上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

Lecture 9

Boundary Value Problems

Ye Ding (丁 烨)

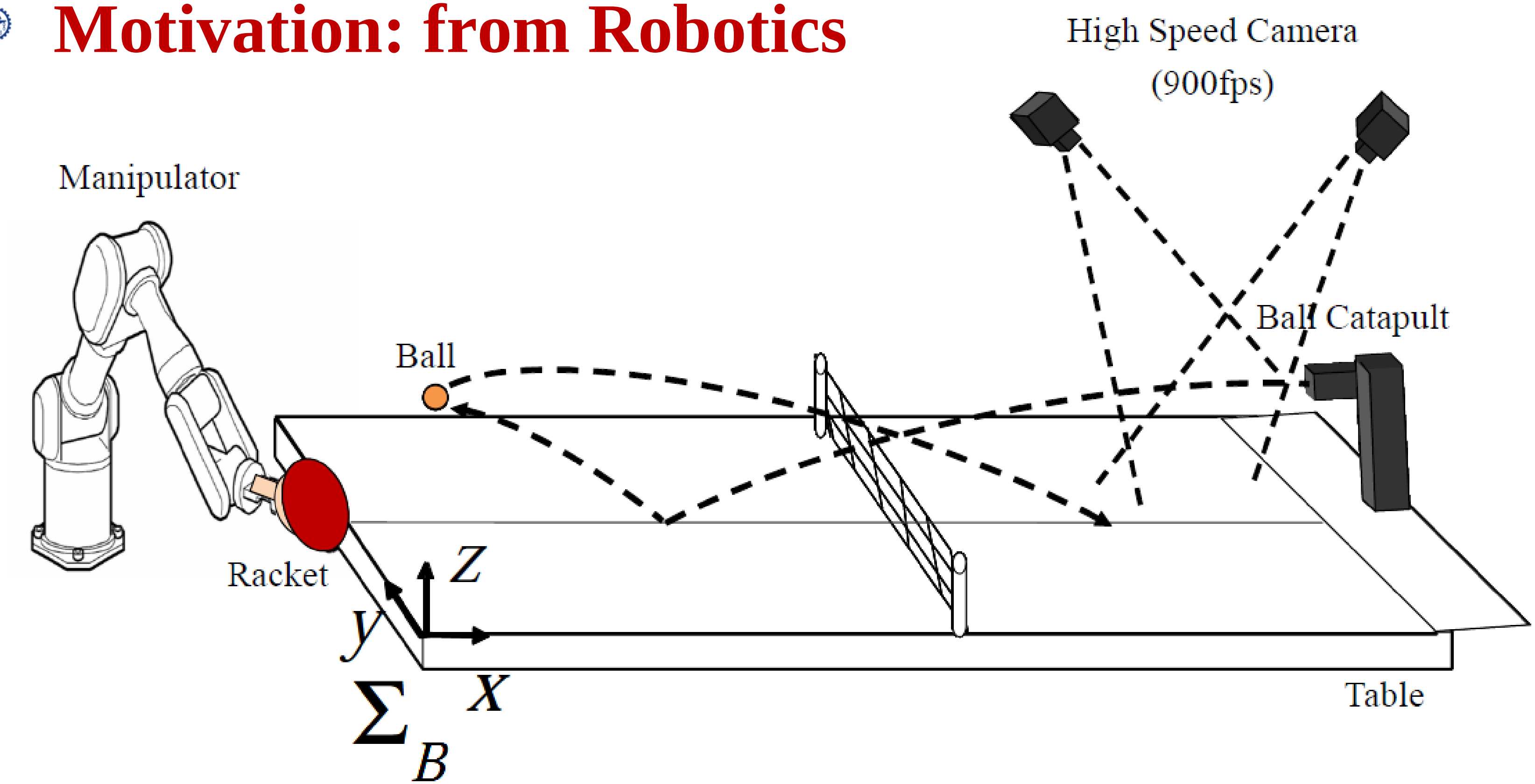
Email: y.ding@sjtu.edu.cn

School of Mechanical Engineering

Shanghai Jiao Tong University

Boundary Value Problems

- **Motivation: from Robotics**



$$\begin{cases} \ddot{\boldsymbol{p}}(t) = \boldsymbol{f}(t, \boldsymbol{p}, \dot{\boldsymbol{p}}), \\ \boldsymbol{p}(t_1) = \boldsymbol{p}_1 \in \mathbb{R}^n, \ \boldsymbol{p}(t_2) = \boldsymbol{p}_2 \in \mathbb{R}^n, \ t_1 < t_2 \in \mathbb{R} \end{cases}$$

Boundary Value Problems

- **References for Boundary Value Problems**

  [1] Timothy Sauer, Numerical analysis (2nd ed.), Pearson Education, 2012. Chapter 7

  [2] Richard L. Burden, J. Douglas Faires, Numerical analysis (9th ed.), Brooks/Cole, 2011. Chapter 11

Boundary Value Problems

- **Boundary Value Problems (BVP)**

A general second-order boundary value problem on a specific interval $a \le t \le b$:

$$
\begin{cases}
y'' = f(t, y, y') \\
y(a) = y_a \\
y(b) = y_b
\end{cases}
$$

Boundary Value Problems

- Symbolic Computation
- Shooting Method
- Finite Difference Methods
  - Linear boundary value problems
  - Nonlinear boundary value problems
- Finite Element Method

Symbolic Computation

Ordinary Differential Equations

```
dsolve(eqn,cond)
```

to solve the ordinary differential equation **eqn** with the initial or boundary condition **cond**

Symbolic Computation

- **Ordinary Differential Equations: Example 1**

$$y'' = -y + 2\cos t$$

```
>> syms t y(t)
>> y1 = dsolve(diff(y, 2) == -y + 2 * cos(t))
```

Symbolic Computation

- **Ordinary Differential Equations: Example 1**

$$\begin{cases} y'' = -y + 2\cos t \\ y(0) = 0 \\ y(\pi) = 0 \end{cases}$$

```
>> syms t y(t)
>> y2 = dsolve(diff(y, 2) == -y + 2 * cos(t),
   y(0)==0,y(pi) == 0)
```

Symbolic Computation

- **Ordinary Differential Equations: Example 2**

$$\begin{cases} y'' = -y \\ y(0) = 0 \\ y(\pi) = 1 \end{cases}$$

```
>> syms t y(t)
>> S = dsolve(diff(y, 2) == -y,y(0) == 0,y(pi) ==1)

Warning: Explicit solution could
not be found.
```

Symbolic Computation

- **Ordinary Differential Equations: Example 2**

$$\begin{cases} y'' = -y \\ y(0) = 0 \\ y(\pi) = 0 \end{cases}$$

```
>> syms t y(t)
>> S = dsolve(diff(y, 2) == -y,y(0) == 0,y(pi) ==0)
```

