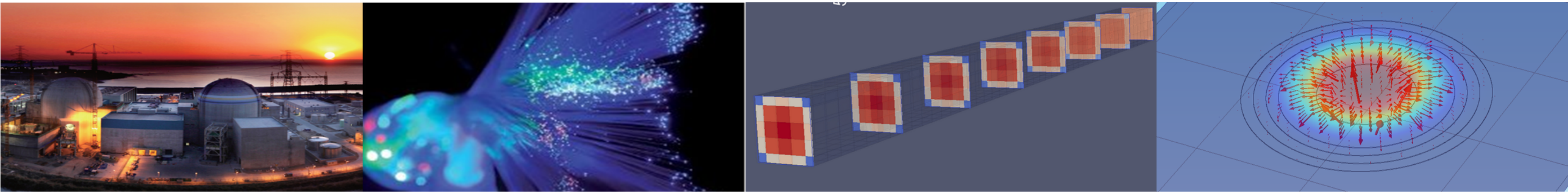


INTRODUCTION TO NUMERICAL ANALYSIS

Cho, Hyoung Kyu

Department of Nuclear Engineering

Seoul National University

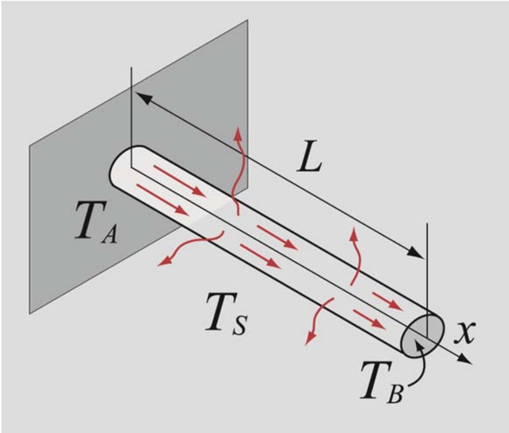
11. ORDINARY DIFFERENTIAL EQUATIONS: BOUNDARY-VALUE PROBLEMS

- Initial value problem vs. boundary value problem
  - A first-order ODE can be solved if one constraint, the value of the dependent variable (initial value) at one point is known.
  - To solve an nth-order equation, n constraints must be known.
    - The constraints can be the value of the dependent variable (solution) and its derivative(s) at certain values of the independent variable.
  - Initial value problem
    - When all the constraints are specified at one value of the independent variable
  - Boundary value problem
    - To solve differential equations of second and higher order that have constraints specified at different values of the independent variable
  - Boundary conditions
    - Because the constraints are often specified at the endpoints or boundaries of the domain of the solution.

- Example of BVP
  - Modeling of temperature distribution in a pin fin used as a heat sink for cooling an object

$$\frac{d^2T}{dx^2} - \alpha_1(T - T_S) - \alpha_2(T^4 - T_S^4) = 0$$

- T_S : temperature of the surrounding air
- a_1 and a_2: coefficients
- Boundary conditions: T_A and T_B



- Problem statement of a second-order boundary value problem

$$\frac{d^2y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right)$$

Possible to have nonlinear boundary conditions !

- Domain: $a \leq x \leq b$
- Dirichlet boundary conditions
- Neumann boundary conditions
- Mixed boundary conditions

$$y(a) = Y_a \quad \text{and} \quad y(b) = Y_b$$

$$\left.\frac{dy}{dx}\right|_{x=a} = D_a \quad \text{and} \quad \left.\frac{dy}{dx}\right|_{x=b} = D_b$$

$$c_1\left.\frac{dy}{dx}\right|_{x=a} + c_2 y(a) = C_a \qquad c_3\left.\frac{dy}{dx}\right|_{x=b} + c_4 y(b) = C_b$$

- BVP of higher order ODEs
  - Require additional boundary conditions
    - Typically the values of higher derivatives of y
  - For example,
    - The differential equation that relates the deflection of a beam, y, due to the application of a distributed load, $p(x)$, is:

