sin\theta$ (usually solved with $\sin\theta \approx \theta$)
  - Taylor series: $\sin\theta = \sin\theta_0 + [d(\sin\theta)/d\theta]_0 (\theta - \theta_0) = \sin\theta_0 + \cos\theta_0 (\theta - \theta_0)$
  - Replace $\sin\theta$ by linear result to iterate

California State University Northridge 19

Nonlinear Example

- Start with $d^2\theta/dt^2 = (-g/l) \sin\theta$
- Replace $\sin\theta$ by linearized series
- Write in iterative form with $\theta^{(m+1)}$ as new iteration and use $\theta^{(m)}$ in nonlinear terms
- $d^2\theta^{(m+1)}/dt^2 = (-g/l) [\sin\theta^{(m)} + \cos\theta^{(m)} (\theta^{(m+1)} - \theta^{(m)})$
- Define $\omega^2 = g/l$ and rearrange
  $d^2\theta^{(m+1)}/dt^2 + \omega^2\theta^{(m+1)} \cos\theta^{(m)} = -\omega^2 [\sin\theta^{(m)} - \theta^{(m)}\cos\theta^{(m)}] = r$

California State University Northridge 20

Nonlinear Example II

- Convert $d^2\theta^{(m+1)}/dt^2 + \omega^2\theta^{(m+1)} \cos\theta^{(m)} = r$ to (linear) finite-difference form in $\theta^{(m+1)}$

$$\frac{\theta_{i+1}^{(m+1)} + \theta_{i-1}^{(m+1)} - 2\theta_i^{(m+1)}}{h^2} + \omega^2\theta_i^{(m+1)} = r_i$$

$$r_i = \omega^2\left[\theta_i^{(m)} \cos\theta_i^{(m)} - \sin\theta_i^{(m)}\right]$$

- Have tridiagonal system

$$\theta_{i-1}^{(m+1)} + \left[\omega^2 h^2 - 2\right]\theta_i^{(m+1)} + \theta_{i+1}^{(m+1)} = h^2 r_i$$

California State University Northridge 21

Nonlinear Example III

- Make initial guesses for $\theta^{(0)}$
  - Linear profile $\theta^{(0)}(t) = \theta(0) + [\theta(L) - \theta(0)]t/T$
- Find all nodal values for $\theta^{(1)}$ using $\theta^{(0)}$ to compute the nonlinear terms
- Repeat the process until the differences between iterations is good enough
  - Compute residuals to test convergence

$$R_i = \theta_{i-1}^{(m+1)} + \left[\omega^2 h^2 - 2\right]\theta_i^{(m+1)} + \theta_{i+1}^{(m+1)} - h^2 r_i^{(m+1)}$$

California State University Northridge 22

Handling Boundary Conditions

- We have used an example problem with fixed (Dirichlet) boundary conditions
- We also mentioned gradient (Neumann) and mixed boundary conditions.
- If the dependent variable is u and the initial and final nodes in a finite-difference grid are 0 and N the fixed boundary conditions are $u_0 = C_0$ at $x = x_0$ and $u_N = C_N$ at $x = x_N$

California State University Northridge 23

Gradient and Mixed Boundaries

- We use a directional derivative to represent boundary gradients

$$\left.\frac{du}{dx}\right]_{x=0} = \frac{-3u_0 + 4u_1 - u_2}{2\Delta x}$$

$$\left.\frac{du}{dx}\right]_{x=N} = \frac{3u_N - 4u_{N-1} + u_{N-2}}{2\Delta x}$$

- Couple this boundary condition equation with first (last) possible finite difference equation at $x = x_0$ and $x = x_N$

California State University Northridge 24

Gradient Boundary at $x = x_0$

- Finite-difference equation for $d^2y/dx^2 - ay = 0$ is $(u_{i+1} - 2u_i + u_{i-1})/h^2 + a^2u_i = 0$
  - Write this as $u_{i+1} + \alpha u_i + u_{i-1} = 0$ where $\alpha = h^2a^2 - 2$
  - First ($x = 0$) equation is $u_2 + \alpha u_1 + u_0 = 0$
  - Gradient at $x = 0$: $2\Delta x g_0 = -3u_0 + 4u_1 - u_2$
  - Combine these to get $2\Delta x g_0 = -3u_0 + 4u_1 - u_2 = -3u_0 + 4u_1 + (\alpha u_1 + u_0) = (4 + \alpha)u_1 - 2u_0$
  - For TDMA solution combined equation $[-2u_0 + (4 + \alpha)u_1 = 2\Delta x g_0]$ is first; previous first equation, $u_2 + au_1 + u_0 = 0$, is second

