Ordinary Differential Equations (BVP)

Boundary Value Problems:

- **Shooting Method**
- **Finite Difference Method**
- Collocation

Boundary-Value versus Initial Value Problems

Initial-Value Problems

The auxiliary conditions are specified at the same value of the independent variable

$$\ddot{x} + 2\dot{x} + x = e^{-2t}$$

$$x(0) = 1, \quad \dot{x}(0) = 2.5$$

same

Boundary-Value Problems

The auxiliary conditions are specified at different values of the independent variable.

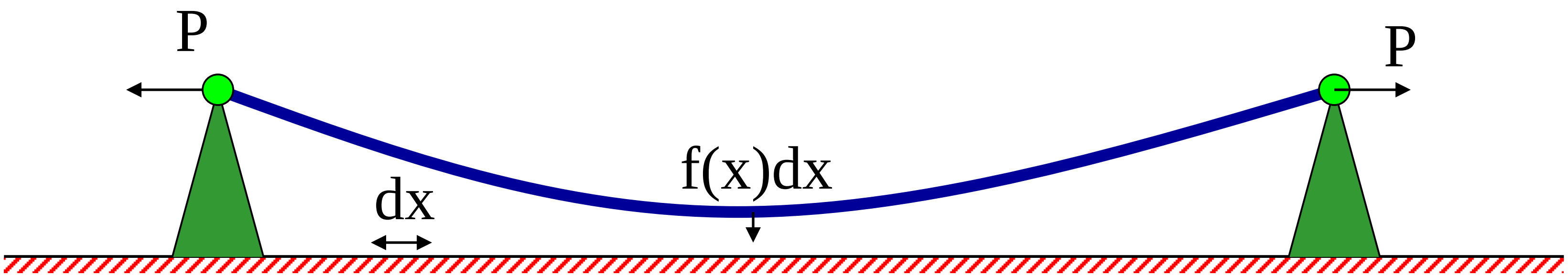
- More difficult to solve than initial value problem

$$\ddot{x} + 2\dot{x} + x = e^{-2t}$$

$$x(0) = 1, \quad x(2) = 1.5$$

different

Example: Beam-deflection Problem



$$\begin{cases} -y'' + c(x)y(x) = f(x), & 0 \le x \le L \\ y(0) = y(L) = 0 \end{cases}$$

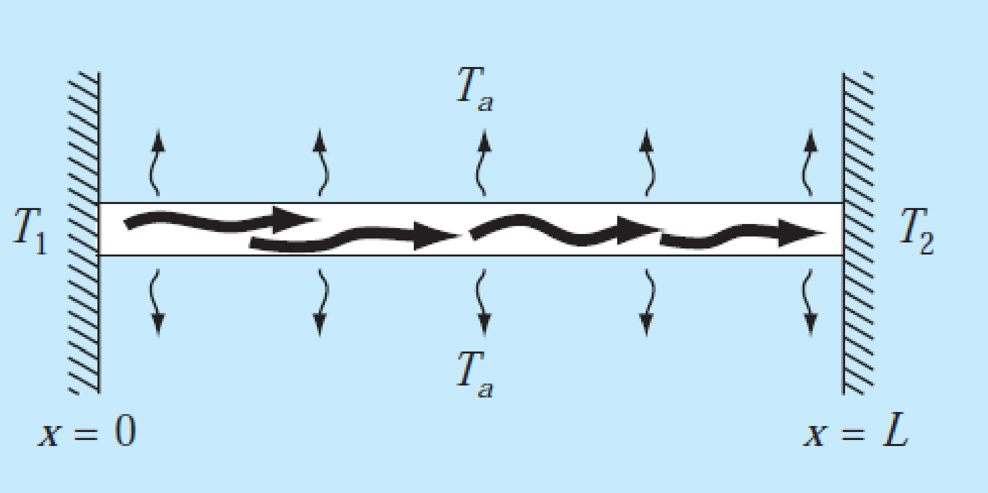
- *y(x)* – deflection of the beam
- f(x)dx – transversal load
- P - constraints

- *c(x)=P/(EI(x))*
- *E* - Young's modulus of elasticity of the beam (material)
- *I(x)* - moment of inertia about the x axis

The solution is unique when $c(x) \ge 0$

Example: Temperature-distribution

- A noninsulated uniform rod positioned between two bodies of constant but different temperature.
- For this case *T*1 > *T*2 and *T*2 > *Ta*.



$$\begin{cases} \dfrac{d^2y}{dx^2} + h'(T_a - y) = 0, \quad 0 \le x \le L \\ y(0) = T_1 \; ; \; y(L) = T_2 \end{cases}$$

- where h' is a heat transfer coefficient that parameterizes the rate of heat dissipation to the surrounding air and T_a is the temperature of the surrounding air (C°).

Boundary Conditions

$$y'' = f(x, y, y'), \qquad a \le x \le b$$

- **Dirichlet boundary conditions**

$$y(a) = \alpha, \quad y(b) = \beta$$

- **Neumann boundary conditions**

$$y'(a) = \alpha, \quad y'(b) = \beta$$

- **Robin boundary conditions**

$$y'(a) + \alpha_1 y(a) = \alpha, \quad y'(b) + \beta_1 y(b) = \beta$$

Physical problems governed by second-order BVP

- Deflection of a beam subjected to transverse load
- Distribution of electric potential between two electrodes
- Temperature distribution in a medium with temperatures set to extremes
- ……

Second order BVP

$$y'' = f(x, y, y'), \qquad a \le x \le b$$

Boundary Conditions:

$$\begin{cases} \alpha_0 y(a) - \alpha_1 y'(a) = \alpha \\ \beta_0 y(b) + \beta_1 y'(b) = \beta \end{cases}$$

Convert into ODE-IVP:

$$\begin{bmatrix} y_1' \\ y_2' \end{bmatrix} = \begin{bmatrix} y_2 \\ f(x, y_1, y_2) \end{bmatrix}, \qquad a \le x \le b$$

When $\begin{cases} y(a) = \alpha \\ y(b) = \beta \end{cases}$

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1(a) \\ y_2(a) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_1(b) \\ y_2(b) \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

Shooting Method

- The shooting method is based on converting the boundary-value problem into an equivalent initial-value problem which is solved by recasting it as a zero-finding problem.
- **ODE**

Non-linear

$$\begin{cases} y'' = f(x, y, y'), \quad a \le x \le b \\ y(a) = \alpha, \quad y(b) = \beta \end{cases}$$

- Step 1: Associate a family of second order IVPs :

$$\begin{cases} y'' = f(x, y, y') \\ y(a) = \alpha, \quad y'(a) = s \end{cases}$$

with guessed slope s

- Step 2: Associate a first order IVP system with 2 eqs
- Step 3: iteratively solve the zero-finding problem.

Example

$$\frac{d^2 y(x)}{dx^2} = -\lambda y(x)$$

with BC $\quad y(0)=0, \;\; y(1)=0$

λ is given

Step 1 $\quad IVP: \dfrac{d^2 y(x)}{dx^2} = -\lambda y(x)$

with IC $\quad y(0)=0, \;\; y'(0)=s$

Step 2: $\quad$ Let $y_1 = y, \;\; y_2 = y', \quad \begin{cases} y_1' = y_2 \\ y_2' = -\lambda y_1 \end{cases}$