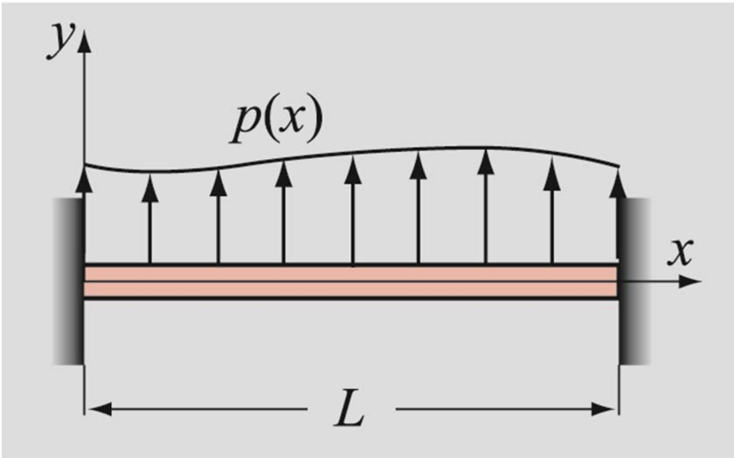
$$\frac{d^4y}{dx^4} = \frac{1}{EI}p(x)$$

E: elastic modulus of the beam's material
I: area moment of inertia of the beam's cross-sectional area

    - Four boundary conditions are necessary.

$$y(0) = 0; \qquad \left.\frac{dy}{dx}\right|_{x=0} = 0$$

$$y(L) = 0; \qquad \left.\frac{dy}{dx}\right|_{x=L} = 0$$



- Overview of numerical methods used for solving boundary value problems
  - Shooting methods
    - Reduce the second-order (or higher order) ODE to an initial value problem.
      - By transforming the equation into a system of 1^{st} order ODEs.
    - The boundary value at the first point of the domain
      - Used as an initial value for the system
    - The additional initial values needed for the solving the system
      - Guessed!
    - The solution at the end of the interval is compared with the specified boundary conditions.
    - Iteration!
  - Finite difference methods
    - The derivative in the differential equation are approximated with FD formula.
    - ODE $\Rightarrow$ a system of linear (or non-linear) algebraic equations
    - Advantages and disadvantages
      - No need to solve the differential equations several times
      - The solution of non-linear ODEs $\Rightarrow$ need to solve a system of non-linear equations iteratively
      - Shooting methods have advantages that the solution of non-linear ODEs is fairly straightforwad.

- Shooting method
  - Boundary value problem $\Rightarrow$ initial value problem
    - For example, a BVP involving an ODE of second-order
    - A system of two first-order ODEs
  - Solution procedure
    - BCs given at the first point: initial conditions for the system
    - Additional initial conditions ? Guessed !
    - The system can be solved!
    - Solution can be obtained at the end point of the domain.
    - Compare them with the BCs at the end point of the domain.
    - Check the error
    - The guessed initial values are changed then the system is solved again.
    - Repeated until the numerical solution agrees with the BCs.

- Shooting method for a two-point BVP
  - BVP with a Second-order ODE

$$\frac{d^2y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right) \text{ for } a \le x \le b \text{ with } y(a) = Y_a \text{ and } y(b) = Y_b$$

