Higher order ODEs: Boundary Value problems

- For a second-order ODE, there need to be two conditions specified on y
- If at different points (e.g., y_0 and y_T): Boundary Value Problem (BVP)
- Generally, the independent variable is x
- Therefore, we now use x instead of t
- For example, $p\left(x, y, \frac{dy}{dx}\right)\frac{d^2y}{dx^2} + q\left(x, y, \frac{dy}{dx}\right)\frac{dy}{dx} + r\left(x, y, \frac{dy}{dx}\right) = 0$
  - $y(0)=y_a$; $y(1)=y_b$
- Linear BVP: $p(x)$, $q(x)$, and r linear in y

Boundary Value problems: Methods of solution

- Convert into a system of equations – Shooting Method
- Approximate the derivatives by finite differences: Direct Method
- Only Linear BVPs are considered

$$p(x)\frac{d^2y}{dx^2}+q(x)\frac{dy}{dx}+r_1(x)y=r_0(x)$$

- Solution domain (a,b) and specified conditions y_a and y_b (or could be y'_b, or any combination of y and y')

Boundary Value problems: Shooting Method

- Convert into two first-order ODEs

$$p(x)\frac{d^2y}{dx^2}+q(x)\frac{dy}{dx}+r_1(x)y=r_0(x)$$

- y_1=>y; y_2=>dy/dx

$$\frac{dy_1}{dx}=f_1(x,y_1,y_2)=y_2$$

$$\frac{dy_2}{dx}=f_2(x,y_1,y_2)=\frac{r_0(x)-r_1(x)y_1-q(x)y_2}{p(x)}$$

- Boundary conditions: $y_1(a)=y_a; y_1(b)=y_b$
- For IVP, we need y_2(a), which is not given
- Assume y_2(a), solve IVP, compare y_1(b)

Shooting Method

- Generally, the computed $y_1(\mathrm{b})$ will not be equal to the given y_b
- Assume a different $y_2(\mathrm{a})$, solve IVP till b, to obtain another value of $y_1(\mathrm{b})$
- Use a linear interpolation/extrapolation to estimate the $y_2(\mathrm{a})$ which will result in $y_1(\mathrm{b})$ equal to $\mathrm{y_b}$.
- Solve the IVP again with this value of $y_2(\mathrm{a})$. For linear problems, the solution could be obtained by linear interpolation.

Shooting Method: Example

- Second-order equation:

$$\frac{d^2y}{dx^2} + 3\frac{dy}{dx} + 2y = 6e^x$$

- Boundary conditions: y(0)=y(1)=0
- Write as:

$$\frac{dy_1}{dx} = y_2$$

$$\frac{dy_2}{dx} = 6e^x - 2y_1 - 3y_2$$

- Solve by Ralston’s method

Shooting Method: Example

- Ralston's method

$$y_{n+1} = y_n + \frac{h}{3}\left(f(x_n, y_n) + 2f\left(x_n + \frac{3}{4}h, y_n + \frac{3}{4}hf(x_n, y_n) \right)\right)$$

$$y_{1,n+1} = y_{1.n} + \frac{h}{3}\left(y_{2,n} + 2\left[y_{2,n} + \frac{3h}{4}\left(6e^{x_n} - 2y_{1,n} - 3y_{2,n}\right)\right]\right)$$

$$y_{2,n+1} = y_{2.n} + \frac{h}{3}\left[\left(6e^{x_n} - 2y_{1,n} - 3y_{2,n}\right) + 2\left(\begin{array}{l} 6e^{x_n + 3h/4} - 2\left\{ y_{1,n} + \dfrac{3h}{4} y_{2,n} \right\} \\ -3\left\{ y_{2,n} + \dfrac{3h}{4}\left(6e^{x_n} - 2y_{1,n} - 3y_{2,n}\right)\right\} \end{array} \right) \right]$$