$$y_1(0)=0, \quad y_2(0)=s$$

Shooting Method

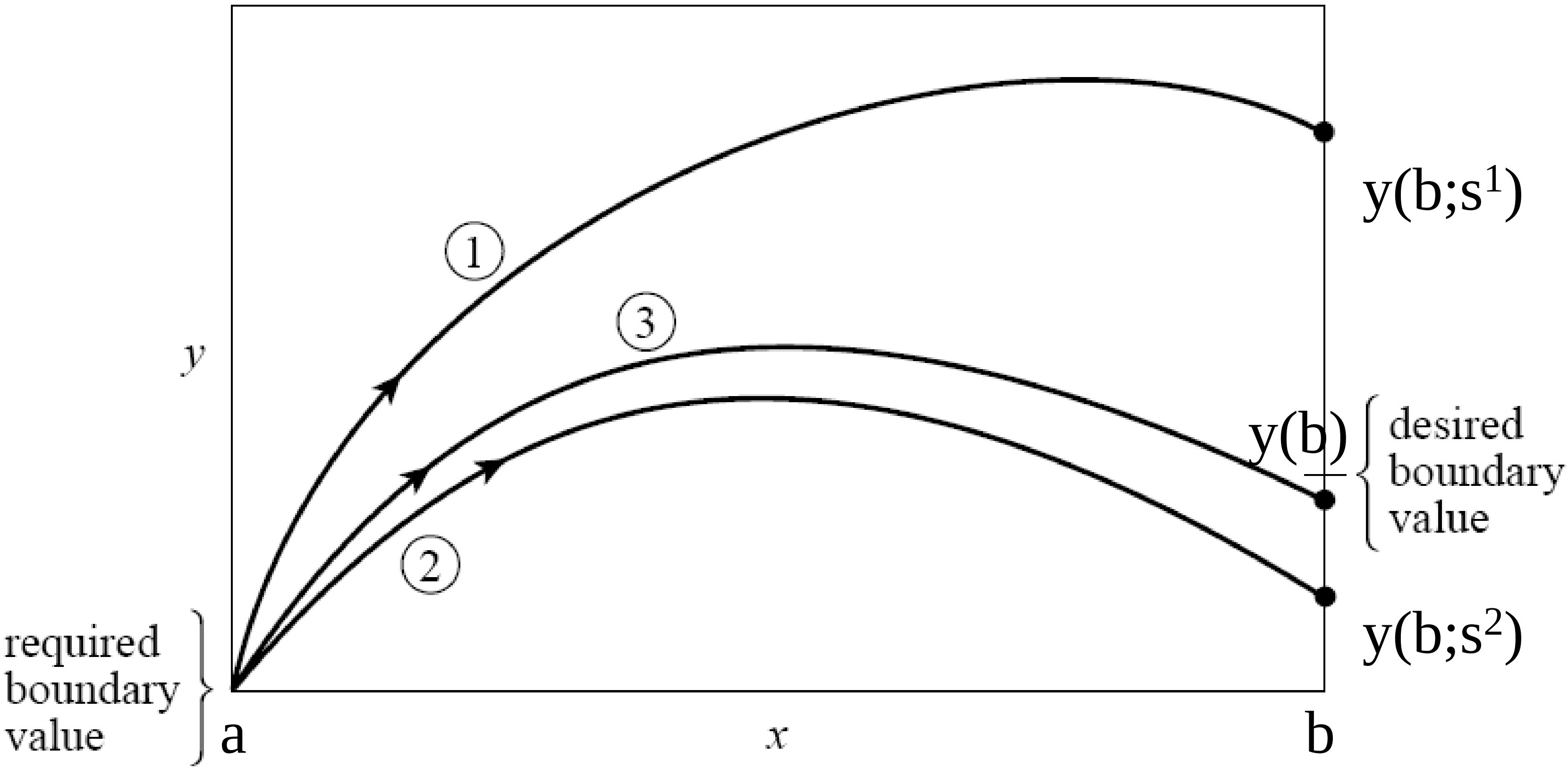


Figure 17.0.1. Shooting method (schematic). Trial integrations that satisfy the boundary condition at one endpoint are "launched." The discrepancies from the desired boundary condition at the other endpoint are used to adjust the starting conditions, until boundary conditions at both endpoints are ultimately satisfied.

Shooting Method

Step 3: iteratively solve the zero-finding problem.

Find the correct s value such that

$$y(b;s)=\beta$$

Satisfy the second BC.

Zero-Finding prolem of the function:

$$y(b;s)=\beta \quad \Leftrightarrow F(s)=y(b;s)-\beta=0$$

Numerical methods for zero-finding (nonlinear systems):

- Bisection (two initial guesses)
- Secant (two initial guesses),
- Newton (one initial guess but it requires to compute F'(x))

Shooting Method based on secant zero-finding method for Step 3

Step 1: solve IVP con $y(a) = \alpha,\ y'(a) = s(1) \quad \Rightarrow \quad \text{Error} = F(1)$

Step 2: solve IVP con $y(a) = \alpha,\ y'(a) = s(2) \quad \Rightarrow \quad \text{Error} = F(2)$

Step 3: secant method to get a new estimate

while $|s(i) - s(i-1)| > tol$

$$s(i) = s(i-1) - \frac{s(i-1) - s(i-2)}{F(i-1) - F(i-2)} F(i-1)$$

solve IVP with $y(a) = \alpha,\ y'(a) = s(i) \quad \Rightarrow \quad \text{Error} = F(i)$

end

Step 4: solve IVP with $y(a) = \alpha,\ y'(a) = s(i)$

Prepare the MATLAB function:

```
function [t,y]=bvpshoot(func,tspan,x0,tol)
t0=tspan(1); tfinal=tspan(2);
ga=x0(1); gb=x0(2);
s(1)=0;s(2)=1;
[t,u]=ode45(funcn, tspan,[ga, s(1)]); F(1)=u(end)-β;
[t,u]=ode45(funcn, tspan,[ga, s(2)]); F(2)=u(end)-β;
i=3;
while (abs(s(i-2)-s(i-1))>tol)
        s(i)=s(i-1)-(s(i-1)-s(i-2))/(F(i-1)-F(i-2)) *F(i-1);
        [t,u]=ode45(funcn, tspan, [ga, s(i)]);
        F(i)=u(end);i=i+1;
end
[t,y]=ode45(funcn, tspan, [ga, s(end)]);
```