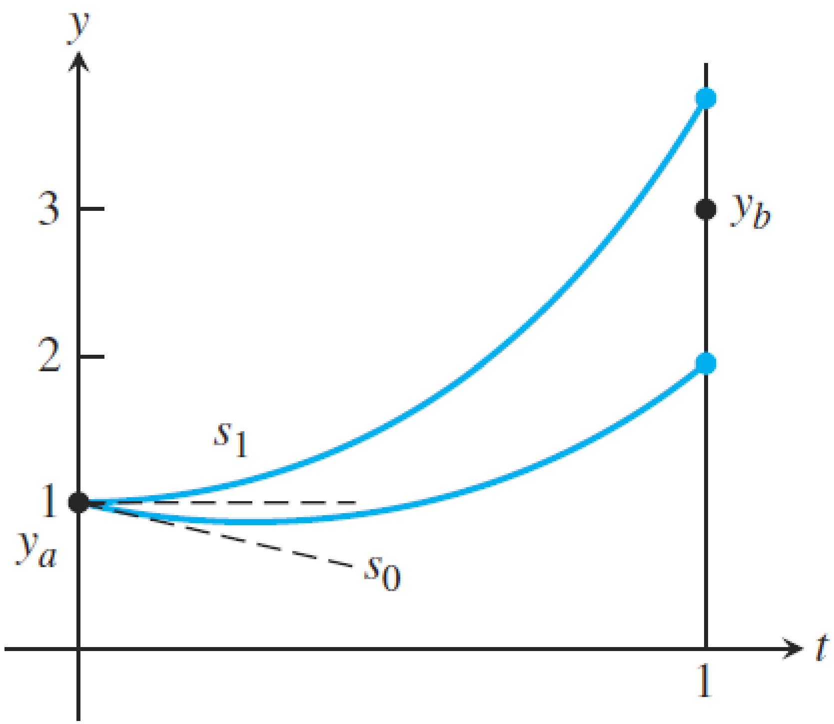
Numerical Computation

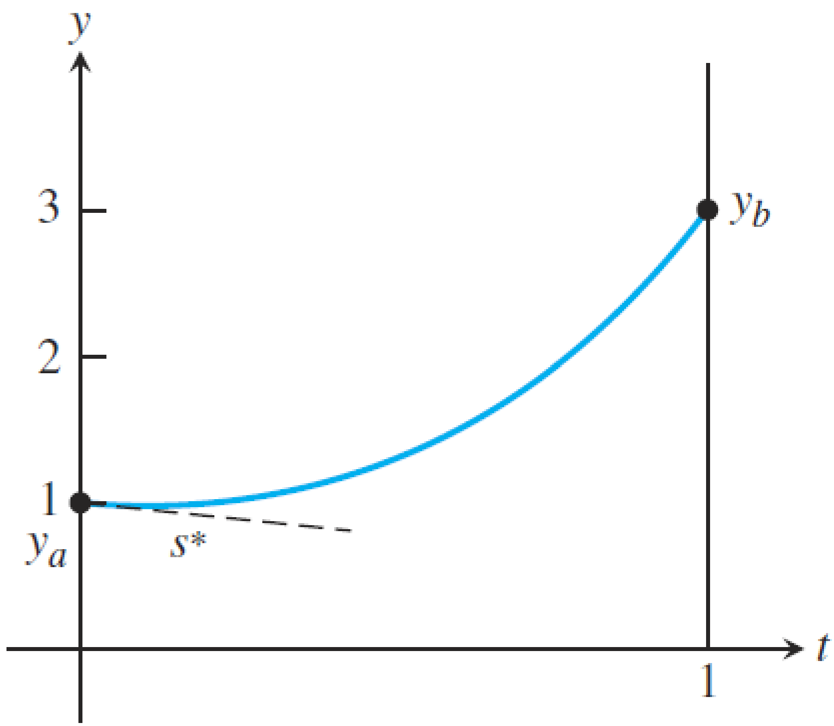
- Shooting Method
- Finite Difference Methods
  - Linear boundary value problems
  - Nonlinear boundary value problems
- Finite Element Method

Numerical Computation

The Shooting Method: Basic Idea

$$\begin{cases} y'' = f(t, y, y') \\ y(a) = y_a \\ y(b) = y_b \end{cases} \Rightarrow \begin{cases} y'' = f(t, y, y') \\ y(a) = y_a \\ y'(a) = s^* \end{cases}$$





Numerical Computation

The Shooting Method: Implementation

- **The BVP is reduced to solving the equation:**

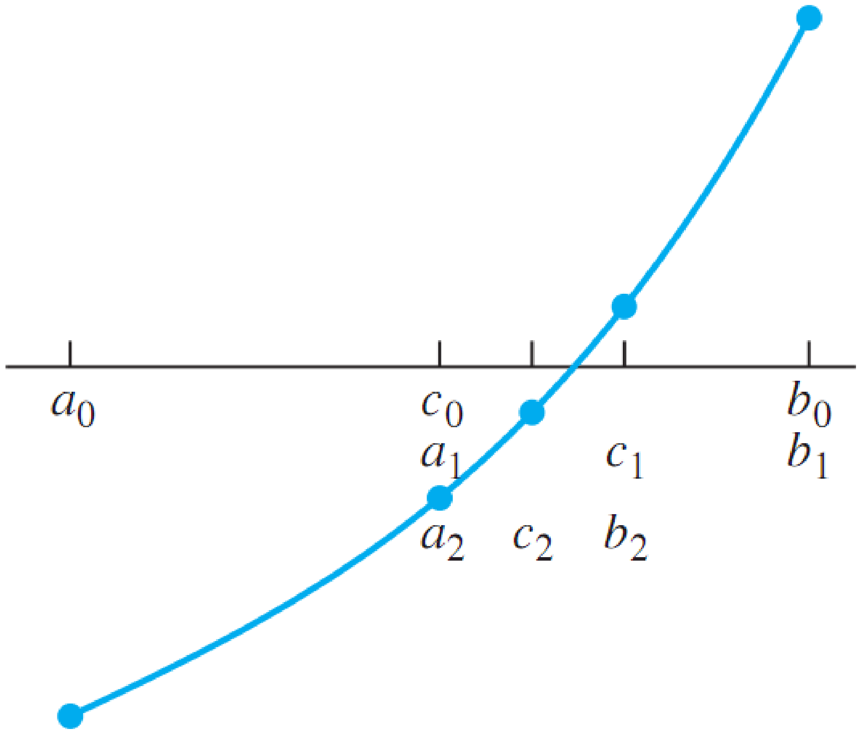
$$F(s) = 0$$

$$F(s) = \begin{cases} \text{difference between } y_b \text{ and} \\ y(b), \text{ where } y(t) \text{ is the} \\ \text{solution of the IVP with} \\ y(a) = y_a \text{ and } y'(a) = s. \end{cases}$$

- **The Bisection Method can be used to solve it.**

Numerical Computation

Bisection Method: Review



Step 1: the sign of $f(c_0)$ is checked. Since $f(c_0)f(b_0)<0$, set $a_1 = c_0$, $b_1 = b_0$. $[a_0, b_0] \to [a_1, b_1]$