Shooting Method: Example

- First assume $y_2(0)=0$, then 1. Use h=0.2

x	y1	y2	f1	f2
0	0	0	0	6.00000
0.15	0	0.9	0.9	4.27101
0.2	0.12	0.969467		
x	y1	y2	f1	f2
0.2	0.12	0.969467	0.969467	4.18001
0.35	0.26542	1.59647	1.59647	3.19416
0.4	**0.397494**	1.674023		
x	y1	y2	f1	f2
0.4	0.397494	1.674023	1.674023	3.13389
0.55	0.648597	2.144106	2.144106	2.67000
0.6	**0.794976**	2.238949		
x	y1	y2	f1	f2
0.6	0.794976	2.238949	2.238949	2.62591
0.75	1.130819	2.632836	2.632836	2.54185
0.8	**1.295284**	2.752924		
x	y1	y2	f1	f2
0.8	1.295284	2.752924	2.752924	2.50390
0.95	1.708223	3.12851	3.12851	2.71228
1	**1.895947**	3.281489		

x	y1	y2	f1	f2
0	0	1	1	3
0.15	0.15	1.45	1.45	2.321005
0.2	0.26	1.509467		
x	y1	y2	f1	f2
0.2	0.26	1.509467	1.509467	2.280014
0.35	0.48642	1.85147	1.85147	1.987156
0.4	**0.607494**	1.926423		
x	y1	y2	f1	f2
0.4	0.607494	1.926423	1.926423	1.956693
0.55	0.896457	2.219926	2.219926	1.946824
0.6	**1.031912**	2.316445		
x	y1	y2	f1	f2
0.6	1.031912	2.316445	2.316445	1.919553
0.75	1.379379	2.604378	2.604378	2.130108
0.8	**1.533592**	2.72843		
x	y1	y2	f1	f2
0.8	1.533592	2.72843	2.72843	2.100772
0.95	1.942857	3.043546	3.043546	2.497908
1	**2.121294**	3.201536		