  - Step 1

$$\frac{dy}{dx} = w \quad \text{with the initial condition: } y(a) = Y_a$$

$$\frac{dw}{dx} = f(x, y, w)$$

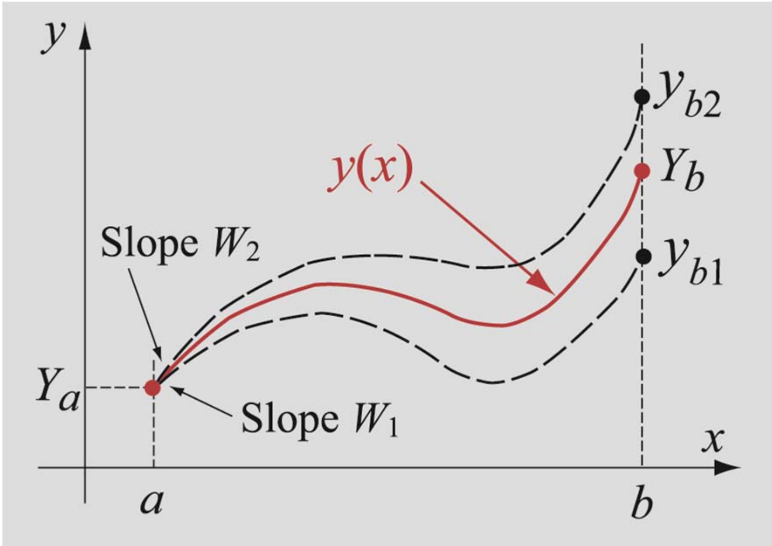
No initial condition

  - Step 2: first guess for the initial value

$$w(a) = \left.\frac{dy}{dx}\right|_{x=a} = W_1$$

  - Step 3: Second guess

$$w(a) = \left.\frac{dy}{dx}\right|_{x=a} = W_2$$



- Shooting method for a two-point BVP
  - Step 4: New estimate

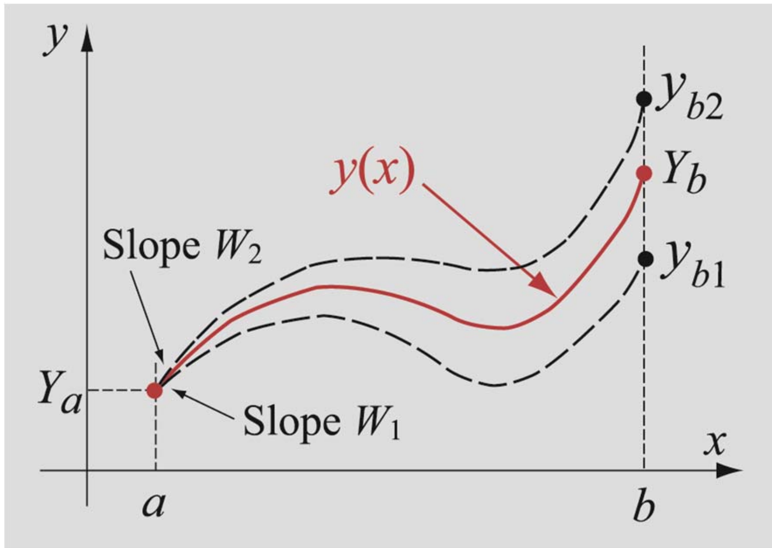
$$w(a) = \left.\frac{dy}{dx}\right|_{x=a} = W_3$$

    - Using the results of the previous two solutions
    - For example,

$$y_{b1} < Y_b < y_{b2}$$

      - Interpolation between W_1 and W_2

  - Repeat



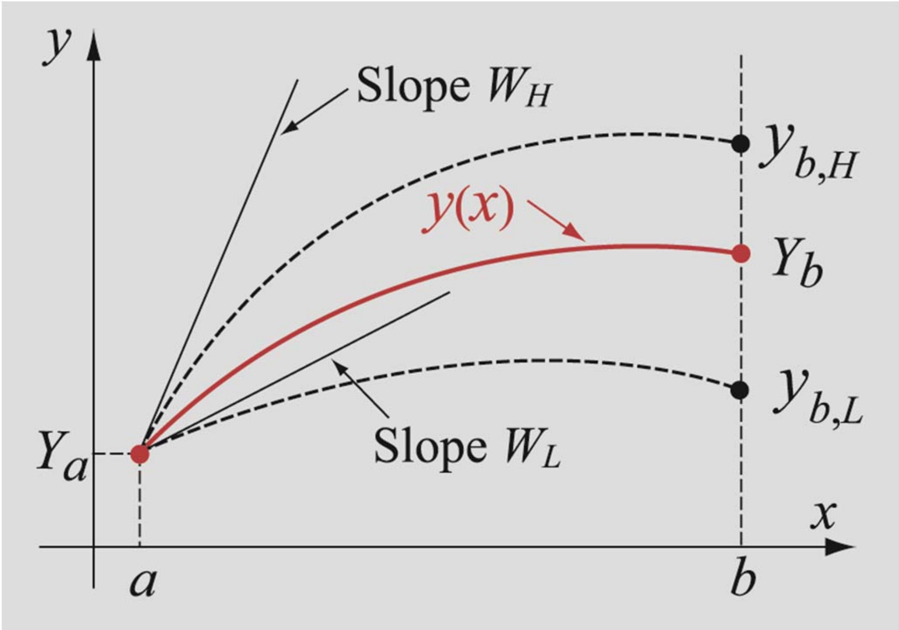
- Estimating the slope at $x = a$
  - Starts by guessing two values for the slope of $y(x)$ at the first point of the domain
  - Two solutions with two guesses
  - Estimate new value