Example 1

Original BVP

$$\ddot{y} - 4y + 4x = 0$$

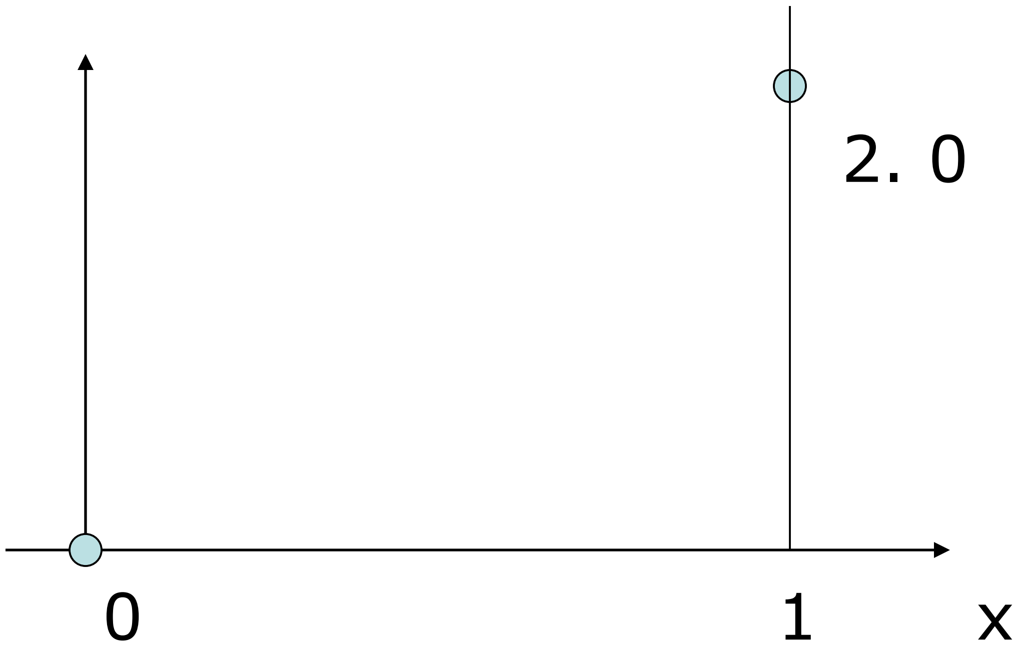
$$y(0) = 0, \quad y(1) = 2$$

Example 1

Original BVP

$$\ddot{y} - 4y + 4x = 0$$

$$y(0) = 0, \quad y(1) = 2$$

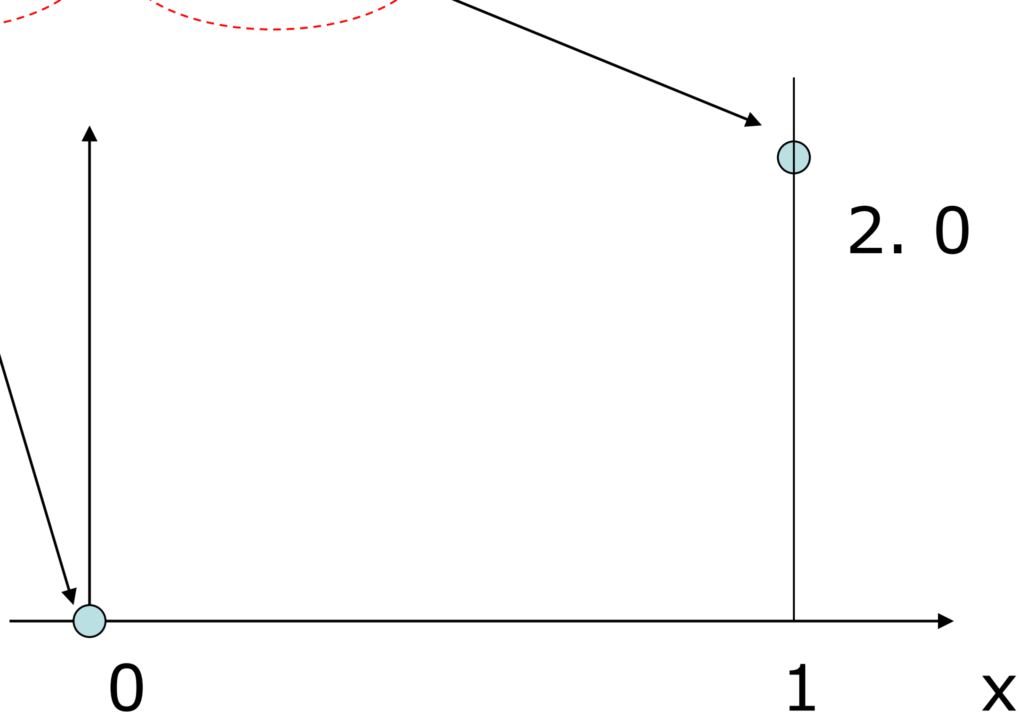


Example 1

Original BVP

$$\ddot{y} - 4y + 4x = 0$$

$$y(0) = 0, \quad y(1) = 2$$

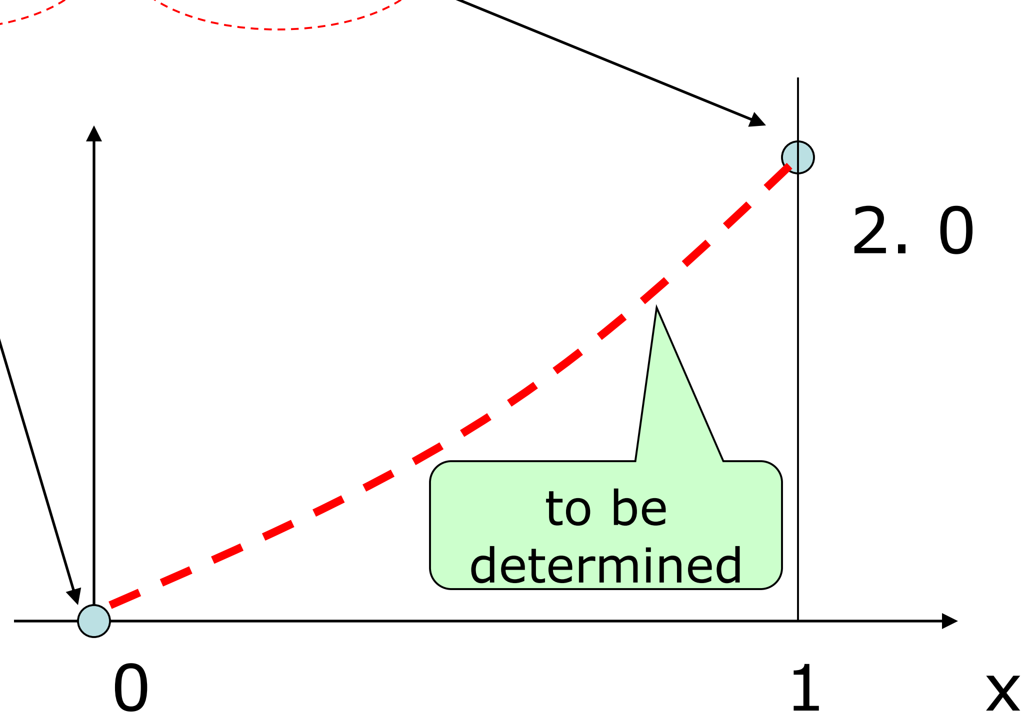


Example 1

Original BVP

$$\ddot{y} - 4y + 4x = 0$$

$$y(0) = 0, \quad y(1) = 2$$



Example 1

Step1 and 2: Convert to a System of First Order ODEs

$$\ddot{y} - 4y + 4x = 0$$

$$y(0) = 0, \quad y(1) = 2$$

Convert to a system of first order Equations

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} y_2 \\ 4(y_1 - x) \end{bmatrix}, \qquad \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ ? \end{bmatrix}$$

The problem will be solved using RK2 with $h = 0.01$

for different values of $y_2(0)$ until we have $y(1) = 2$

Example 1

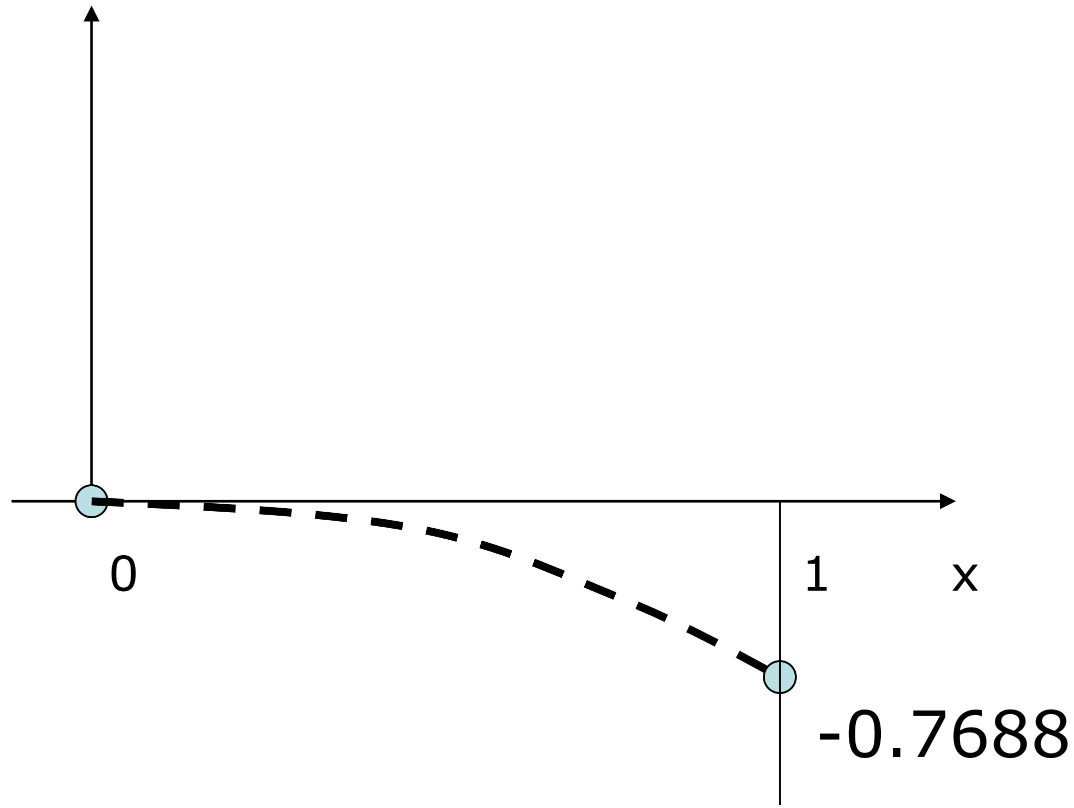
Guess # 1

$$\ddot{y} - 4y + 4x = 0$$

$$y(0) = 0, \quad y(1) = 2$$

$$Guess\#1$$

$$\dot{y}(0) = 0$$



Example 1

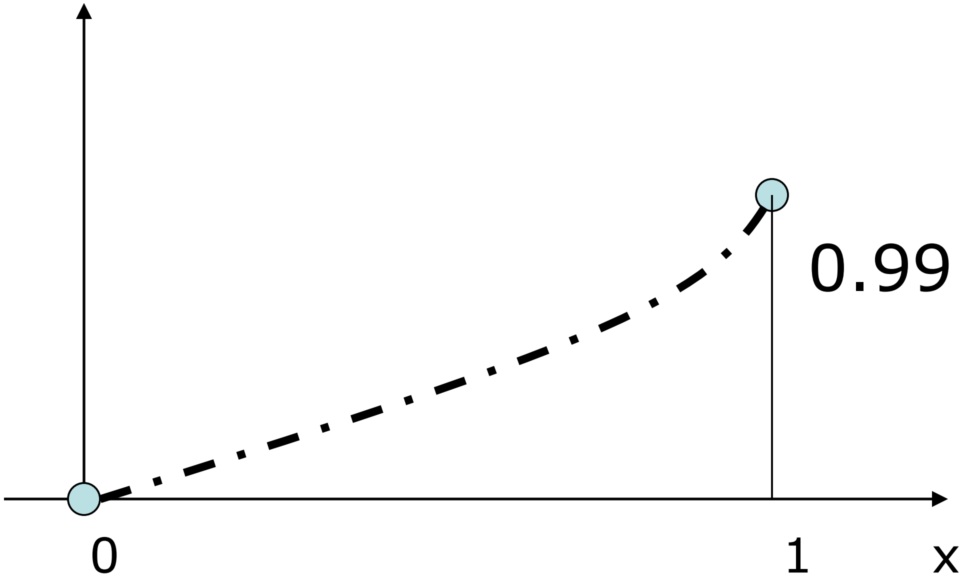
Guess # 2

$$\ddot{y} - 4y + 4x = 0$$

$$y(0) = 0, \quad y(1) = 2$$

$$Guess\#2$$

$$\dot{y}(0) = 1$$



One secant step gives s=1.5743

Example 1

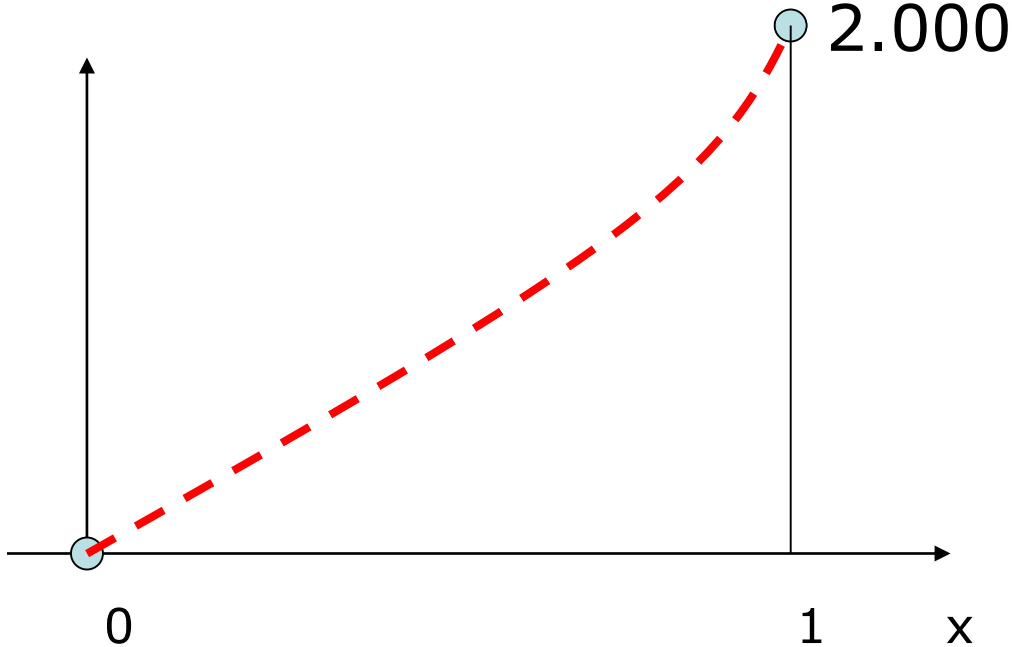
Guess # 3

$$\ddot{y} - 4y + 4x = 0$$

$$y(0) = 0, \quad y(1) = 2$$

$$Guess\#3$$

$$\dot{y}(0) = 1.5743$$



y(1)=2.000

This is the solution to the boundary value problem.

