Lecture 16: Boundary Value Problems

Boundary value problems (as opposed to initial value problems) are given by ordinary differential equations with conditions at the boundary point rather than at an initial point.

The conditions at the end/boundary points could themselves contain derivatives.

An example of a second order Boundary Value Problem (BVP)

$$\frac{d^2y}{dt^2} = f(t, y, \frac{dy}{dt})$$

for $t \in [a, b]$, with boundary conditions

$$\alpha_1 y(a) \;+\; \beta_1 \frac{dy(a)}{dt} = \gamma_1$$

$$\alpha_2 y(b) \;+\; \beta_2 \frac{dy(b)}{dt} = \gamma_2$$

We introduce the shooting method on a simplified BVP

$$\frac{d^2y}{dt^2} = f(t, y, \frac{dy}{dt}) \tag{1}$$

for $t \in [a, b]$, with boundary conditions

$$\begin{aligned} y(a) &= \alpha \\ y(b) &= \beta \end{aligned}$$

From our previous knowledge for Initial Value Problems (IVP) we can solve the equation (1) with two initial conditions for y and y' at $t = a$,

$$\begin{aligned} y(a) &= \alpha \\ y'(a) &= A \end{aligned}$$

Because we were given boundary conditions rather than the initial conditions part of the problem will be to find the A which satisfies the initial conditions for the solutions of equation (1) and satisfies the boundary conditions.

Computational Algorithm for the shooting method

- 1. Solve the ODE (say using ode45) with the initial conditions $y(a) = \alpha$, $y'(a) = A$.

- 2. Evaluate the solution $y(b)$ and compare with the target for $y(b)$: $y(b) = \beta$

- 3. Adjust A until a desired level of accuracy is achieved.

Implement for the simple problem with

$f(t, y, \frac{dy}{dt}) = -y$

over the interval $[0, 1]$ and where $\alpha = -1$ and $\beta = 2$:

Let $y_1' = y_2$ and $y_2' = -y_1$

which satisfies the ODE $y_1'' = -y_1$ and the boundary conditions become:

$y_1(0) = -1$ and $y_1(1) = 2$.

Solve the ODE using the ode45 for 3 different A's, thus we will apply the concept manually: In MATLAB open the files: f.m and solve_bvp.m.

We can find the correct solution by trial and error, and further we can write a code that finds the correct A using the shooting method with the bisection method.

Open the files solve_bvp_2.m

Next class we'll cover more methods for solving boundary value problems, in the meantime for checking your solutions in MATLAB when solving a BVP you can use a built in funciton which uses a method not covered in this class called collocation.

The built in function is bvp4c and you must supply it a function as well as boundary conditions. For an example open file fbc.m

```
solinit = bvpinit([0:.1:1],[-1,1])
sol = bvp4c('f','fbc',solinit);

% Look at solutions
sol.x'
sol.y'

plot(sol.x, sol.y(1,:),'*-')
```