California State University Northridge 25

Gradient Boundary at $x = x_N$

- Start with same finite-difference equation $u_{i+1} + \alpha u_i + u_{i-1} = 0$ where $\alpha = h^2a^2 - 2$
  - Last ($x = x_N$) equation is $u_N + \alpha u_{N-1} + u_{N-2} = 0$
  - Gradient at $x = x_N$: $2\Delta x g_N = 3u_N - 4u_{N-1} + u_{N-2}$
  - Combine these to get $2\Delta x g_N = 3u_N - 4u_{N-1} + u_{N-2} = 3u_N - 4u_{N-1} - (\alpha u_{N-1} + u_N) = -(4 + \alpha)u_{N-1} + 2u_N$
  - For TDMA solution combined equation $[2u_N - (4 + \alpha)u_{N-1} = 2\Delta x g_N]$ is last; previous last equation, $u_{N-2} + \alpha u_{N-1} + u_N = 0$, is second-to-last

California State University Northridge 26

Mixed Boundary Condition

- Mixed boundary condition relates boundary gradient to boundary value of dependent variable
- Common example is convection heat transfer boundary condition:
  - At $x = x_0$: $-k(dT/dx) = h(T_\infty - T)$
  - At $x = x_N$: $-k(dT/dx) = h(T - T_\infty)$
  - Use forward/backward expression for gradients

California State University Northridge 27

Mixed Boundary Condition II

- For $x = x_0$: $-k(dT/dx) = h(T_\infty - T)$ with forward difference becomes
  $-k\frac{-3T_0+4T_1-T_2}{2\Delta x} = \mathrm{h}(T_\infty - T_0)$
- For $x = x_N$: $-k(dT/dx) = h(T - T_\infty)$ with backward difference becomes
  $-k\frac{3T_N-4T_{N-1}+T_{N-2}}{2\Delta x} = \mathrm{h}(T_N - T_\infty)$
- Combine with finite difference equations: $T_2 + \alpha T_1 + T_0 = 0$ / $T_{N-2} + \alpha T_{N-1} + T_N = 0$

California State University Northridge 28

Mixed Boundary at $x = x_0$

- Finite-difference: $T_2 + \alpha T_1 + T_0 = 0$
- Boundary: $-k\frac{-3T_0+4T_1-T_2}{2\Delta x} = \mathrm{h}(T_\infty - T_0)$
- $-3T_0 + 4T_1 - T_2 = -2h\Delta x(T_\infty - T_0)/k$
- $-3T_0 + 4T_1 + \alpha T_1 + T_0 = -2h\Delta x(T_\infty - T_0)/k$
- $-\left(2 + \frac{2h\Delta x}{k}\right)T_0 + (4 + \alpha)T_1 = \frac{-2h\Delta x T_\infty}{k}$
- This becomes the first equation in the TDMA algorithm and the first finite-difference equation above becomes the second

California State University Northridge 29

Mixed Boundary at $x = x_N$

- Finite-difference: $T_{N-2} + \alpha T_{N-1} + T_N = 0$
- Boundary: $-k\frac{3T_N-4T_{N-1}+T_{N-2}}{2\Delta x} = \mathrm{h}(T_N - T_\infty)$
- $3T_N - 4T_{N-1} + T_{N-2} = -2h\Delta x(T_N - T_\infty)/k$
- $3T_N - 4T_{N-1} - \alpha T_{N-1} - T_N = -2h\Delta x(T_N - T_\infty)/k$
- $\left(2 + \frac{2h\Delta x}{k}\right)T_N - (4 + \alpha)T_{N-1} = \frac{2h\Delta x T_\infty}{k}$
- This becomes the last equation in the TDMA algorithm and the first finite-difference equation above becomes the second-to-last