Example: nonlinear BVP

$$\begin{cases} y'' = -2yy', & 0 \le x \le 1 \\ y(0) = 1, & y(1) + y'(1) - 0.25 = 0 \end{cases}$$

$$\text{exact solution} \quad y = 1/(x+1)$$

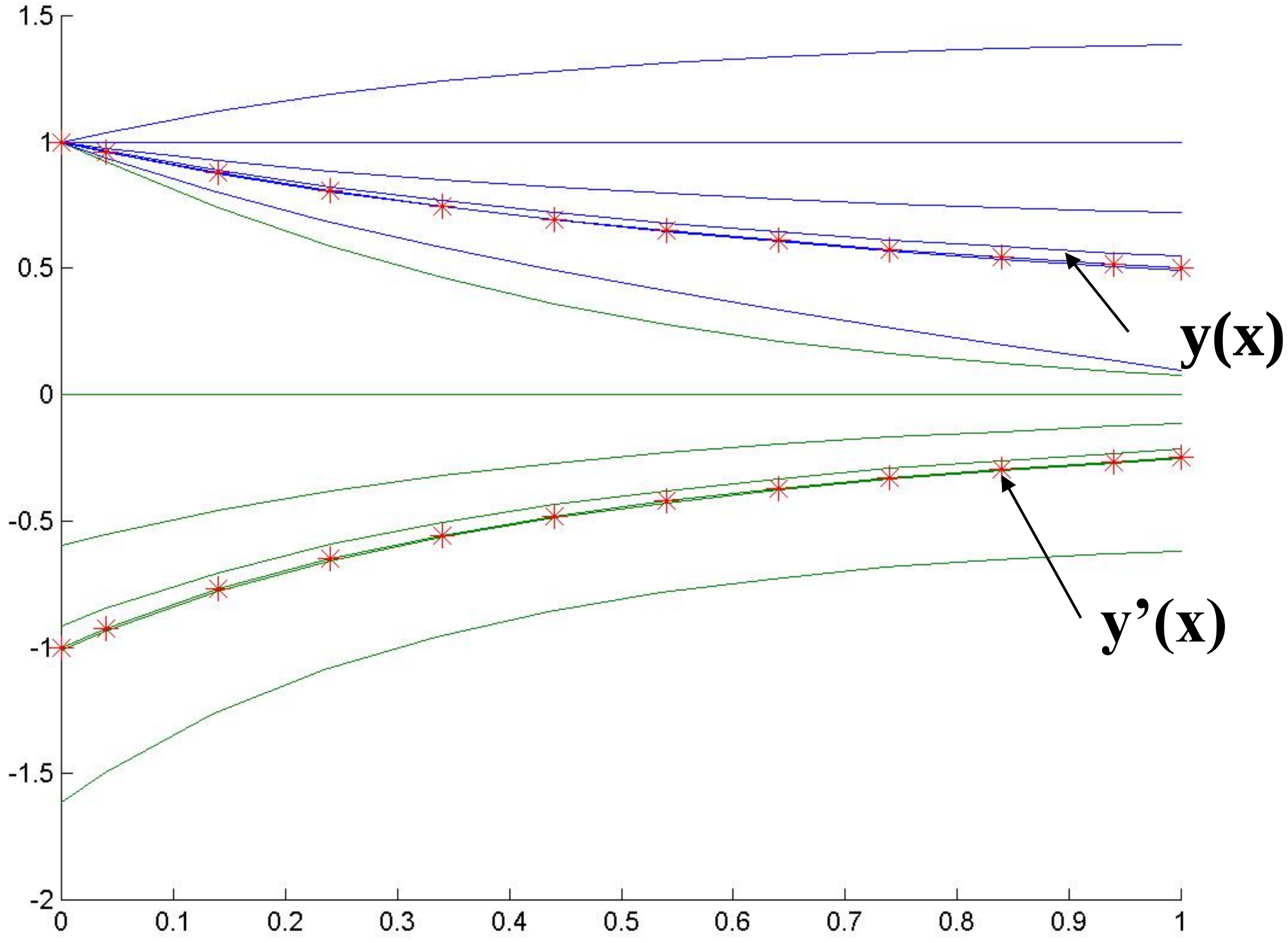
- **Convert to two first-order ODE-IVPs**

$$\text{Let } z_1 = y, \quad z_2 = y'$$

$$\begin{cases} z_1' = z_2, & z_1(0) = 1 \\ z_2' = -2z_1 z_2, & z_2(0) = s \end{cases}$$

- **Update s using the secant method**

Example: nonlinear BVP



Solution of Boundary-Value Problems

Finite Difference Method

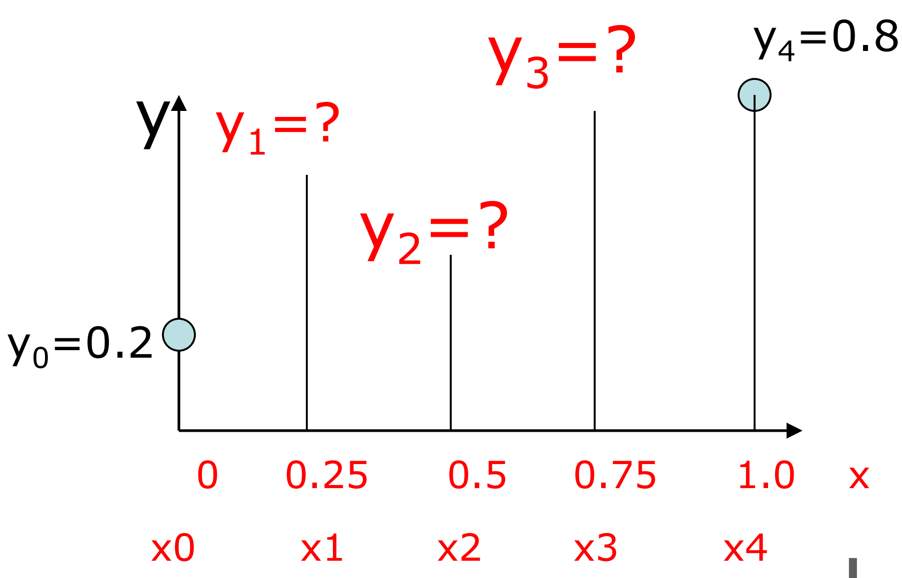
Boundary-Value Problems → convert → Algebraic Equations

Find the unknowns y_1, y_2, y_3

$$Find\ y(x)\ to\ solve\ BVP$$

$$\ddot{y} + 2\dot{y} + y = x^2$$

$$y(0) = 0.2,\quad y(1) = 0.8$$



BVP:
Finite Difference (FD) Method

1. Domain discretization: define a partition of the interval [a,b] in uniform subintervals

$$x_0 = a, x_{n+1} = b, \quad x_i = x_0 + ih, \qquad h = \frac{b-a}{n+1}$$

2. Replace the derivatives by appropriate finite-difference approximations

3. Solve the system of algebraic equations
   (nonlinear system if the BVP is nonlinear)

FD Method for linear BVP

linear ODE-BVPs
$$\begin{cases} y'' = p(x)y' + q(x)y + r(x), & x_0 \le x \le x_{n+1} \\ y(x_0) = a, \quad y(x_{n+1}) = b & (*) \end{cases}$$

- The central finite difference approximation for the first and second derivatives are

$$y'(x_i) = \frac{y_{i+1} - y_{i-1}}{2h} \qquad y''(x_i) = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$$

- By substituting them for the derivatives in the original equation (*) we have a **tridiagonal linear system**

$$\boldsymbol{p}_{\mathrm{i}} \equiv \boldsymbol{p}(\boldsymbol{x}_i), \quad \boldsymbol{q}_i \equiv \boldsymbol{q}(\boldsymbol{x}_i), \qquad \boldsymbol{r}_i \equiv \boldsymbol{r}(\boldsymbol{x}_i)$$

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = p_i \frac{y_{i+1} - y_{i-1}}{2h} + q_i y_i + r_i \qquad i = 1,..,n$$

$$y_0 = a \qquad y_{n+1} = b$$

$$\frac{y_{i+1}-2y_i+y_{i-1}}{h^2}-p_i\frac{y_{i+1}-y_{i-1}}{2h}-q_iy_i=r_i, \qquad i=1,\dots,n$$

$\Updownarrow$

$$\left(-1-\frac{h}{2}p_i\right)y_{i-1}+\left(2+h^2q_i\right)y_i+\left(-1+\frac{h}{2}p_i\right)y_{i+1}=-h^2r_i, \quad i=1,\dots,n$$