Step 2: $[a_1, b_1] \to [a_2, b_2]$

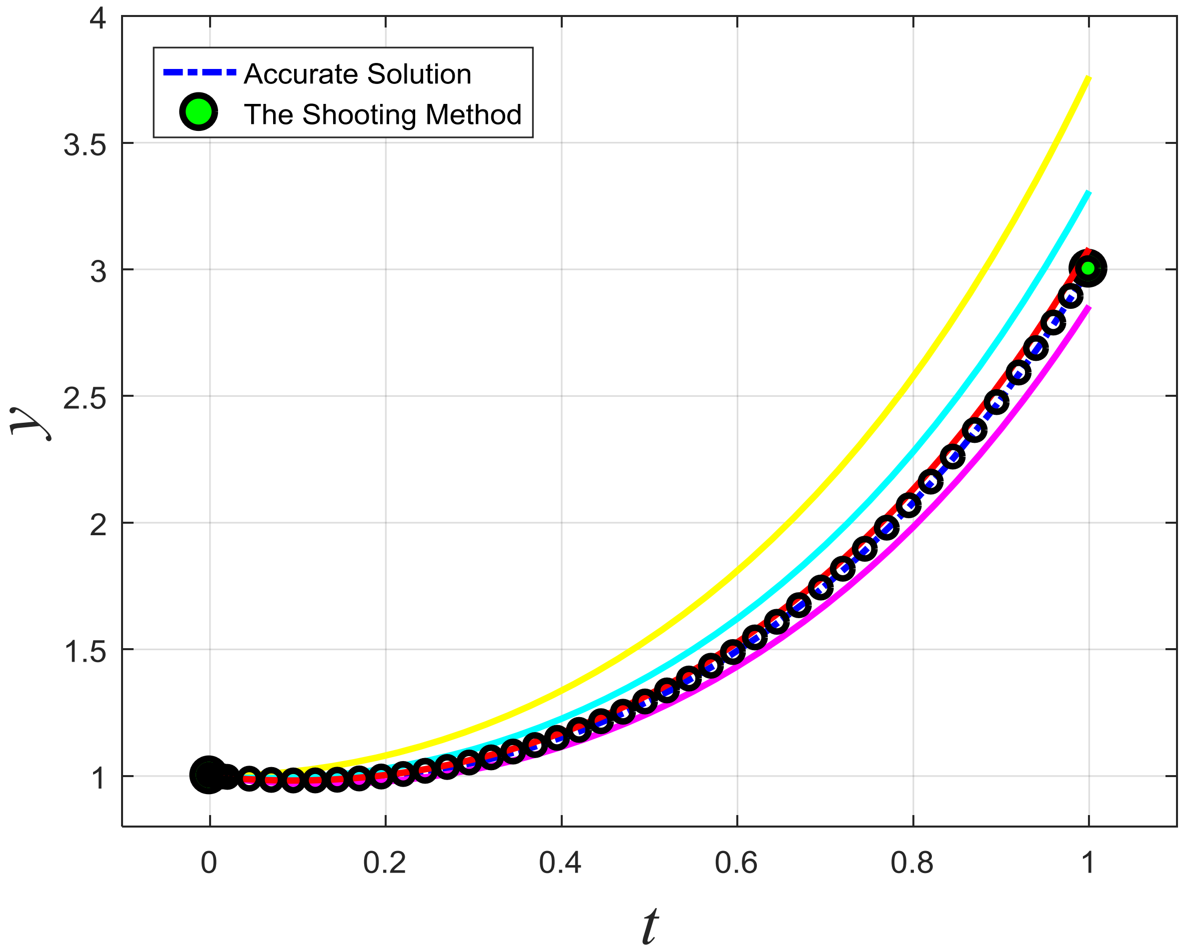
Step 3: …

Numerical Computation

The Shooting Method: Example 3

- **Apply the Shooting Method to the boundary value problem:**

$$\begin{cases} y'' = 4y \\ y(0) = 1 \\ y(1) = 3 \end{cases}$$



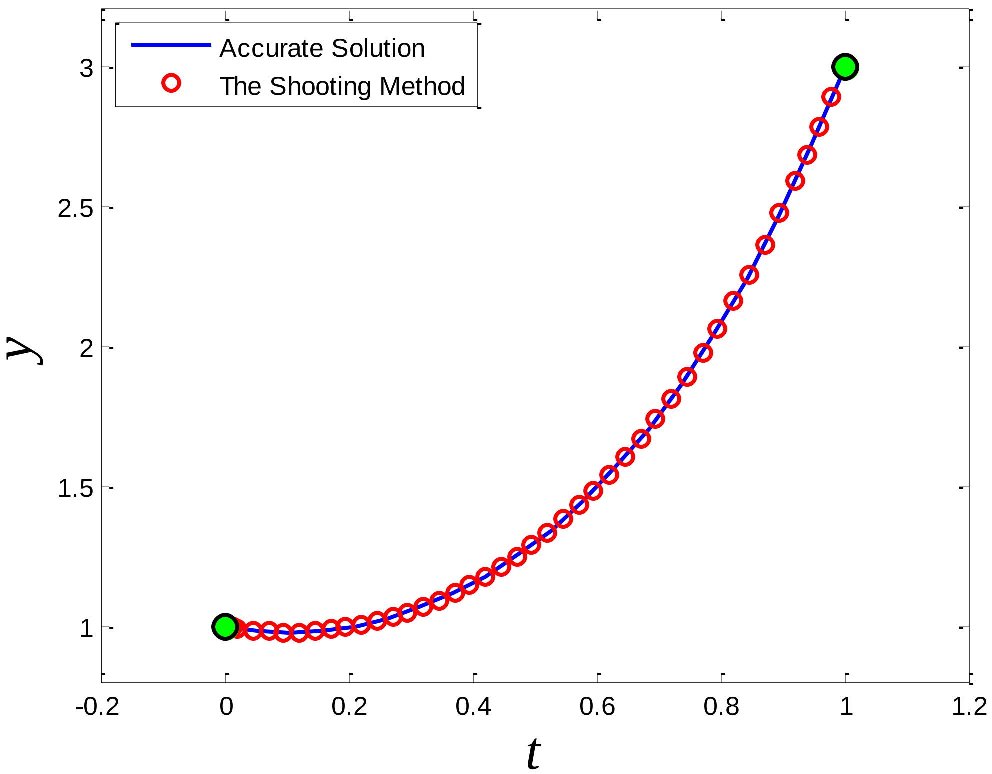
Numerical Computation

The Shooting Method: Example 3

- **Apply the Shooting Method to the boundary value problem:**

$$\begin{cases} y'' = 4y \\ y(0) = 1 \\ y(1) = 3 \end{cases}$$

$$s^* = y'(0) \approx -0.4203$$



Numerical Computation

- Shooting Method
- Finite Difference Methods
  - Linear boundary value problems
  - Nonlinear boundary value problems
- Finite Element Method

Numerical Computation

Finite Difference Methods: Basic Idea

A general second-order boundary value problem on a specific interval $a \le t \le b$:

$$\begin{cases} y'' = f(t, y, y') \\ y(a) = y_a \\ y(b) = y_b \end{cases}$$

How to replace derivatives in the differential equation by discrete approximations?

Numerical Computation

Finite Difference Formulas: Review

Three-point centered-difference formula

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6} f'''(c)$$

where $x - h < c < x + h$.

Three-point centered-difference formula for second derivative

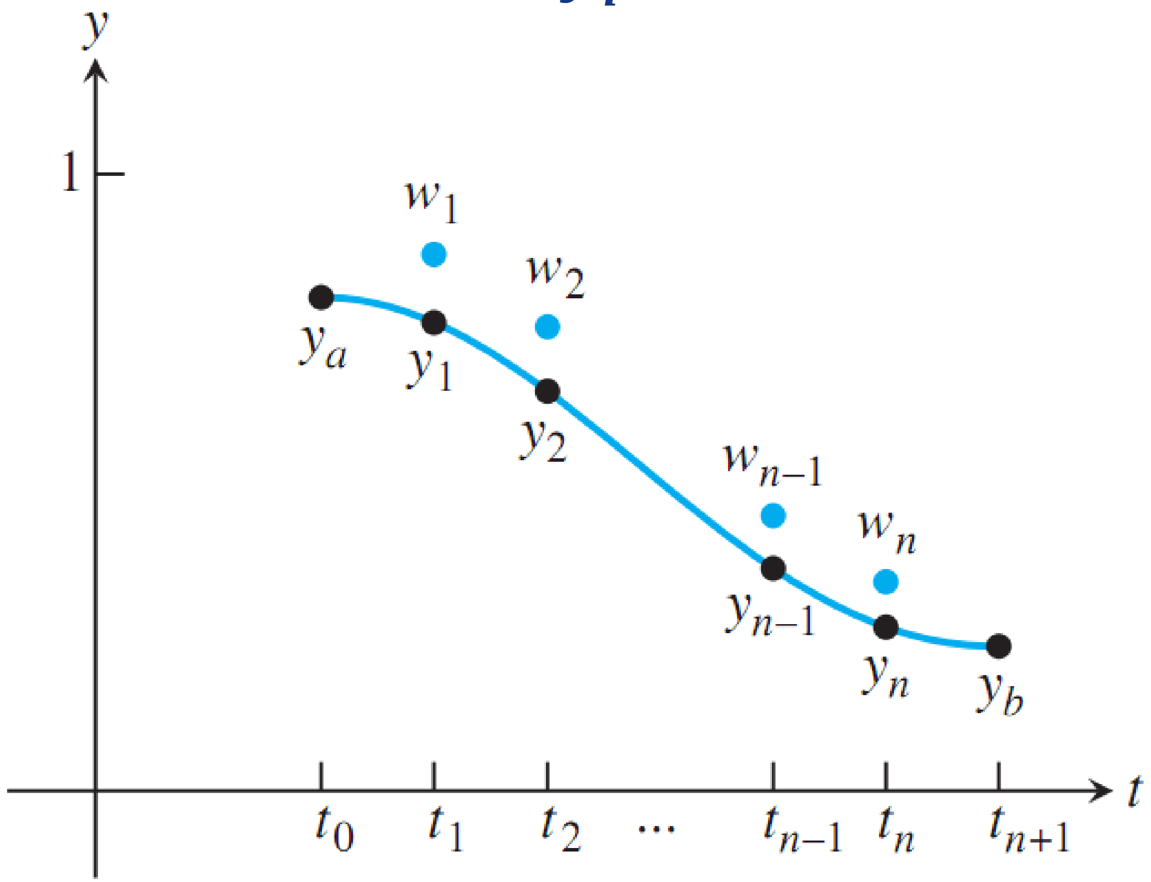
$$f''(x) = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} - \frac{h^2}{12} f^{(\mathrm{iv})}(c)$$