California State University Northridge 30

Eigenvalue Problems

- Numerical eigenvalue problems occur when the number of boundary conditions is greater than the order of the differential equation
  - Example of this is solution for burning velocity of a laminar flame
- Basic approach is to use finite-differences and transform problem into a numerical matrix eigenvalue problem

California State University Northridge 31

Eigenvalue Problems II

- Look at simple problem with known solution as an example
  - $d^2y/dx^2 + \lambda^2 y = 0$ with $y(0) = 0$, $y(1) = 0$ and $\int_0^1 y dx = 1$
  - Have three boundary conditions and only a second order equation
  - Nontrivial solution: $y = A \sin \lambda x$ with $\lambda = n\pi$
- Use second order finite differences
  - $(y_{i+1} + y_{i-1} - 2y_i)/h^2 + \lambda^2 y_i = 0$

California State University Northridge 32

Eigenvalue Problem III

- Finite-difference equations in matrix form with $\alpha = \lambda^2 h^2$; what is solution?

$$\begin{bmatrix} -2+\alpha & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -2+\alpha & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -2+\alpha & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \ddots & & \vdots & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & 0 & \cdots & -2+\alpha & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 & -2+\alpha \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \vdots \\ y_{N-2} \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

California State University Northridge 33

Eigenvalue Problem IV

- Have matrix eigenvalue problem with $\alpha = -\lambda^2 h^2$ as the eigenvalue

$$\begin{bmatrix} -2 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \ddots & & \vdots & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & 0 & \cdots & -2 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 & -2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \vdots \\ y_{N-2} \\ y_{N-1} \end{bmatrix} = -\lambda^2 h^2 \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \vdots \\ y_{N-2} \\ y_{N-1} \end{bmatrix}$$

California State University Northridge 34

Eigenvalue Problems V

- Solve by numerical techniques for finding matrix eigenvalues
- The accuracy of the eigenvalues depends on the grid
- Often need only one (lowest or highest)
- Can only find as many eigenvalues as there are grid nodes (not counting boundary nodes)

California State University Northridge 35

Example Eigenvalue Problem

- ODE: $d^2y/dx^2 + k^2 y = 0$ with $y(0)=y(1)=0$
- k is an unknown parameter (eigenvalue)
- Solution is $y = A\sin(kx)$ where $k = n\pi$
- Solve with $\Delta x = 0.2$
- Finite difference equation is

$$\frac{y_{i+1} + y_{i-1} - 2y_i}{(\Delta x)^2} + k^2 y_i = 0$$

$$y_{i+1} + y_{i-1} - [2 - (\Delta x)^2 k^2] y_i = 0$$

- Write as matrix equation for $\Delta x = 0.2$

California State University Northridge 36

Example Eigenvalue Problem II

$$\begin{bmatrix} 2-0.04k^2 & -1 & 0 & 0 \\ -1 & 2-0.04k^2 & -1 & 0 \\ 0 & -1 & 2-0.04k^2 & -1 \\ 0 & 0 & -1 & 2-0.04k^2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- This is eigenvalue equation $(\mathbf{A} - \mathbf{I}\lambda)\mathbf{y} = \mathbf{0}$
- Here $\lambda = 0.04k^2$
- Det$(\mathbf{A} - \mathbf{I}\lambda) = 0$ is $(2 - 0.04k^2)^4 - 3(2 - 0.04k^2)^2 + 1 = 0$
- Numerical solutions for k compared to exact values on next slide

California State University Northridge 37

Example Eigenvalue Problem III

Eigenvalues Numerical	Exact	Percent error
3.090	3.142	1.66%
5.878	6.283	6.45%
8.090	9.425	14.16%
9.511	12.566	24.31%

- Note larger errors for higher eigen-values

Joe D. Hoffman, *Numerical methods for Engineers and Scientists*, (2nd ed), Marcel Dekker (2001), p. 482.

California State University Northridge 38

MATLAB ODE Eigenvalues

- MATLAB has two solvers bvp4c and bvp5c for solving boundary-value ODEs
- MATLAB documentation shows the use of bvp4c for computing the eigenvalue of an ODE
  - https://www.mathworks.com/help/matlab/ref/bvp4c.html
- Example shows the computation of a single eigenvalue as unknown parameter in the solution based on initial guess of eigenvalue

California State University Northridge 39