Linear System $\mathbf{Ax}=\mathbf{b}$ with

$$\mathbf{A}_{\mathbf{i,i}}=2+h^2q_i, i=1,\dots,n$$

$$\mathbf{A}_{\mathbf{i,i+1}}=-1+\frac{h}{2}p_i, i=1,\dots,n-1$$

$$\mathbf{A}_{\mathbf{i-1,i}}=-1-\frac{h}{2}p_i, i=2,\dots,n$$

Tridiagonal Linear System

$$
\begin{bmatrix}
(2+h^2q_1) & -(1-(h/2)p_1) & 0 & \cdots & 0 \\
-(1+(h/2)p_2) & (2+h^2q_2) & -(1-(h/2)p_2) & \cdots & 0 \\
0 & -(1+(h/2)p_3) & (2+h^2q_3) & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & (2+h^2q_n)
\end{bmatrix}
\begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{Bmatrix}
= -\begin{Bmatrix}
h^2r_1 - (1+(h/2)\mathrm{p}_1)y_0 \\
h^2r_2 \\
h^2r_3 \\
\vdots \\
h^2r_n - (1-(h/2)\mathrm{p}_n)y_{n+1}
\end{Bmatrix}
$$

$y_0 \rightarrow \alpha$

$y_{n+1} \rightarrow \beta$

Case p(x)=0

$$\begin{cases} y'' = q(x)y + r(x)\ , \\ y(x_0) = a,\ \ y(x_{n+1}) = b \end{cases}$$

Discretize by finite difference formulas:

$$y_{i-1} - \left(2 + h^2 q_i\right) y_i + y_{i+1} = h^2 r_i\ , \qquad i = 1, \ldots, n$$

$$\begin{bmatrix} (2+h^2q_1) & -1 & 0 & \cdots & 0 \\ -1 & (2+h^2q_2) & -1 & \cdots & 0 \\ 0 & -1 & (2+h^2q_3) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & (2+h^2q_n) \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} \alpha - h^2 r_1 \\ -h^2 r_2 \\ -h^2 r_3 \\ \vdots \\ \beta - h^2 r_n \end{pmatrix}$$

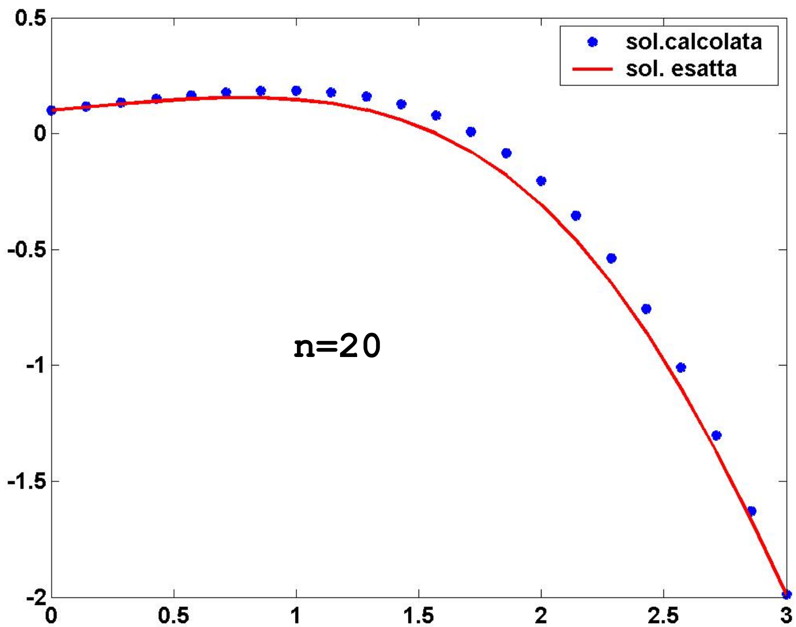
FD for BVP accuracy

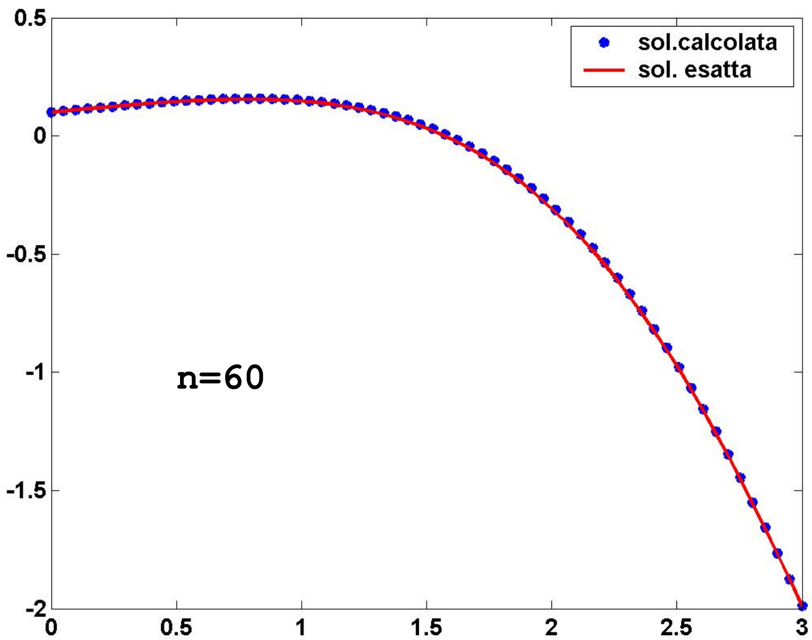
The accuracy of FD numerical methods for BVP can be improved by decreasing the step sizes.

Example

$$\begin{cases} y'' = 2y' - 2y \\ \mathrm{y}(0) = 0.1, \;\; \mathrm{y}(3) = 0.1\exp(3)\cos(3) \end{cases}$$

exact solution: $y(x) = 0.1\exp(x)\cos(x)$





Finite Difference Method Example

$$\ddot{y} + 2\dot{y} + y = x^2$$

$$y(0) = 0.2, \quad y(1) = 0.8$$

Divide the interval [0,1] into n = 4 intervals

Base points are

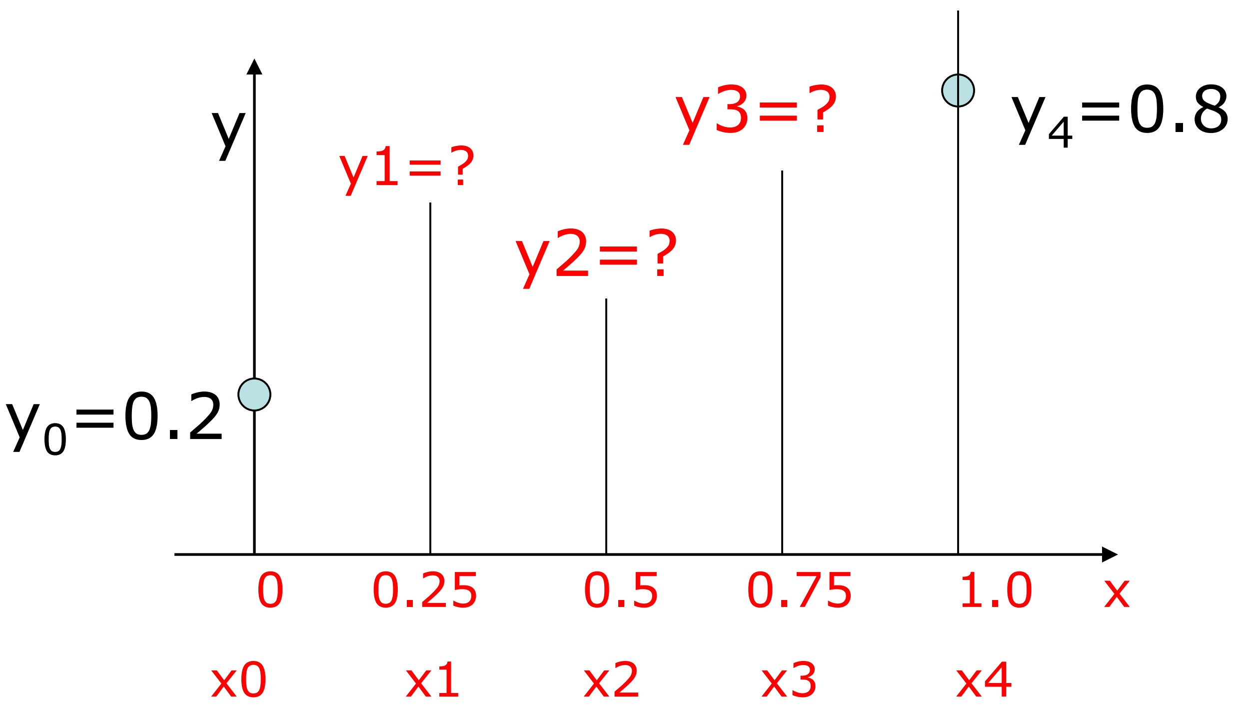
x0=0

x1=0.25

x2=.5

x3=0.75

x4=1.0



Finite Difference Method Example

$$\ddot{y} + 2\dot{y} + y = x^2$$

$$y(0) = 0.2, \quad y(1) = 0.8$$

Divide the interval [0,1] into n = 4 intervals

Base points are

x0=0

x1=0.25

x2=.5

x3=0.75

x4=1.0

Replace

$$\ddot{y} = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} \qquad \textit{central difference formula}$$

$$\dot{y} = \frac{y_{i+1} - y_{i-1}}{2h} \qquad \textit{central difference formula}$$

$$\ddot{y} + 2\dot{y} + y = x^2$$

Becomes

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + 2\frac{y_{i+1} - y_{i-1}}{2h} + y_i = x_i{}^2$$