  - Linear interpolation

$$W_N = W_L + (Y_b - y_{b,L})\frac{W_H - W_L}{y_{b,H} - y_{b,L}}$$

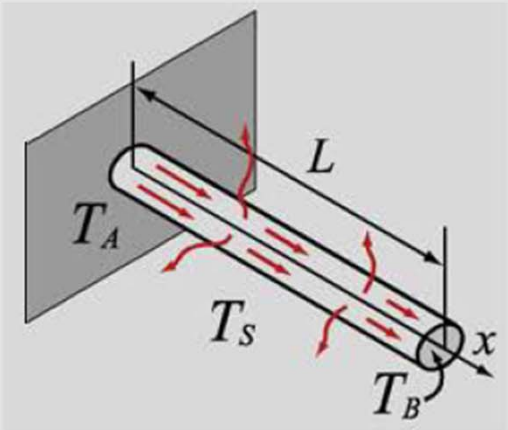
  - Bisection method

  - Secant method



❖ Example 11-1: Temperature distribution in a pin fin. Solving a second-order ODE (BVP) using the shooting method.

A pin fin is a slender extension attached to a surface in order to increase the surface area and enable greater heat transfer. When convection and radiation are included in the analysis, the steady-state temperature distribution, $T(x)$, along a pin fin can be calculated from the solution of the equation:



$$\frac{d^2T}{dx^2} - \frac{h_cP}{kA_c}(T - T_S) - \frac{\varepsilon\sigma_{SB}P}{kA_c}(T^4 - T_S^4) = 0 \ , \ 0 \le x \le L \qquad (11.15)$$

with the boundary conditions: $T(0) = T_A$ and $T(L) = T_B$.

In Eq. (11.15), h_c is the convective heat transfer coefficient, P is the perimeter bounding the cross section of the fin, ε is the radiative emissivity of the surface of the fin, k is the thermal conductivity of the fin material, A_c is the cross-sectional area of the fin, T_S is the temperature of the surrounding air, and $\sigma_{SB} = 5.67 \times 10^{-8}$ W/(m^{2}K^4) is the Stefan–Boltzmann constant.

Determine the temperature distribution if $L = 0.1$m, $T(0) = 473$K, $T(0.1) = 293$K, and $T_S = 293$K. Use the following values for the parameters in Eq. (11.15): $h_c = 40$ W/m^2/K, $P = 0.016$ m, $\varepsilon = 0.4$, $k = 240$ W/m/K, and $A_c = 1.6 \times 10^{-5}$m^2.

- Example 11-1: Temperature distribution in a pin fin. Solving a second-order ODE (BVP) using the shooting method.

$$\frac{dT}{dx} = w \qquad \frac{dw}{dx} = \frac{h_c P}{k A_c}(T - T_S) + \frac{\varepsilon \sigma_{SB} P}{k A_c}(T^4 - T_S^4)$$

- $T(0) = 473$
- Assume $w(0) = -1000$
- Assume $w(0) = -3000$

- Use Sys2ODEsRK2

- Example 11-1: Temperature distribution in a pin fin. Solving a second-order ODE (BVP) using the shooting method.

```
% Solving Example 11-1
clear all
a=0; b=0.1; TINI=473; wINI1=-1000; h=0.001; Tend=293;

wINI1=(TINI-Tend)/(a-b);
[x, T1, w] = ...
            Sys2ODEsRK2(@odeChap11Exmp1dTdx,@odeChap11Exmp1dwdx,a,b,h,TINI,wINI1);
n=length(x) ;
fprintf('The temperature at x=0.1 is %5.3f, for initial value of dt/dx= %4.1f\n',...
            T1(n) ,wINI1)

wINI2=0.5*wINI1;
[x, T2, w] = ...
            Sys2ODEsRK2(@odeChap11Exmp1dTdx,@odeChap11Exmp1dwdx,a,b,h,TINI,wINI2);
fprintf('The temperature at x=0.1 is %5.3f, for initial value of dt/dx= %4.1f\n',...
            T2(n) ,wINI2)
plot (x, T1,x,T2);
hold on;
```