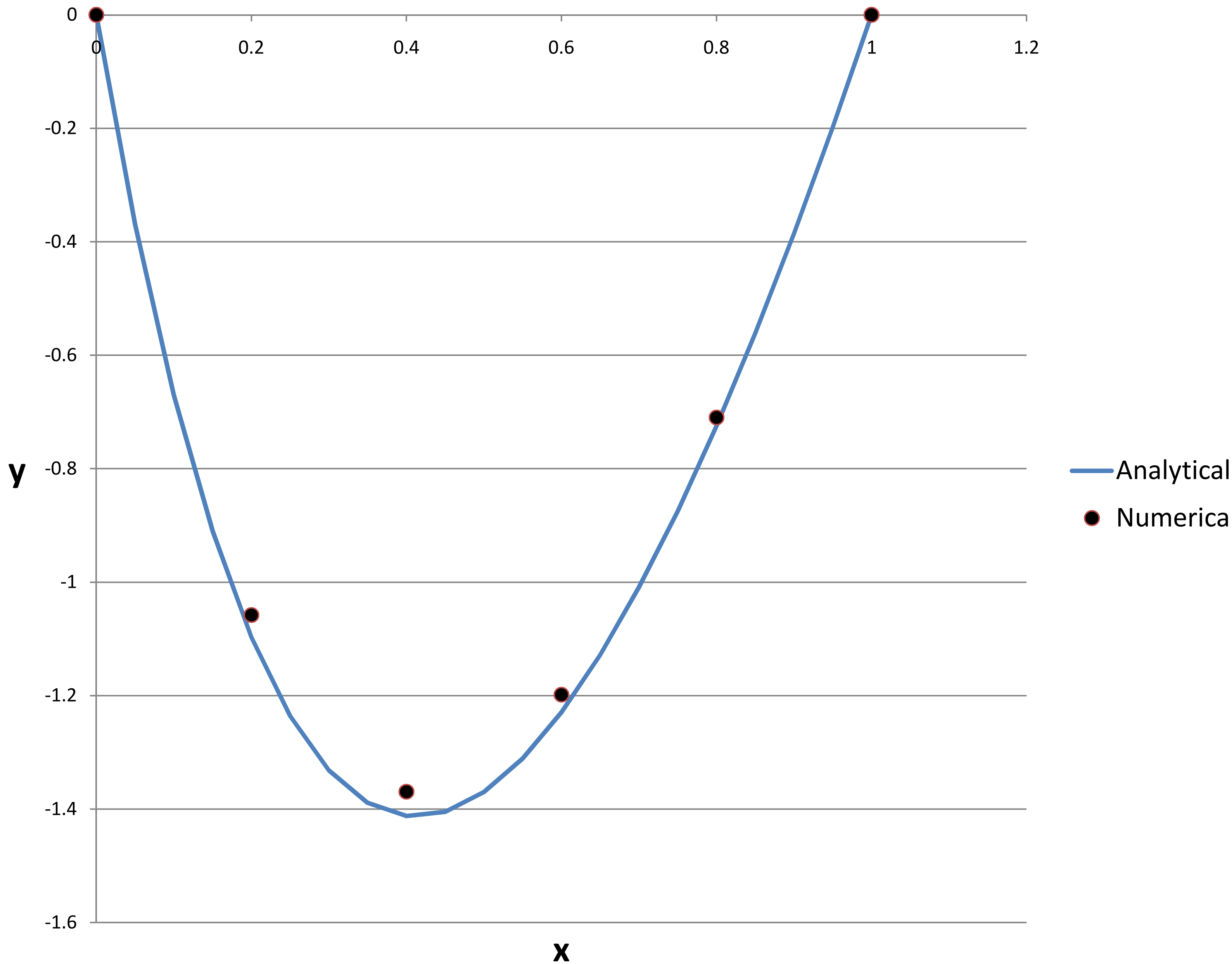
Shooting Method: Example

- For $y_2(0)=0$, $y_1(1)=1.896$
- For $y_2(0)=1$, $y_1(1)=2.121$
- Specified value is $y_1(1)=0$
- Linear extrapolation => $y_2(0)= -8.41348$
- Solve the IVP again
- In this case, we do not need to solve again. Just use linear extrapolation of values obtained for the two assumed derivative values.

Shooting Method: Example

- Solution of IVP with $y_2(0)$= -8.41348

x	y1	y2	f1	f2
0	0	-8.41348	-8.41348	31.24043
0.15	-1.26202	-3.72741	-3.72741	20.67728
0.2	-1.05789	-3.57381		
x	y1	y2	f1	f2
0.2	-1.05789	-3.57381	-3.57381	20.16562
0.35	-1.59396	-0.54897	-0.54897	13.34922
0.4	-1.36934	-0.44954		
x	y1	y2	f1	f2
0.4	-1.36934	-0.44954	-0.44954	13.03824
0.55	-1.43677	1.506197	1.506197	8.75446
0.6	-1.19848	1.586939		
x	y1	y2	f1	f2
0.6	-1.19848	1.586939	1.586939	8.56886
0.75	-0.96044	2.872267	2.872267	6.00608
0.8	-0.70971	2.959006		
x	y1	y2	f1	f2
0.8	-0.70971	2.959006	2.959006	5.89566
0.95	-0.26586	3.843354	3.843354	4.51592
1	0	3.954172		



0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
-1.6
0
0.2
0.4
0.6
0.8
1
1.2
y
x
Analytical
Numerical

Shooting Method: Different Boundary Conditions

- What if dy/dx is specified at "b"?
- Same methodology, compare $y_2(b)$
- If both y and dy/dx are specified at b?
- IVP with a negative h
- If dy/dx specified at a and y at b?
- Assume two different $y_1(a)$, solve the IVP and compare $y_1(b)$
- For nonlinear problems, more iterations are needed. To avoid that: Direct Method

Boundary Value problems: Direct Method

- Approximate the derivatives by finite differences using a grid of points (generally equally spaced)
- Take linear equation:

$$p(x)\frac{d^2y}{dx^2}+q(x)\frac{dy}{dx}+r_1(x)y=r_0(x)$$

- with the boundary conditions

$$y(a)=y_a; y(b)=y_b$$

- Let (a,b) be divided into n equal intervals [h=(b-a)/n]