for some c between $x - h$ and $x + h$.

Numerical Computation

Finite Difference Methods: Basic Idea

Solving the algebraic equations for approximations w_i to the correct values y_i:



Numerical Computation

Finite Difference Methods: Basic Idea

$$\begin{cases} y'' = f(t, y, y') \\ y(a) = y_a \\ y(b) = y_b \end{cases}$$

Step 1: **dividing the interval [*a*,*b*] into *n*+1 equally spaced subintervals, i.e.,** $t_i = a + ih$, ***i*** **= 0, 1 ,…, *n*+1.**

Step 2: **replacing the derivatives with finite difference approximations.**

Step 3: **solving the system of algebraic equations.**

Numerical Computation

Finite Difference Methods: Implementation

$$\begin{cases} y'' = f(t, y, y') \\ y(a) = y_a \\ y(b) = y_b \end{cases}$$

$t_i = a + ih$, ***i*** **= 0, 1 ,…,** ***n*****+1.**

$$y'(t_i) \approx \frac{y_{i+1} - y_{i-1}}{2h} \quad \text{and} \quad y''(t_i) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$$

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

for ***i*** **= 1, …,** ***n*****.**

Numerical Computation

The Finite Difference Method: Example 4

- **Apply the Finite Difference Method to the boundary value problem:**

$$\begin{cases} y'' = 4y \\ y(0) = 1 \\ y(1) = 3 \end{cases}$$

Step 1: $n = 3$, $h = 1/4$; $t_i = a + ih$, $i = 0, 1, \ldots, 4$

Numerical Computation

- **The Finite Difference Method: Example 4**

Step 2:

$$y'' = 4y$$

$$\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} - 4w_i = 0$$

$$w_{i-1} + (-4h^2 - 2)w_i + w_{i+1} = 0$$

for i = 1 ,…, n . $w_0 = 1$ and $w_4 = 3$

Numerical Computation

The Finite Difference Method: Example 4

Step 3: algebraic equations

$$1 + (-4h^2 - 2)w_1 + w_2 = 0$$

$$w_1 + (-4h^2 - 2)w_2 + w_3 = 0$$

$$w_2 + (-4h^2 - 2)w_3 + 3 = 0.$$

***h* = 1/4**

$$\begin{bmatrix} -\frac{9}{4} & 1 & 0 \\ 1 & -\frac{9}{4} & 1 \\ 0 & 1 & -\frac{9}{4} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ -3 \end{bmatrix}$$

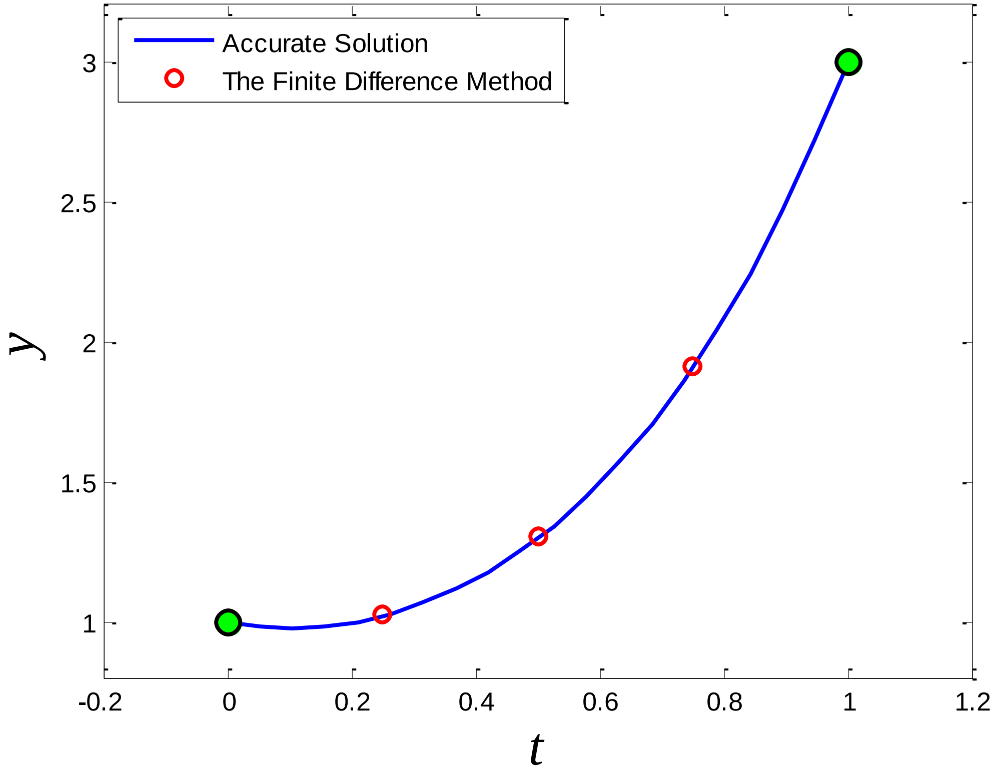
Numerical Computation

The Finite Difference Method: Example 4

Result:

$n = 3,\ h = 1/4$

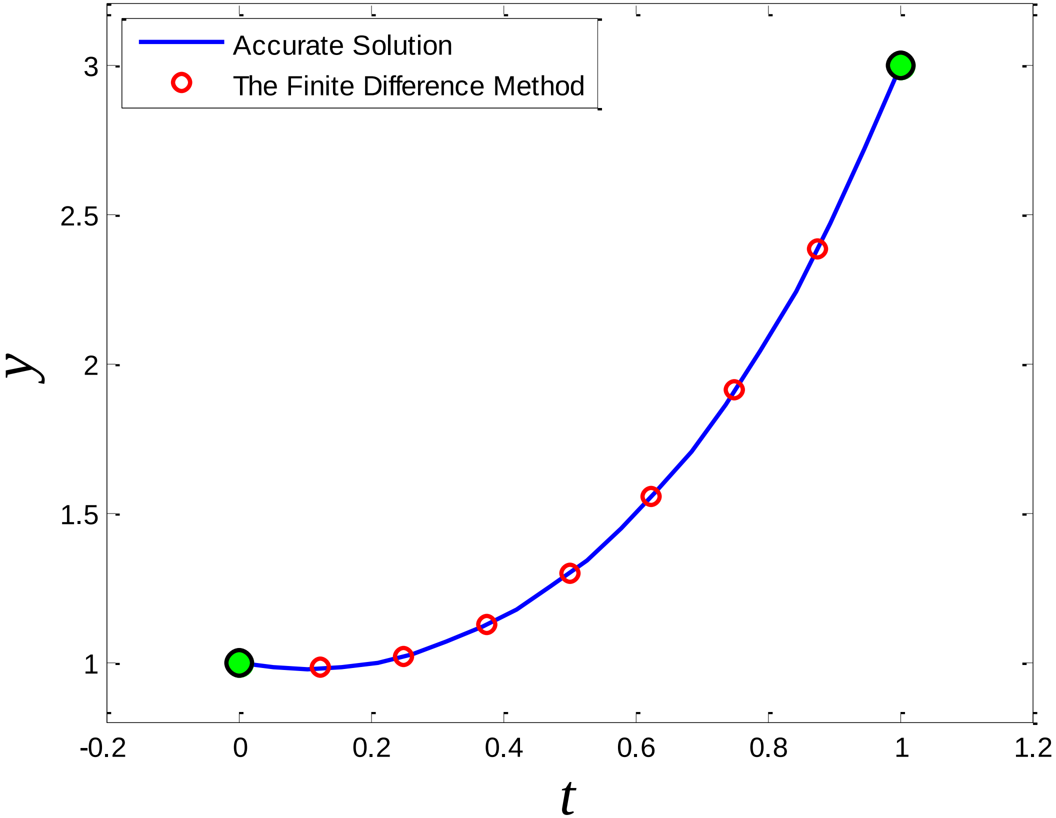
i	t_i	w_i	y_i
0	0.00	1.0000	1.0000
1	0.25	1.0249	1.0181
2	0.50	1.3061	1.2961
3	0.75	1.9138	1.9049
4	1.00	3.0000	3.0000



Numerical Computation

The Finite Difference Method: Example 4

Result: $n = 7, h = 1/8$



Numerical Computation

The Finite Difference Method: Example 4

Comparison: $h = 1/4$, $h = 1/8$, $h = 1/16$

t_i	$e_i\,(h = 1/4)$	$e_i\,(h = 1/8)$	$e_i\,(h = 1/16)$
0.25	0.00683	0.00174	1.0e-03 * 0.43737
0.50	0.01001	0.00255	1.0e-03 * 0.64156
0.75	0.00889	0.00227	1.0e-03 * 0.57051

Numerical Computation

- Shooting Method
- Finite Difference Methods
  - Linear boundary value problems
  - **Nonlinear boundary value problems**
- Finite Element Method

Numerical Computation

The Finite Difference Method: Example 5

- Apply the Finite Difference Method to the **nonlinear** boundary value problem:

$$\begin{cases} y'' = y - y^2 \\ y(0) = 1 \\ y(1) = 4 \end{cases}$$

Step 1: $n = 3$, $h = 1/4$; $t_i = a + ih$, $i = 0, 1, \ldots, 4$

Numerical Computation

The Finite Difference Method: Example 5

Step 2:

$$y'' = y - y^2$$

***for i* = 1 ,…, *n*.**

$$\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} - w_i + w_i^2 = 0$$

$$w_{i-1} - (2 + h^2)w_i + h^2 w_i^2 + w_{i+1} = 0,\ i = 1, \ldots, n.$$

with $w_0 = y_a = 1$ and $w_{n+1} = y_b = 4$

Numerical Computation

The Finite Difference Method: Example 5

Step 3: algebraic equations

$$y_a - (2 + h^2)w_1 + h^2 w_1^2 + w_2 = 0$$

$$w_{i-1} - (2 + h^2)w_i + h^2 w_i^2 + w_{i+1} = 0, \; i = 2, \ldots, n-1.$$

$$w_{n-1} - (2 + h^2)w_n + h^2 w_n^2 + y_b = 0$$

Numerical Computation

- **Newton's Method for Nonlinear Equations: Review**

Systems of Equations:

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}$$

Taylor Expansion:

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\mathbf{x}_i) + \mathbf{f}'(\mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i) = \mathbf{0}$$

$$\Downarrow$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \left[\mathbf{f}'(\mathbf{x}_i)\right]^{-1} \mathbf{f}(\mathbf{x}_i)$$

Multivariate Newton's Method

Numerical Computation

The Finite Difference Method: Example 5

Step 3: algebraic equations. The function *F*(w) is

$$F\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n-1} \\ w_n \end{bmatrix} = \begin{bmatrix} y_a - (2+h^2)w_1 + h^2 w_1^2 + w_2 \\ w_1 - (2+h^2)w_2 + h^2 w_2^2 + w_3 \\ \vdots \\ w_{n-2} - (2+h^2)w_{n-1} + h^2 w_{n-1}^2 + w_n \\ w_{n-1} - (2+h^2)w_n + h^2 w_n^2 + y_b \end{bmatrix}$$

The Jacobian *DF*(w) of *F* is

$$\begin{bmatrix} 2h^2 w_1 - (2+h^2) & 1 & 0 & \cdots & 0 \\ 1 & 2h^2 w_2 - (2+h^2) & \ddots & \ddots & \vdots \\ 0 & 1 & \ddots & 1 & 0 \\ \vdots & \ddots & \ddots & 2h^2 w_{n-1} - (2+h^2) & 1 \\ 0 & \cdots & 0 & 1 & 2h^2 w_n - (2+h^2) \end{bmatrix}$$

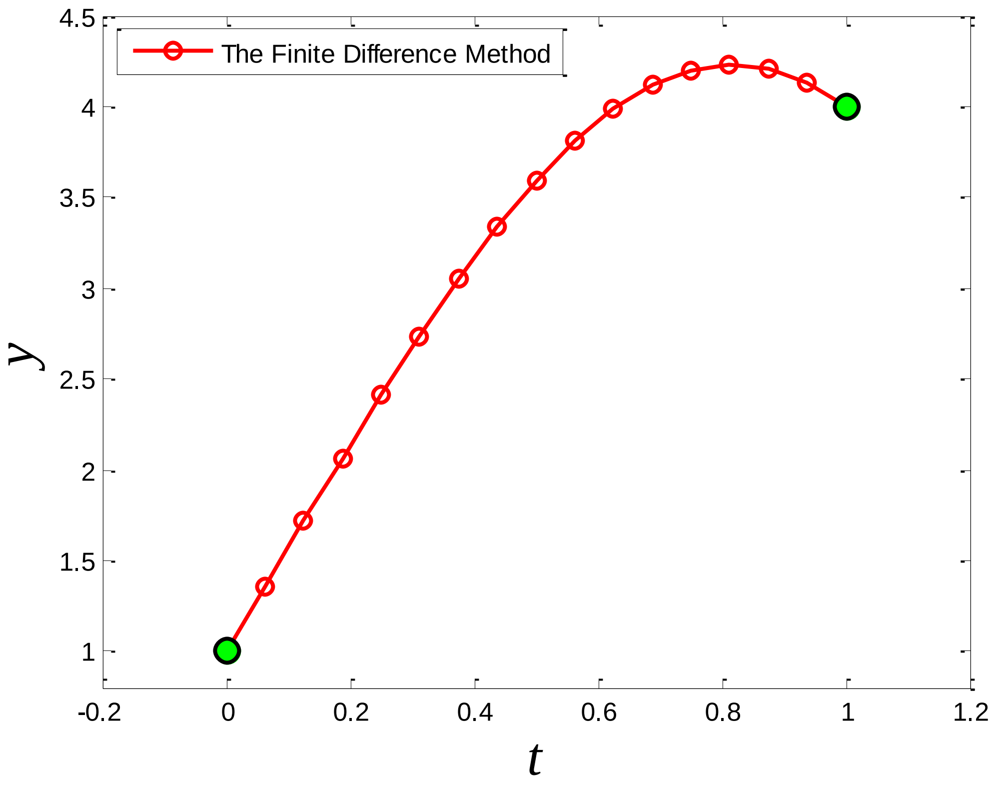
Numerical Computation

The Finite Difference Method: Example 5

Result:

$n = 15,\ h = 1/16$

$$\begin{cases} y'' = y - y^2 \\ y(0) = 1 \\ y(1) = 4 \end{cases}$$



Numerical Computation

- Shooting Method
- Finite Difference Methods
  - Linear boundary value problems
  - Nonlinear boundary value problems
- Finite Element Method

Numerical Computation

Finite Element Method: Basic Idea

The finite element approach to the BVP:

$$\begin{cases} y'' = f(t, y, y') \\ y(a) = y_a \\ y(b) = y_b. \end{cases}$$

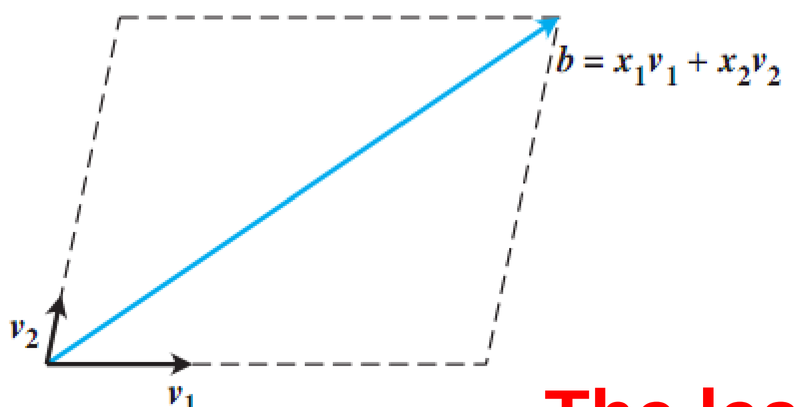
to choose the approximate solution *y* so that the residual $r = y'' - f$ is as small as possible.

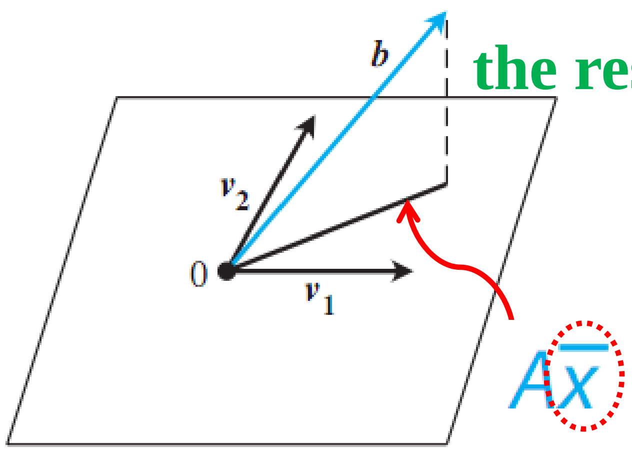
Linear Least Squares: Review

The Normal Equation: Basic Idea

$$\boxed{x_1 \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}_{\mathbf{v}_1} + x_2 \underbrace{\begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}}_{\mathbf{v}_2}} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} \triangleq \mathbf{b}$$

Combinations of two three-dimensional vectors v_1 and v_2





The least squares solution

Linear Least Squares: Review

- **The Normal Equation: Basic Idea**

Search for a formula for $\overline{x}$

$$(b - A\overline{x}) \perp \{Ax \mid x \in R^n\}$$

$$\Downarrow$$

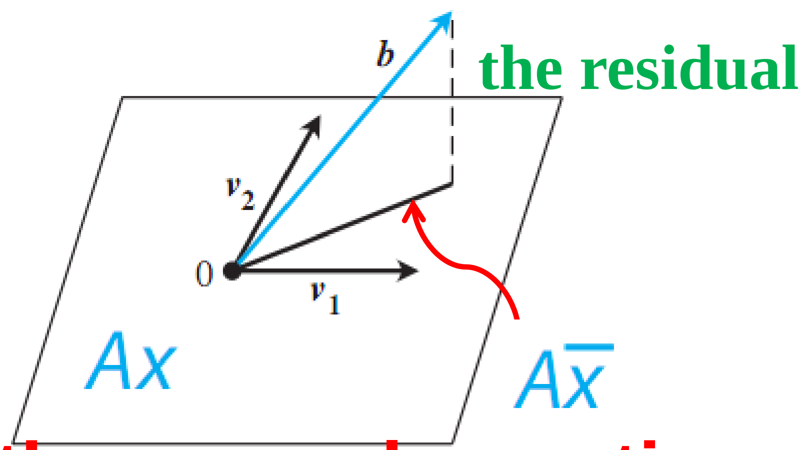
$$(Ax)^T(b - A\overline{x}) = 0 \text{ for all } x \text{ in } R^n$$

$$\Downarrow$$

$$A^T(b - A\overline{x}) = 0$$

$$\Downarrow$$

$$\boxed{A^T A\overline{x} = A^T b}$$



the normal equations

Numerical Computation

Finite Element Method: Basic Idea

In analogy with the least squares methods, this is accomplished by choosing *y* to make the residual orthogonal to the vector space of potential solutions.

(1) How to characterize "orthogonal" ?

(2) How to express "the vector space of potential solutions"?

Numerical Computation

- **Finite Element Method:**

(1) How to characterize “orthogonal” ?

For an interval [a,b], define the vector space of square integrable functions:

$$L^2[a,b] = \left\{ \text{functions } \; y(t) \; \text{ on } \; [a,b] \; \middle| \; \int_a^b y(t)^2 \, dt \text{ exists and is finite} \right\}$$

The L^2 function space has an inner product:

$$\langle y_1, y_2 \rangle = \int_a^b y_1(t) \, y_2(t) \, dt$$

Numerical Computation

- **Finite Element Method:**

(1) How to characterize "orthogonal" ?

The L^2 function space has an inner product:

$$\langle y_1, y_2 \rangle = \int_a^b y_1(t)\, y_2(t)\, dt$$

The usual properties:

1. $\langle y_1, y_1 \rangle \geq 0;$
2. $\langle \alpha y_1 + \beta y_2, z \rangle = \alpha \langle y_1, z \rangle + \beta \langle y_2, z \rangle$ for scalars α, β;
3. $\langle y_1, y_2 \rangle = \langle y_2, y_1 \rangle.$

Numerical Computation

- **Finite Element Method:**

(1) How to characterize "orthogonal" ?

Two functions $y_1(t)$ and $y_2(t)$ are orthogonal in $L^2[a, b]$, if

$$\langle y_1, y_2 \rangle = 0$$

Numerical Computation

Finite Element Method:

(2) How to express "the infinite-dimensional vector space of potential solutions"?

Choose a set of basis functions $\phi_0(t), \ldots, \phi_{n+1}(t)$ **, (finite-dimensional):**

$$y(t) = \sum_{i=0}^{n+1} c_i \phi_i(t)$$

where the basis functions may be polynomials, trigonometric functions, splines, or other simple functions.

Numerical Computation

- **Finite Element Method:**

(2) How to express "the vector space of potential solutions"?