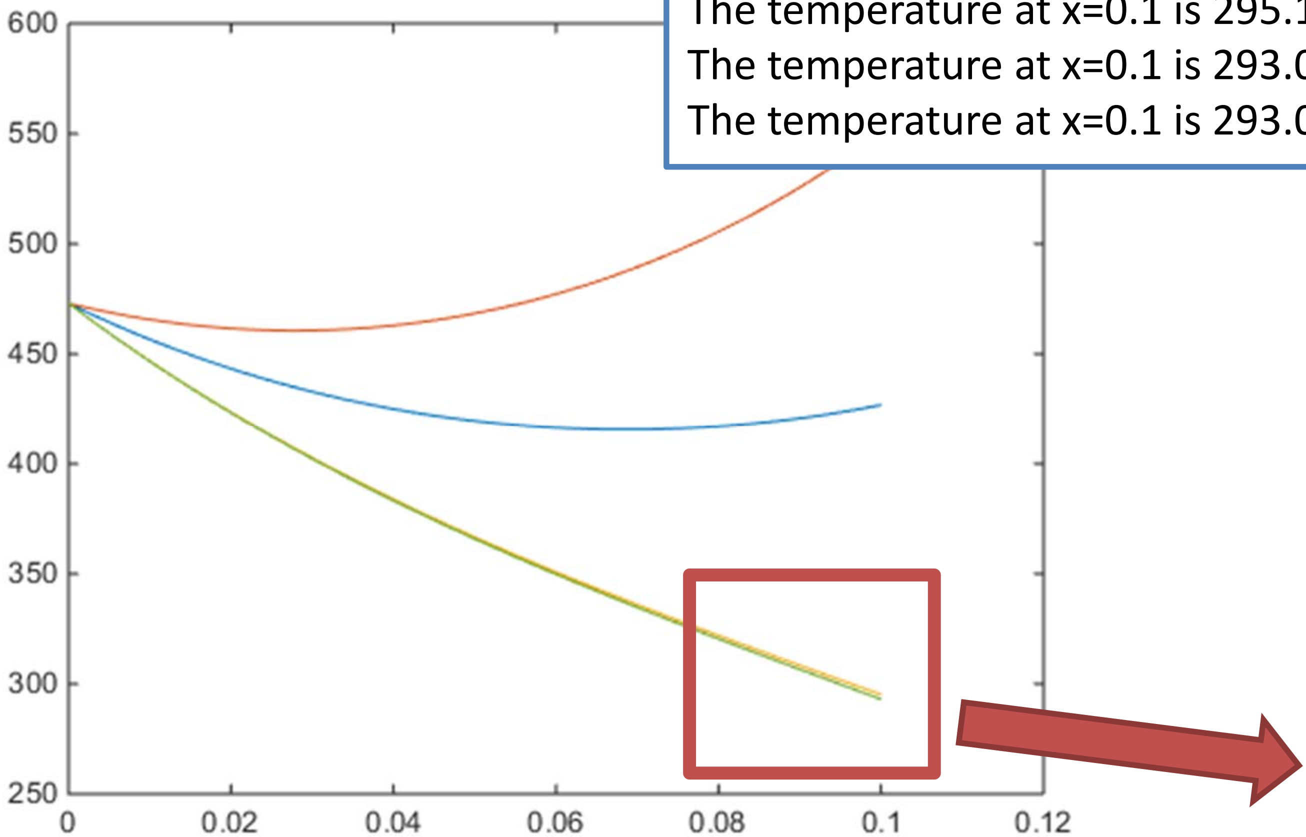
❖ Example 11-1: Temperature distribution in a pin fin. Solving a second-order ODE (BVP) using the shooting method.