Direct Method

- The grid points are called Nodes
- Let the node numbers be denoted by 0 (at a),1,2,…,i-1,i,i+1,…,n-1,n (at b)
- The derivatives at the nodes are approximated by appropriate finite difference formula (generally central)
- For example,

$$p(x)\frac{d^2y}{dx^2}+q(x)\frac{dy}{dx}+r_1(x)y=r_0(x)$$

Direct Method

$$p(x)\frac{d^2y}{dx^2}+q(x)\frac{dy}{dx}+r_1(x)y=r_0(x)$$

- Using the lowest order central difference, at the i^th node:

$$p(x_i)\frac{y_{i+1}-2y_i+y_{i-1}}{h^2}+q(x_i)\frac{y_{i+1}-y_{i-1}}{2h}+r_1(x_i)y_i=r_0(x_i)$$

- Clearly, it will not work at the 0^th and n^th nodes. We will see later how to handle it

Direct Method

- At each node, we get an equation relating the y values at nodes i-1, i, and i+1 (or more, if higher order finite difference formula is used)

$$a_{i,i-1}y_{i-1} + a_{i,i}y_i + a_{i,i+1}y_{i+1} = b_i$$

- where:

$$a_{i,i-1} = \frac{p(x_i)}{h^2} - \frac{q(x_i)}{2h}; a_{i,i} = -2\frac{p(x_i)}{h^2} - r_1(x_i);$$

$$a_{i,i+1} = \frac{p(x_i)}{h^2} + \frac{q(x_i)}{2h}; b_i = r_0(x_i)$$

Direct Method

- This results into a tridiagonal system

$$[A]\{y\} = \{b\}$$

(Higher order approximations will result in a banded coefficient matrix)

- May be solved by Thomas algorithm to obtain the nodal values of y
- As mentioned before, the equations are not applicable at 0^{th} and n^{th} nodes
- May use Forward at 0 and Backward at n

Direct Method: Boundary Conditions

- However, it will destroy the tri-diagonal nature of the matrix
- We, therefore, have n-1 equations and n+1 unknowns (nodal values of y)
- The other two equations come from the boundary conditions
- Simplest boundary conditions to incorporate are in terms of specified y at x=a and x=b (known as the Dirichlet BC or the First-type BC)

Direct Method: Boundary Conditions

- For Dirichlet B.C., we do not need to write the equations at node 0 and n
- The equations for nodes 1 and n-1 are modified by using y_a and y_b as follows:

$$a_{1,1}y_1 + a_{1,2}y_2 = b_1 - a_{1,0}y_a$$
$$a_{n-1,n-2}y_{n-2} + a_{n-1,n-1}y_{n-1} = b_1 - a_{n-1,n}y_b$$

- We get n-1 equations in tridiagonal form
- If the derivative is specified at, say, x=b (known as Neumann or second-type BC)

Direct Method: Boundary Conditions

- For this Neumann B.C., y_n is unknown and we do need an equation at node n
- Let the specified derivative be y'_b
- We could use a backward difference at b

$$\frac{y_n - y_{n-1}}{h} = y'_b \Rightarrow -y_{n-1} + y_n = hy'_b$$

- This will preserve the tridiagonal nature
- However, the order of accuracy is lower
- May use higher order backward difference

Direct Method: Boundary Conditions

- Higher order backward difference

$$y_{n-2} - 4y_{n-1} + 3y_n = 2hy'_b$$

- More accurate but not tridiagonal
- Virtual, Imaginary, or Ghost Node:
  - Add a fictitious node (n+1)
  - The equation at node n can now be written
  - Write central difference approximation as

$$\frac{y_{n+1} - y_{n-1}}{2h} = y'_b \Rightarrow y_{n+1} = y_{n-1} + 2hy'_b$$

Boundary Conditions: Ghost Node

- The equation at node n becomes

$$\left(a_{n,n-1} + a_{n,n+1}\right) y_{n-1} + a_{n,n} y_n = b_1 - a_{n,n+1} 2h y_b'$$

- Preserves tridiagonal nature
- n equations for n unknowns (y_a is given)
- If the Neumann condition is specified at x=a, we could use a backward difference approximation at node 0, or a ghost node (-1) to the left