With a grid $t_0 < t_1 < \cdots < t_n < t_{n+1}$

for *i* = 1,...,*n*, define

$$\phi_i(t) = \begin{cases} \dfrac{t - t_{i-1}}{t_i - t_{i-1}} & \text{for } \; t_{i-1} < t \le t_i \\ \dfrac{t_{i+1} - t}{t_{i+1} - t_i} & \text{for } \; t_i < t < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

Numerical Computation

Finite Element Method:

(2) How to express "the vector space of potential solutions"?

for i = 0:

$$\phi_0(t) = \begin{cases} \dfrac{t_1 - t}{t_1 - t_0} & \text{for} \quad t_0 \le t < t_1 \\ 0 & \text{otherwise} \end{cases}$$

for i = n+1:

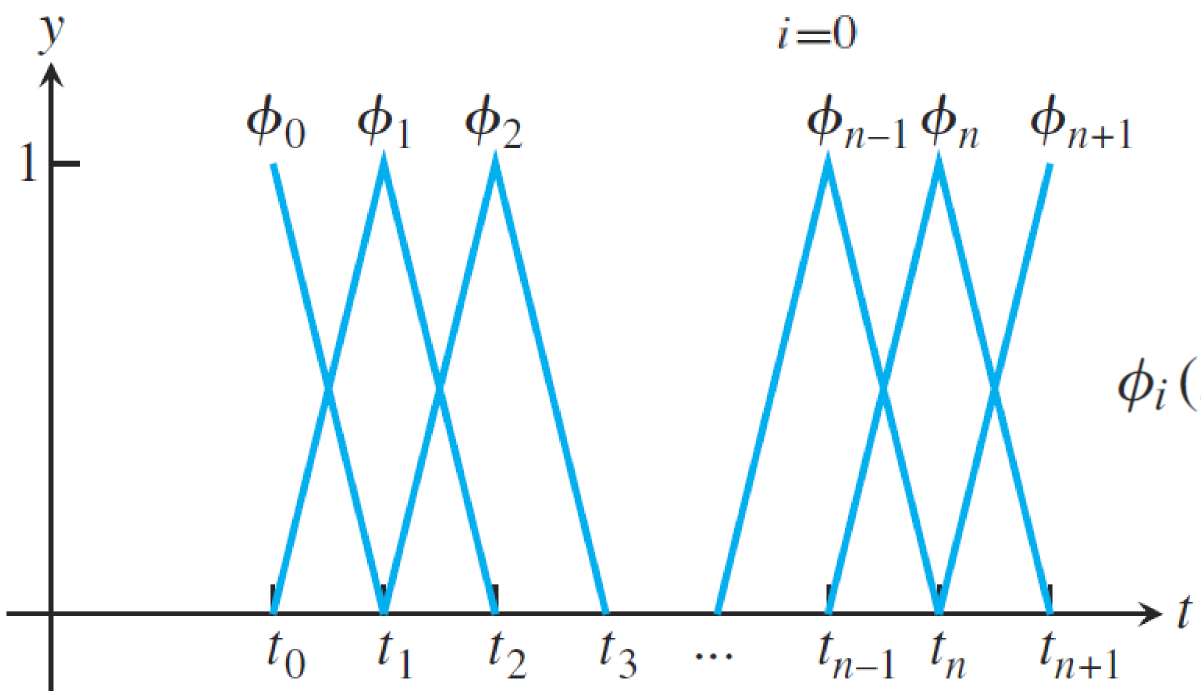
$$\phi_{n+1}(t) = \begin{cases} \dfrac{t - t_n}{t_{n+1} - t_n} & \text{for} \quad t_n < t \le t_{n+1} \\ 0 & \text{otherwise} \end{cases}$$

Numerical Computation

- **Finite Element Method:**

For a set of data points (t_i, c_i), the piecewise-linear B-splines:

$$S(t) = \sum_{i=0}^{n+1} c_i \phi_i(t)$$



$$\phi_i(t_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Numerical Computation

Finite Element Method: Implementation

The finite element approach to the BVP

$$\begin{cases} y'' = f(t, y, y') \\ y(a) = y_a \\ y(b) = y_b. \end{cases}$$

the residual $r = y'' - f$

To minimize *r* by forcing it to be orthogonal to the basis functions

$$\int_a^b (y'' - f)\phi_i \, dt = 0$$

for each $0 \le i \le n + 1$.

Numerical Computation

Finite Element Method: Implementation

$$\int_a^b (y'' - f)\phi_i \, dt = 0 \quad \text{for each } 0 \le i \le n+1.$$

$$\int_a^b y''(t)\phi_i(t)\, dt = \int_a^b f(t, y, y')\phi_i(t)\, dt$$

$$\int_a^b y''(t)\phi_i(t)\, dt = \phi_i(t) y'(t)\Big|_a^b - \int_a^b y'(t)\phi_i'(t)\, dt$$

$$= \phi_i(b) y'(b) - \phi_i(a) y'(a) - \int_a^b y'(t)\phi_i'(t)\, dt$$

Numerical Computation

Finite Element Method: Implementation

$$\int_a^b (y'' - f)\phi_i \, dt = 0$$ **for each $0 \le i \le n+1$.**

$$\int_a^b f(t, y, y')\phi_i(t)\, dt = \phi_i(b)y'(b) - \phi_i(a)y'(a) - \int_a^b y'(t)\phi_i'(t)\, dt$$

the piecewise-linear B-splines $y(t) = \sum_{i=0}^{n+1} c_i \phi_i(t)$

$$\int_a^b f(t, y, y')\phi_i(t)\, dt + \int_a^b y'(t)\phi_i'(t)\, dt = 0$$ ***for i* = 1,…,*n***

Numerical Computation

Finite Element Method: Implementation

$$y(a) = \sum_{i=0}^{n+1} c_i \phi_i(a) = c_0 \phi_0(a) = c_0$$

$$y(b) = \sum_{i=0}^{n+1} c_i \phi_i(b) = c_{n+1} \phi_{n+1}(b) = c_{n+1}$$

$$\int_a^b f(t, y, y') \phi_i(t)\, dt + \int_a^b y'(t) \phi_i'(t)\, dt = 0$$ ***for i* = 1,…,*n***

$$\int_a^b \phi_i(t) f\left(t, \sum c_j \phi_j(t), \sum c_j \phi_j'(t)\right) dt + \int_a^b \phi_i'(t) \sum c_j \phi_j'(t)\, dt = 0$$

Numerical Computation

Finite Element Method: Implementation

Assume that the grid is evenly spaced with step size *h*.

$$\int_a^b \phi_i(t)\phi_{i+1}(t)\,dt = \int_0^h \frac{t}{h}\left(1-\frac{t}{h}\right)dt = \int_0^h \left(\frac{t}{h}-\frac{t^2}{h^2}\right)dt$$

$$= \frac{t^2}{2h} - \frac{t^3}{3h^2}\Big|_0^h = \frac{h}{6}$$

$$\int_a^b (\phi_i(t))^2\,dt = 2\int_0^h \left(\frac{t}{h}\right)^2 dt = \frac{2}{3}h$$

$$\int_a^b \phi_i'(t)\phi_{i+1}'(t)\,dt = \int_0^h \frac{1}{h}\left(-\frac{1}{h}\right)dt = -\frac{1}{h}$$

$$\int_a^b (\phi_i'(t))^2\,dt = 2\int_0^h \left(\frac{1}{h}\right)^2 dt = \frac{2}{h}$$

Numerical Computation

Finite Element Method: Example 6

- **Apply the Finite Element Method to the boundary value problem:**

$$\begin{cases} y'' = 4y \\ y(0) = 1 \\ y(1) = 3 \end{cases}$$

$$0 = \int_0^1 \left(4\phi_i(t) \sum_{j=0}^{n+1} c_j \phi_j(t) + \sum_{j=0}^{n+1} c_j \phi_j'(t) \phi_i'(t) \right) dt$$

$$= \sum_{j=0}^{n+1} c_j \left[4 \int_0^1 \phi_i(t) \phi_j(t)\, dt + \int_0^1 \phi_j'(t) \phi_i'(t)\, dt \right]$$