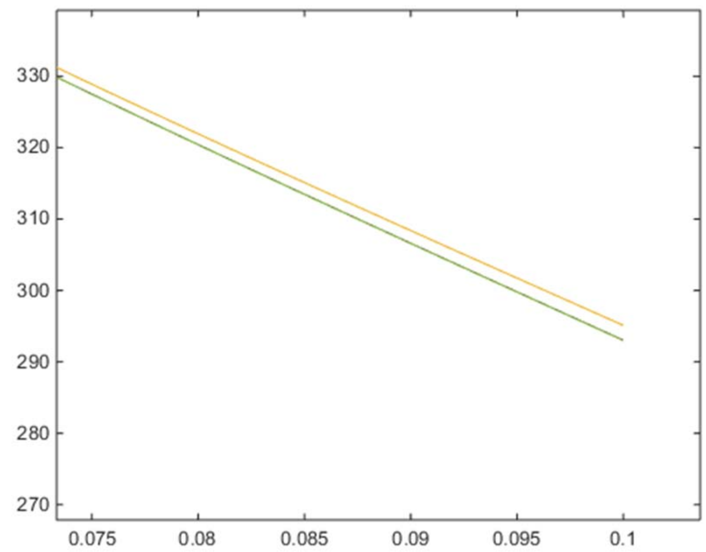
```
Told=T1(n);
Tnew=T2(n);
error=abs(Tend-Tnew);

while (error>1.0e-5)
    wINI3 = wINI1 + (Tend - Told) * (wINI2 - wINI1) / (Tnew - Told);
    [x, T3, w] =...
                Sys2ODEsRK2(@odeChap11Exmp1dTdx,@odeChap11Exmp1dwdx,a,b,h,TINI,wINI3);
    fprintf('The temperature at x=0.1 is %5.3f, for initial value ofdt/dx= %4.1f\n', ...
            T3(n) ,wINI3)
    Told=Tnew;
    Tnew=T3(n);
    wINI1=wINI2;
    wINI2=wINI3;
    error=abs(Tend-Tnew);
    plot (x, T3);
    hold on;
end
```

11.2 The Shooting Method

- Example 11-1: Temperature distribution in a pin fin. Solving a second-order ODE (BVP) using the shooting method.

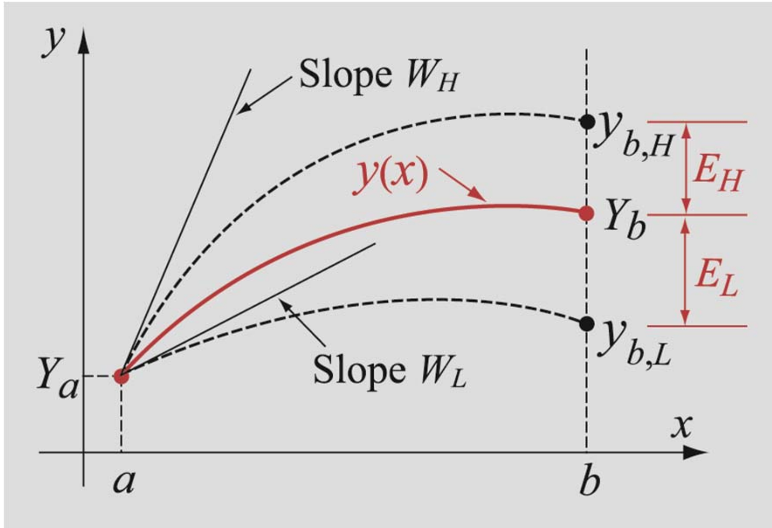
The temperature at x=0.1 is 426.894, for initial value of dt/dx= -1800.0
The temperature at x=0.1 is 550.330, for initial value of dt/dx= -900.0
The temperature at x=0.1 is 295.111, for initial value of dt/dx= -2776.2
The temperature at x=0.1 is 293.030, for initial value of dt/dx= -2791.8
The temperature at x=0.1 is 293.000, for initial value of dt/dx= -2792.0