Finite Difference Method Example

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

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + 2\frac{y_{i+1} - y_i}{h} + y_i = x_i^{\ 2} \quad i = 1, 2, 3$$

$$x_0 = 0,\ x_1 = 0.25, x_2 = 0.5, x_3 = 0.75, x_4 = 1$$

$$h = 1/4$$

$$16\left(y_{i+1} - 2y_i + y_{i-1}\right) + 8\left(y_{i+1} - y_i\right) + y_i = x_i^{\ 2}$$

$$24y_{i+1} - 39y_i + 16y_{i-1} = x_i^{\ 2}$$

Finite Difference Method Example

$$24y_{i+1} - 39y_i + 16y_{i-1} = x_i^{\ 2}$$

$$i = 1 \quad 24y_2 - 39y_1 + 16y_0 = x_1^{\ 2}$$

$$i = 2 \quad 24y_3 - 39y_2 + 16y_1 = x_2^{\ 2}$$

$$i = 3 \quad 24y_4 - 39y_3 + 16y_2 = x_3^{\ 2}$$

$$\begin{bmatrix} -39 & 24 & 0 \\ 16 & -39 & 24 \\ 0 & 16 & -39 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0.25^2 - 16(0.2) \\ 0.5^2 \\ 0.75^2 - 24(0.8) \end{bmatrix}$$

$$Solution \quad y_1 = 0.4791, y_2 = 0.6477, y_3 = 0.7436$$

Finite Difference Method Example

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

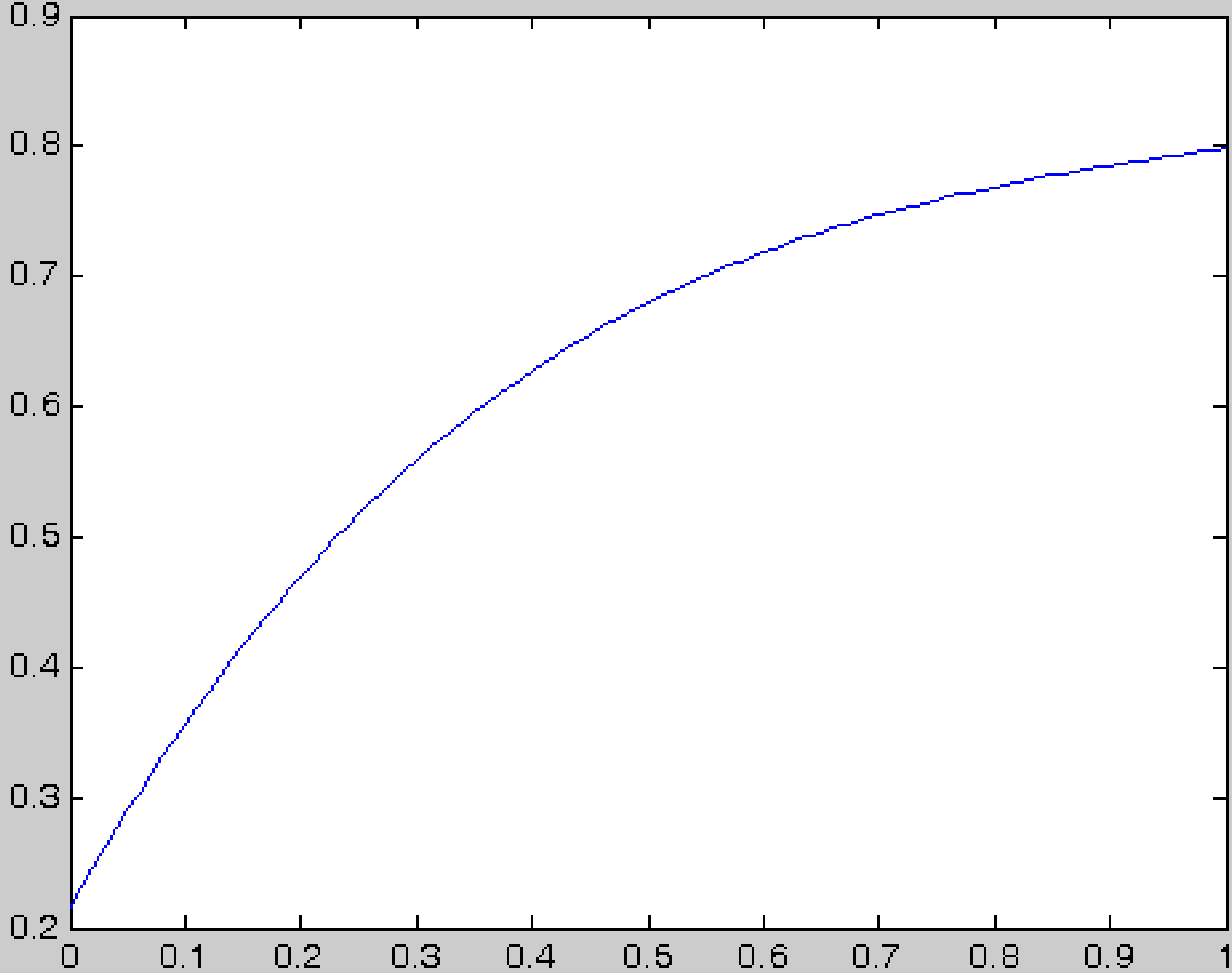
$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + 2\frac{y_{i+1} - y_i}{h} + y_i = x_i^{\,2} \quad i = 1, 2, \ldots, 100$$

$$x_0 = 0,\ x_1 = 0.01, x_2 = 0.02 \quad \ldots \quad x_{99} = 0.99, x_{100} = 1$$

$$h = 1/100$$

$$10000\left(y_{i+1} - 2y_i + y_{i-1}\right) + 200\left(y_{i+1} - y_i\right) + y_i = x_i^{\,2}$$

$$10200\,y_{i+1} - 20199\,y_i + 10000\,y_{i-1} = x_i^{\,2}$$


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Local Truncation Error (LTE) with Centered Finite Difference

$$y'(x_i) = \frac{y(x_{i+1}) - y(x_{i-1})}{2h} - \frac{h^2}{3} y'''(x_i) + \cdots$$

$$y''(x_i) = \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} + \frac{h^2}{12} y^{(iv)}(x_i) + \cdots$$

$$y_i \equiv y(x_i)$$

$$y'_i \equiv \frac{y_{i+1} - y_{i-1}}{2h}$$

$$y''_i \equiv \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$$

⇒

$$y'(x_i) = y'_i + O\left(\frac{h^2}{3}\right)$$

$$y''(x_i) = y''_i + O\left(\frac{h^2}{12}\right)$$

The method is consistent of order 2. ($O(h^2)$)

Continuos Model (case p=0)

- Replacing the relations we get:

$$\begin{cases} y_0 = \alpha \\ y_{i-1} - \left(2 + h^2 q_i\right) y_i + y_{i+1} = h^2 r_i + \boxed{\tau_i(y),} \qquad i = 1, \dots, n \\ y_n = \beta \end{cases}$$

$$\mathbf{A_h} \equiv \frac{1}{\mathbf{h}^2} \begin{bmatrix} (2+h^2 q_1) & -1 & 0 & \cdots & 0 \\ -1 & (2+h^2 q_2) & -1 & \cdots & 0 \\ 0 & -1 & (2+h^2 q_3) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & (2+h^2 q_n) \end{bmatrix} \quad b_h = \begin{pmatrix} \alpha / h^2 - r_1 \\ -r_2 \\ -r_3 \\ \vdots \\ \beta / h^2 - r_n \end{pmatrix}$$

- Discrete Model: $\mathbf{A_h} \boldsymbol{y_h} = \boldsymbol{b_h}$

- Continuous Model: $\mathbf{A_h} \bar{\boldsymbol{y}} = \boldsymbol{b_h} + \boldsymbol{\tau}(\boldsymbol{y})$
- Issues:
  - Existence and uniqueness of the solution $\boldsymbol{y_h}$
  - Numerical calculation of $\boldsymbol{y_h}$
  - Conditioning of the matrix $\boldsymbol{A_h}$
  - Error estimate $\boldsymbol{y_h} - \bar{\boldsymbol{y}}$

Existence and uniqueness of the solution

Properties of the matrix A_h

- A_h is symmetric and definite positive
- The elements of A_h^{-1} are non neg. $\left(A_h^{-1} \geq 0\right)$
- $q_i \geq 0 \Rightarrow$ the matrix is diagonal dominant:

$$\sum_{j=1}^{n} a_{i,j} = h^2 q_i$$

- The matrix is sparse tridiagonal

Solve the linear system using:

- Direct methods: LU,
- Iterative methods: SOR, Conjugate Gradient

Convergence (consistent + stable)

- Global error: $\boldsymbol{e_h = \bar{y} - y_h}$

$$\mathbf{A_h} \boldsymbol{y_h = b_h}$$

$$\mathbf{A_h} \boldsymbol{\bar{y} = b_h + \tau(y)} \quad \Rightarrow \quad \boxed{\boldsymbol{A_h e_h = \tau(y)}}$$

$$\|e_h\| = \|A_h^{-1}\tau(y)\| \le \|A_h^{-1}\|\|\tau(y)\|$$

- Consistency is related to $\|\tau(y)\|$
- Stability is related to find an upper bound of $\|\boldsymbol{A_h}^{-1}\|$

Theorem: If $\boldsymbol{y \in C^4_{[a,b]}}$ the system is convergent for h->0 and

$$\|e_h\|_\infty = \max_{1 \le i \le n} |y(x_i) - y_h(x_i)| \le \frac{h^2}{24} \|y^{(4)}\|_\infty \|(x-a)(b-x)\|_\infty$$

Neumann BCs

$$y'' = f(x, y, y'), \qquad a \le x \le b$$
$$y'(a) = \alpha, \quad y'(b) = \beta$$

- imposing Neumann boundary conditions at both ends gives an ill-posed problem that has either no solution or infinitely many solutions.
- Ex: heat flux at a specified rate giving y'(a)=α at this boundary...
- **Homogeneous Neumann BC α=β=0 , f(x)=0**, both ends of the rod are insulated, there is no heat flux through the ends, and there is no heat source within the rod, The BVP has <u>infinitely many solutions</u> in this case y(x)=c, constant c

Neumann BCs

$$y'' = f(x, y, y'), \qquad a \le x \le b$$
$$y'(a) = \alpha, \quad y'(b) = \beta$$

- **Homogeneous Neumann BC α=β=0 , f(x)<0**, both ends of the rod are insulated, there is no heat flux through the ends, and we are constantly adding heat to the rod. Since no heat can escape through the insulated ends, we expect the temperature to keep rising without bound. In this case we never reach a steady state, and the BVP has <u>no solution</u>.
- **Homogeneous Neumann BC α=β=0 ,** f1(x)<0 half interval and f2(x)>0 half interval and f1=f2 → ?