Numerical Computation

Finite Element Method: Example 6

- The first and last of the c_i are found by:

$$c_0 = f(a) \qquad c_{n+1} = f(b)$$

- for $i = 1,\ldots,n$:

$$\left[\frac{2}{3}h - \frac{1}{h}\right]c_0 + \left[\frac{8}{3}h + \frac{2}{h}\right]c_1 + \left[\frac{2}{3}h - \frac{1}{h}\right]c_2 = 0$$

$$\left[\frac{2}{3}h - \frac{1}{h}\right]c_1 + \left[\frac{8}{3}h + \frac{2}{h}\right]c_2 + \left[\frac{2}{3}h - \frac{1}{h}\right]c_3 = 0$$

$$\vdots$$

$$\left[\frac{2}{3}h - \frac{1}{h}\right]c_{n-1} + \left[\frac{8}{3}h + \frac{2}{h}\right]c_n + \left[\frac{2}{3}h - \frac{1}{h}\right]c_{n+1} = 0$$

Numerical Computation

Finite Element Method: Example 6

- The matrix form of the equations is

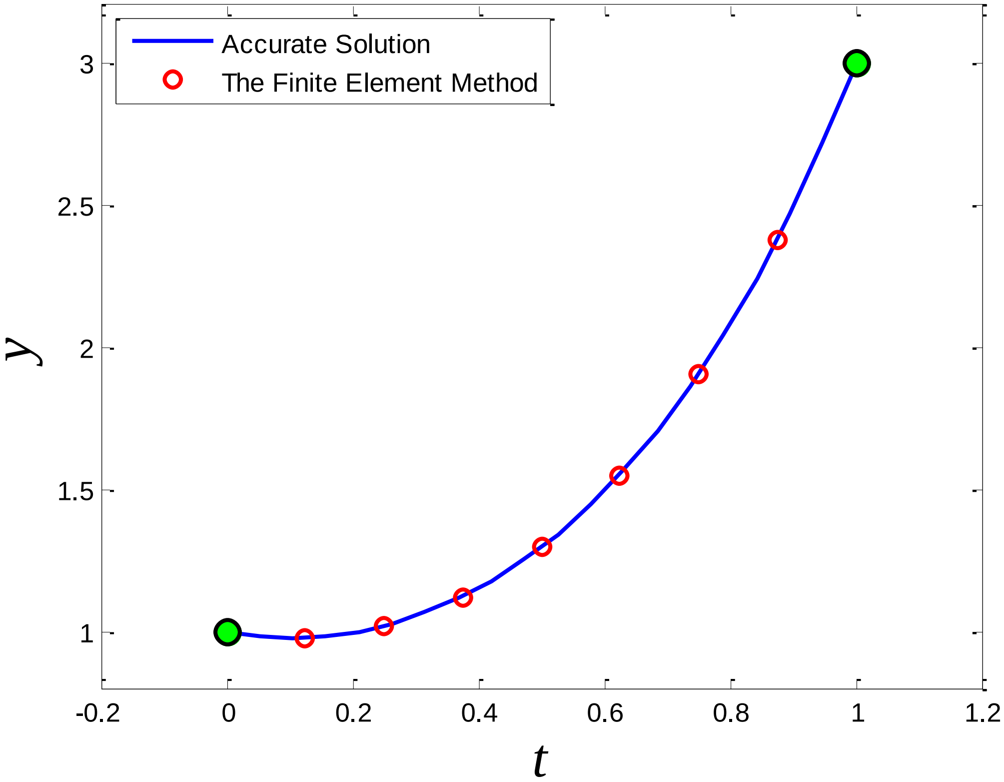
$$\begin{bmatrix} \alpha & \beta & 0 & \cdots & 0 \\ \beta & \alpha & \ddots & \ddots & \vdots \\ 0 & \beta & \ddots & \beta & 0 \\ \vdots & \ddots & \ddots & \alpha & \beta \\ 0 & \cdots & 0 & \beta & \alpha \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix} = \begin{bmatrix} -y_a\beta \\ 0 \\ \vdots \\ 0 \\ -y_b\beta \end{bmatrix}$$

where $\alpha = \frac{8}{3}h + \frac{2}{h}$ and $\beta = \frac{2}{3}h - \frac{1}{h}$

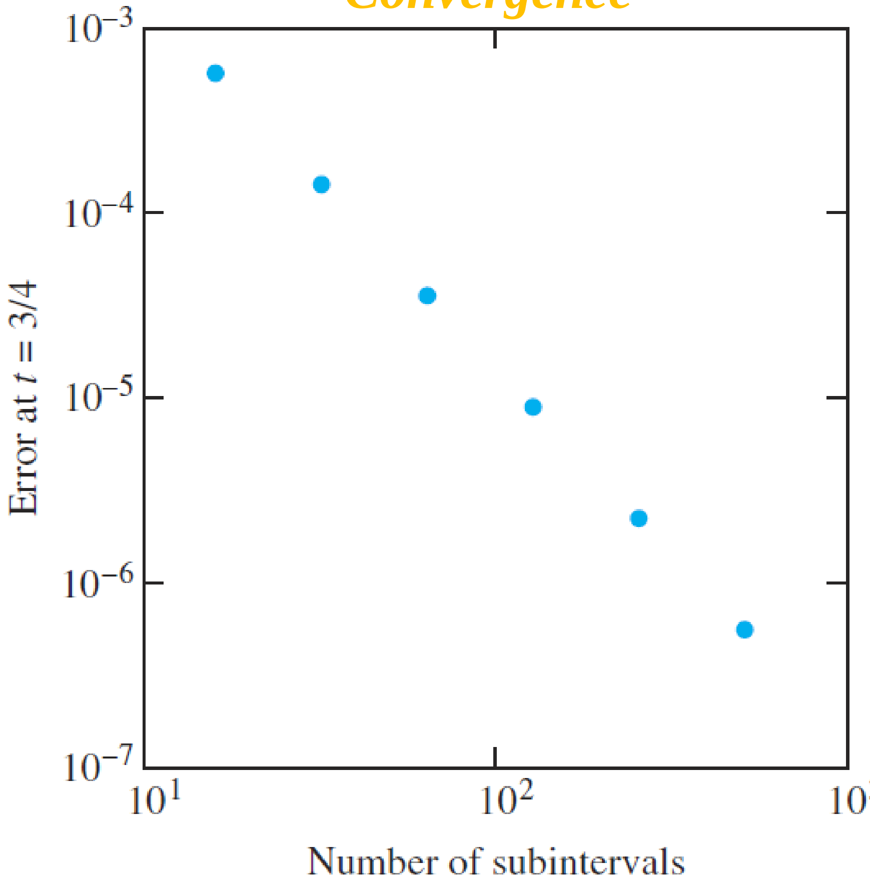
Numerical Computation

Finite Element Method: Example 6

- **Result with $n = 7, h = 1/8$**



Convergence



MATLAB Built-in Functions

MATLAB Built-in Functions for ODEs

bvp4c	Solve boundary value problems for ordinary differential equations.
bvp5c	Solve boundary value problems for ordinary differential equations.
bvpinit	Form initial guess for BVP solvers.

Summary

- Symbolic Computation
- Shooting Method
- Finite Difference Methods
  - Linear boundary value problems
  - Nonlinear boundary value problems
- Finite Element Method

Thank You !

Second-order Boundary Value Problem

Theorem 11.1. cf. Ref. [2], P. 672

Suppose the function f in the boundary-value problem

$$y'' = f(x, y, y'), \quad \text{for } a \le x \le b, \text{ with } y(a) = \alpha \text{ and } y(b) = \beta,$$

is continuous on the set

$$D = \{ (x, y, y') \mid \text{ for } a \le x \le b, \text{ with } -\infty < y < \infty \text{ and } -\infty < y' < \infty \},$$

and that the partial derivatives f_y and $f_{y'}$ are also continuous on D. If

(i) $f_y(x, y, y') > 0$, for all $(x, y, y') \in D$, and

(ii) a constant M exists, with

$$\left| f_{y'}(x, y, y') \right| \le M, \quad \text{for all } (x, y, y') \in D,$$

then the boundary-value problem has a unique solution.