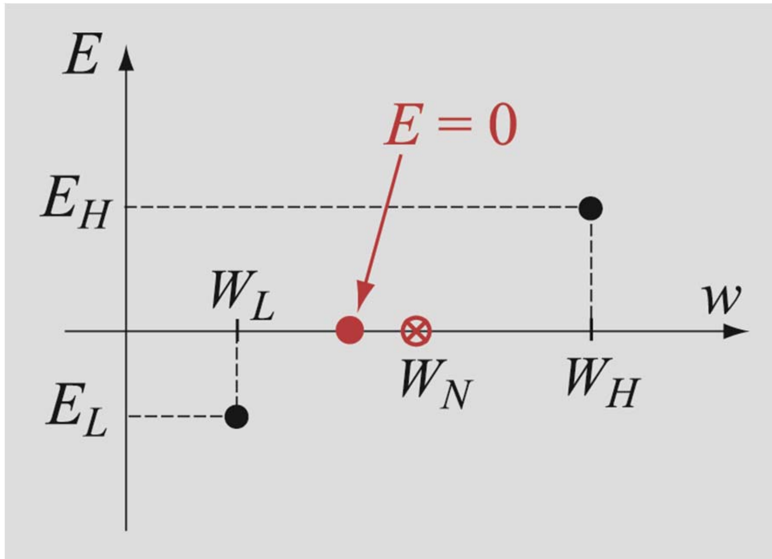




- Shooting method using the bisection method
  - With W_H $\quad y_{b,H} > Y_b$
  - With W_L $\quad y_{b,L} < Y_b$
  - At new iteration

$$W_N = \frac{1}{2}(W_H + W_L)$$





❖ Shooting method using the bisection method

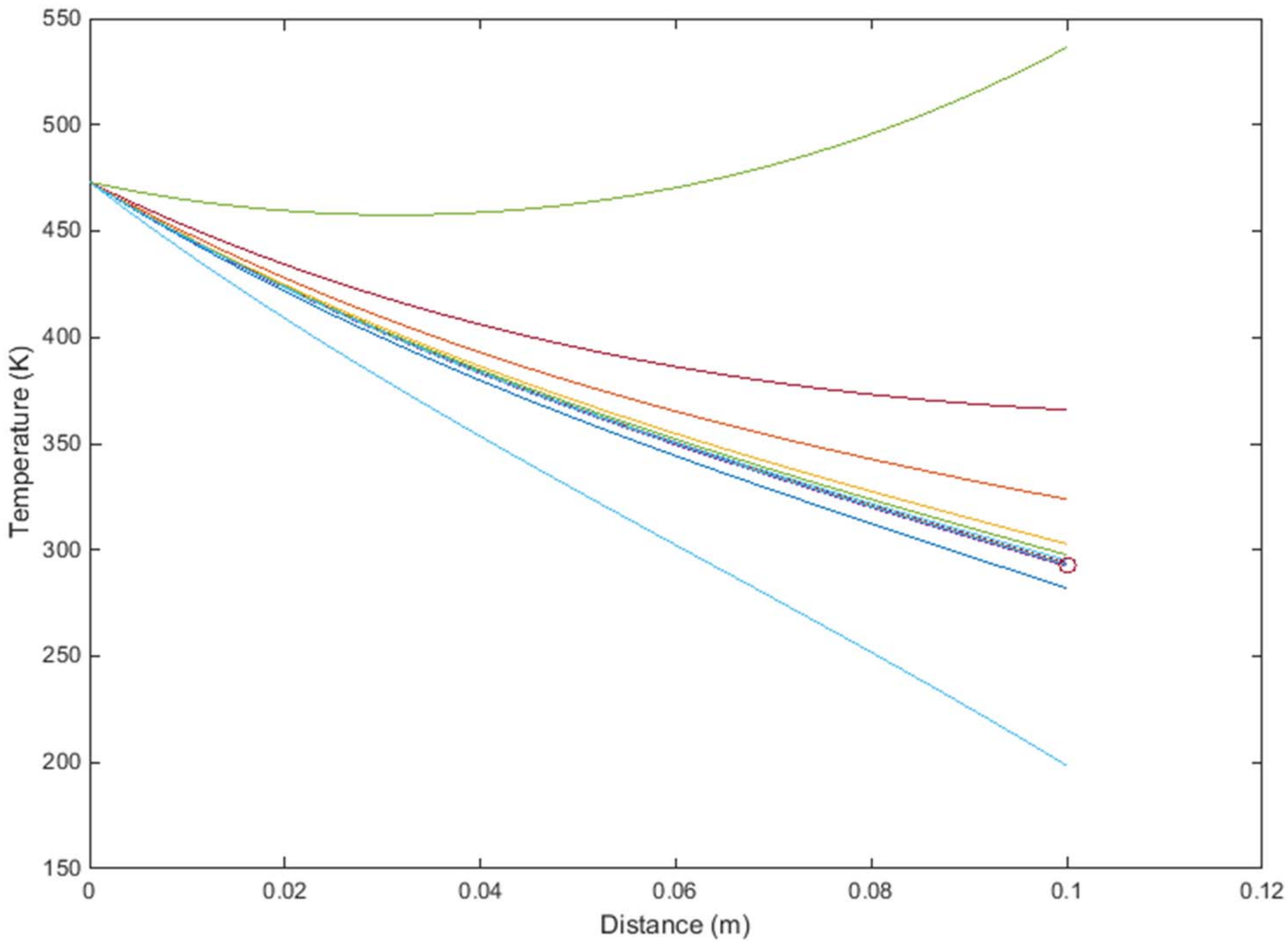
```
% Solving Example 11-1
clear all
a=0; b=0.1; TINI=473; wH=-1000; h=0.001; Tend=293;

[x, T1, w] = ...
            Sys2ODEsRK2(@odeChap11Exmp1dTdx,@odeChap11Exmp1dwdx,a,b,h,TINI,wH);
n=length(x) ;
fprintf('The temperature at x=0.1 is %5.3f, for initial value of dt/dx= %4.1f\n',...
            T1(n) ,wH)

wL=-3500;
[x, T2, w] = ...
            Sys2ODEsRK2(@odeChap11Exmp1dTdx,@odeChap11Exmp1dwdx,a,b,h,TINI,wL);
fprintf('The temperature at x=0.1 is %5.3f, for initial value of dt/dx= %4.1f\n',...
            T2(n) ,wL)
plot (x, T1,x,T2);
hold on;
```