Non-homogeneous Neumann BC

$$y'' = f(x), \qquad a \le x \le b$$

$$y'(a) = \sigma, \qquad y(b) = \beta$$

$$x_0 = a, x_{n+1} = b \qquad y'(a) = \frac{y(x_1) - y(x_0)}{h} + O(h)$$

$$y_{i-1} - 2\,y_i + y_{i+1} = h^2 f_i, \qquad i = 1, \ldots, n$$

$$y_1 - y_0 = h\sigma$$

$$y_{n+1} = \beta$$

Extra equation for the first node

$$\frac{1}{h^2}\begin{bmatrix} -h & h & & & \\ 1 & -2 & 1 & & \\ & .. & .. & .. & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & \end{bmatrix}\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} = \begin{pmatrix} \sigma \\ f(x_1) \\ \vdots \\ f(x_{n-1}) \\ f(x_n) - \dfrac{\beta}{h^2} \end{pmatrix}$$

Non-homogeneous Neumann BC

$$y'' = f(x), \qquad a \le x \le b$$

$$y'(a) = \sigma, \qquad y(b) = \beta$$

$$x_0 = a, x_{n+1} = b \qquad y'(a) = \frac{y(x_1) - y(x_{-1})}{2h} + O(h^2)$$

$$y_{i-1} - 2\,y_i + y_{i+1} = h^2 f_i, \qquad i = 0,\dots,n$$

$$y_1 - y_{-1} = 2h\sigma$$

$$y_{n+1} = \beta$$

$$y_{-1} - 2\,y_0 + y_1 = h^2 f_0, \qquad y_1 - y_{-1} = 2h\sigma$$

$$\Rightarrow \frac{1}{h}(-y_0 + y_1) = \sigma + \frac{h}{2} f_0$$

To ensure the accuracy of order $O(h^2)$ we use centered formula

$$\frac{1}{h^2}\begin{bmatrix} -h & h & & & \\ 1 & -2 & 1 & & \\ & .. & .. & .. & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & \end{bmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} = \begin{pmatrix} \sigma + \frac{h}{2} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{n-1}) \\ f(x_n) - \frac{\beta}{h^2} \end{pmatrix}$$

Robin BC

$$y'' = f(x, y, y'), \qquad a \le x \le b$$

$$y'(a) + c_1 y(a) = 0, \qquad y'(b) + c_2 y(b) = 0$$

$$x_0 = a, x_{n+1} = b$$

To ensure the accuracy of order $O(h^2)$ we use centered formula:

$$y'(a) = \frac{(y(x_1) - y(x_{-1}))}{2h} + O(h^2)$$

$$y'(b) = \frac{(y(x_{n+2}) - y(x_n))}{2h} + O(h^2)$$

Introduce two ghost points x_{-1} and x_{n+2} and approximate the BVP in x_0 and x_{n+1}. Extended linear system.

FD Method for Nonlinear BVP

- Nonlinear ODE-BVPs

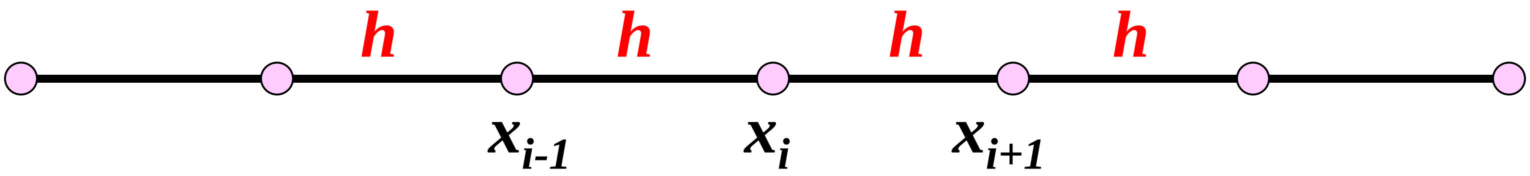
$$\begin{cases} y'' = f(x, y, y'), \quad a \le x \le b \\ y(a) = \alpha, \quad y(b) = \beta \end{cases}$$

- Evaluate f_i by appropriate finite-difference approximations

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - f\left(x_i, y_i, \left(\frac{y_{i+1} - y_{i-1}}{2h}\right)\right) = 0 \quad i = 1, .., N$$

$$y_0 = \alpha, \quad y_{N+1} = \beta$$

- We get a system of linear/nonlinear equations



Example 1: nonlinear BVP

$$\begin{cases} y'' = \frac{1}{2}(1+x+y)^3, \quad 0 \le \text{x} \le 1 \\ \text{y}(0) = 0, \quad \text{y}(1) = 0 \end{cases}$$

- Discretize using a uniform grid

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - \frac{1}{2}(1 + x_i + y_i)^3 = 0 \quad i = 1,..,n$$

$$y_0 = 0, \quad y_{n+1} = 0$$

- In Matrix-Vector form the system is : $Ay + h^2 B(y) = 0$

$$A = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{bmatrix}$$

$$B(y) = diag(f(x_i, y_i)) \; i = 1,..,n$$

Example 2: nonlinear BVP

$$\begin{cases} y'' = f(x, y, y') = -(y')^2 / y, \\ \quad y(0) = 1, \quad y(1) = 2 \end{cases}$$

- Discretize using a uniform grid

$$\frac{y_{i-1} - 2\, y_i + y_{i+1}}{h^2} = f_i \qquad i = 1, .., n$$

$$f_i = -\frac{\left[(y_{i+1} - y_{i-1}) / 2h\right]^2}{y_i} = -\frac{(y_{i+1} - y_{i-1})^2}{4h^2 y_i}$$

- In Matrix-Vector form the system is :

$Ay + h^2 B(y) = 0$ Non-linear system → Newton Method

- Starting from $y^{(0)}$ (initial estimate of the solution), solve by Newton Method iteratively for y

$$
\begin{array}{l}
\text{Initial guess } y^{(0)} \\
\textit{repeat} \\
J(y^{(k)})s^{(k)} = -F(y^{(k)}) \\
y^{(k+1)} = y^{(k)} + s^{(k)} \\
\textit{until} \quad \| y^{(k+1)} - y^{(k)} \| < tol
\end{array}
$$

- Jacobian matrix:

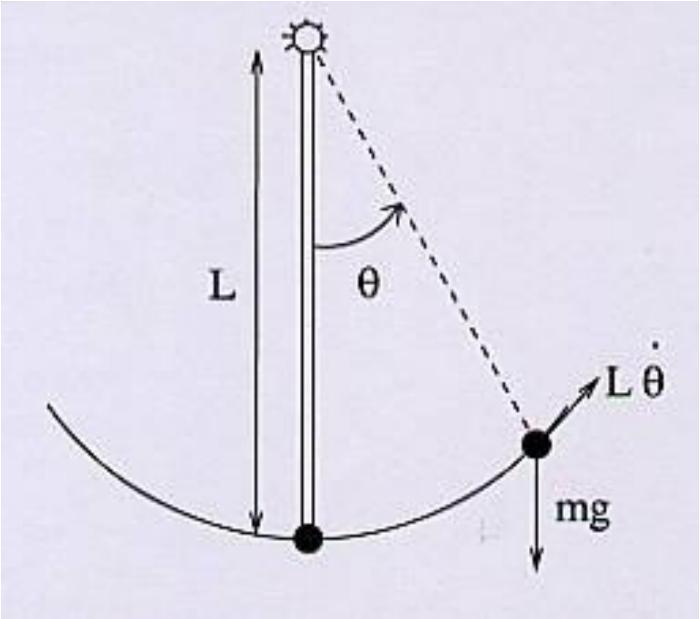
$$J = A + h^2 B_y \qquad B_y = \frac{\partial f}{\partial y}$$

$$(A + h^2 B_y(y^{(k)}))s^{(k)} = -(Ay^{(k)} + h^2 B(y^{(k)}))$$

- Since $B_y \geq 0$, J is def. pos. diagonal dominant, then the Newton Method is convergent for any initial guess.

Equation of motion of a Pendulum (no friction)

L is the length of the pendulum and ***g*** is the local acceleration of gravity, mass m.



Assume $\omega^2=1$

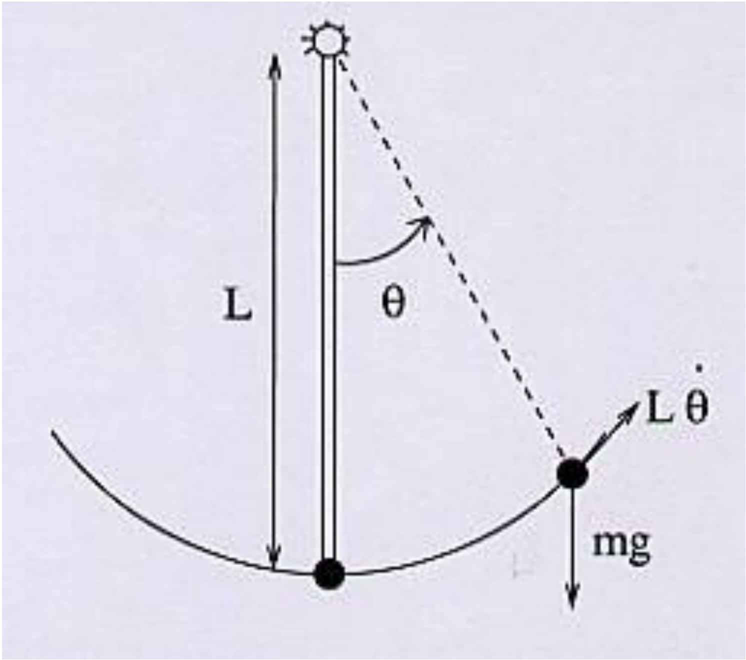
Temporal evolution of the amplitude ϴ (width of swing) formed by the pendulum with the vertical to the pivot.

$$\begin{cases} \ddot{\theta} + \omega^2 \sin\theta = 0 \quad \text{con} \quad \omega^2 = \dfrac{g}{L} \\ \theta(0) = \theta_0 \\ \dot{\theta}(0) = \dot{\theta}_0 \end{cases}$$

IVP

Equation of motion of a Pendulum (no friction)

L is the length of the pendulum and ***g*** is the local acceleration of gravity, mass m.



Assume $\omega^2=1$

The pendulum starts from a predetermined position α with initial angular velocity not known, and

oscillates until reaching the position β at a specified time T

$$\begin{cases} \ddot{\theta} + \omega^2 \sin\theta = 0 \\ \theta(0) = \alpha \\ \theta(T) = \beta \end{cases}$$

BVP

Equation of motion of a Pendulum (no friction) Nonlinear BVP

$$\begin{cases} \ddot{\theta} + \omega^2 \sin\theta = 0 \\ \theta(0) = \alpha; \quad \theta(T) = \beta \end{cases}$$

Discretize using a uniform grid

$$\underbrace{\frac{\vartheta_{i-1} - 2\,\vartheta_i + \vartheta_{i+1}}{h^2} + \sin(\vartheta_i) = 0}_{F(\theta)=0} \qquad i = 1,..,n$$

$$\vartheta_0 = \alpha, \quad \vartheta_{n+1} = \beta$$

Non linear System →Newton Method

Initial guess $\vartheta^{(0)}$

repeat

$J(\vartheta^{(k)})s^{(k)} = -F(\vartheta^{(k)})$

$\vartheta^{(k+1)} = \vartheta^{(k)} + s^{(k)}$

until $\|\vartheta^{(k+1)} - \vartheta^{(k)}\| < tol$

$$J_{ij}(\vartheta) = \frac{\partial}{\partial \vartheta_j} F_i(\vartheta)$$

$$J_{ij}(\vartheta) = \begin{cases} 1/h^2 & \text{if } j = i\text{-}1 \text{ or } j = i+1 \\ -2/h^2 + \cos(\vartheta_i) & \text{if } j = i \\ 0 & else \end{cases}$$

$$J(\vartheta)=\frac{1}{h^2}\begin{bmatrix}(-2+h^2\cos(\vartheta_1)) & 1 & & & \\ 1 & (-2+h^2\cos(\vartheta_2)) & 1 & & \\ & 1 & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & (-2+h^2\cos(\vartheta_N))\end{bmatrix}$$

Initial guess $\vartheta^{(0)}$

repeat

$$J(\vartheta^{(k)})s^{(k)}=-F(\vartheta^{(k)})$$

Solve a tridiagonal linear system at each iteration k

$$\vartheta^{(k+1)}=\vartheta^{(k)}+s^{(k)}$$

until $\|\vartheta^{(k+1)}-\vartheta^{(k)}\|<tol$

Function MATLAB bvp4c() for BVP problems

- Test Problem $\begin{cases} y'' + |y| = 0 \\ y(0) = 0 \qquad y(4) = -2 \end{cases}$
- Convert into two ODE-IVP first order

$$\begin{cases} y_1' = y_2 \\ y_2' = -|y_1| \end{cases}$$

Boundary conditions:

$$bc\big(y(a), y(b)\big) = 0$$

$$ya(1) = y(a), ya(2) = y'(a)$$

$$yb(1) = y(b), yb(2) = y'(b)$$

```
function dydx = twoode(t,y)
  dydx = [ y(2)
         -abs(y(1))];
```

```
function res = twobc(ya,yb)
  res = [ ya(1)
         yb(1) + 2];
```

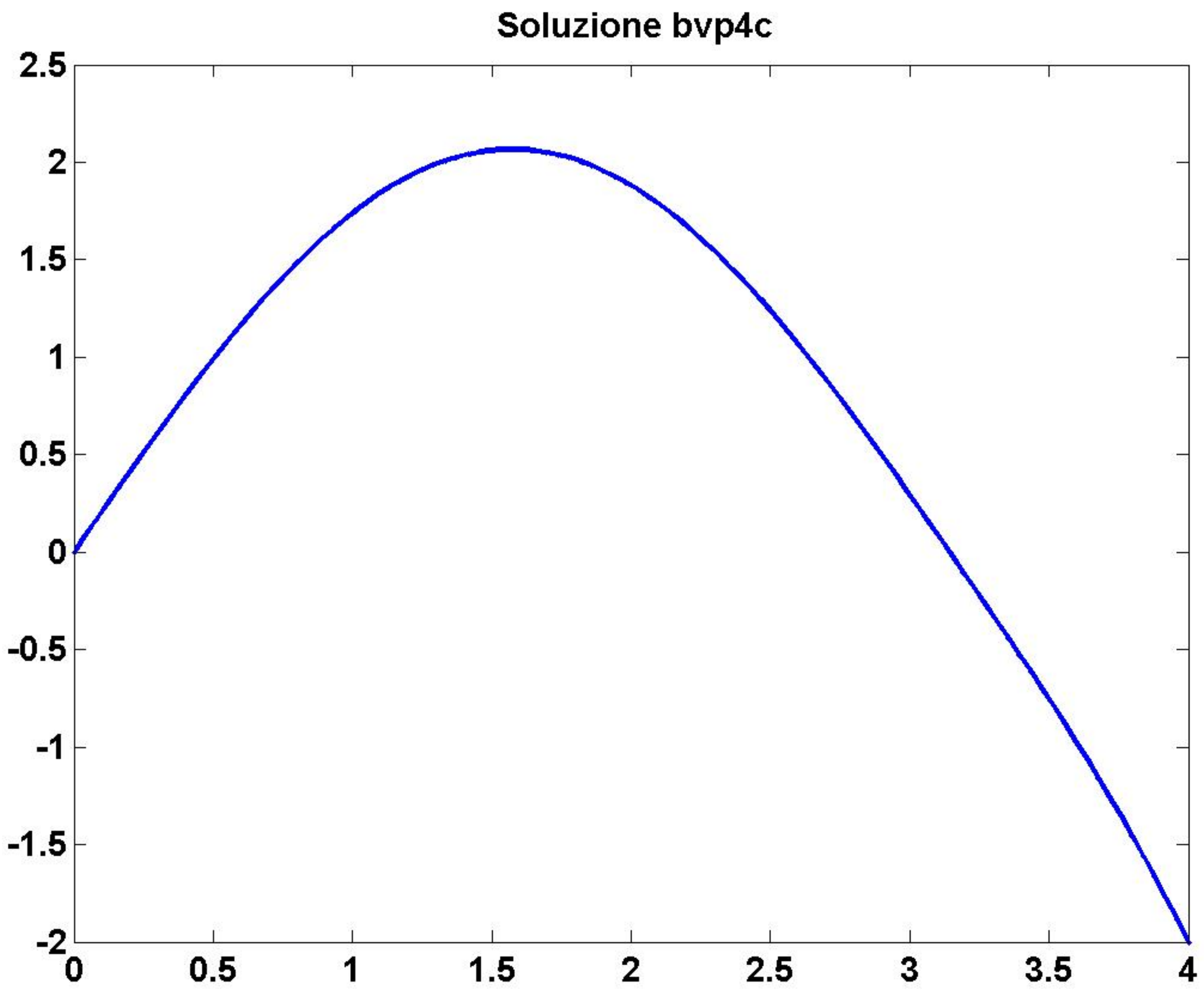
```
function testbvp
solinit = bvpinit(linspace(0,4,5),[1 0]);
sol = bvp4c(@twoode,@twobc,solinit);
x = linspace(0,4);
y = deval(sol,x);
plot(x,y(1,:));
```

- **deval** evaluates the solution at points x (sol structure contains the solution)

```
sol =
        x: [1x22 double]
        y: [2x22 double]
       yp: [2x22 double]
   solver: 'bvp4c'
```

- **bvpinit(x,yinit)** forms the initial guess for a boundary value problem solver.
  x is a vector that specifies an initial mesh (5 points uniformly distributed in [0,4]).
  yinit is a guess for the solution.
  It can be either a vector, or a function.
  (y1(x)=1 and y2(x)=0)

```
solinit =
   x: [0 1 2 3 4]
   y: [2x5 double]
```


Soluzione bvp4c
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
0.5
1
1.5
2
2.5
3
3.5
4

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Serena Morigi

Dipartimento di Matematica

serena.morigi@unibo.it

http://www.dm.unibo.it/~morigi