- Shooting method using the bisection method

```
tol=0.0001; imax = 100;
for i = 1: imax + 1
    wi = (wH+wL)/2;
    [x,T,w]=Sys2ODEsRK2(@odeChap11Exmp1dTdx,@odeChap11Exmp1dwdx,a,b,h,TINI,wi);
    E=T (n)-Tend;
    if abs(E) < tol
        break
    end
    if E > 0
        wH = wi;
    else
        wL = wi;
    end
end

if i > imax
    fprintf('Solution was not obtained in %i iterations.',imax)
else
    plot(x,T);
    hold on;
    xlabel('Distance (m)'); ylabel('Temperature (K)')
    fprintf('The calculated temperature at x = 0.1 is %5.3f K.\n' ,T(n))
    fprintf('The solution was obtained in %2.0f iterations.\n' ,i)
    plot(0.1,293,'Marker','o','MarkerSize',7)
end
```

❖ Shooting method using the bisection method

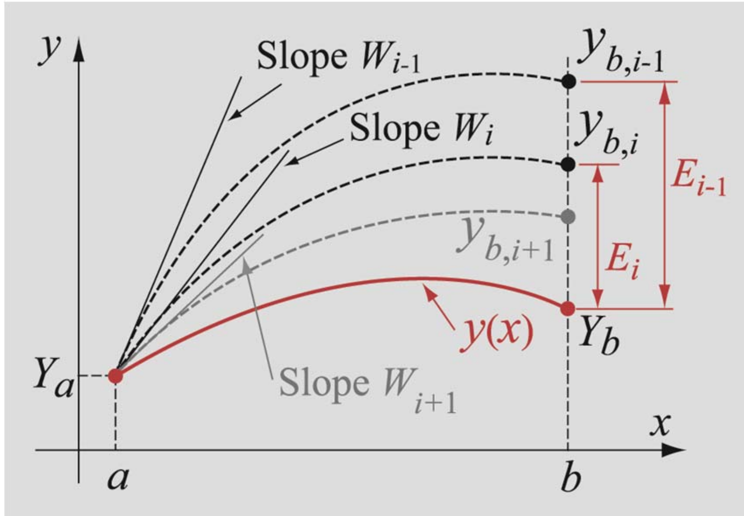
The temperature at x=0.1 is 536.502, for initial value of dt/dx= -1000.0
The temperature at x=0.1 is 198.431, for initial value of dt/dx= -3500.0
The calculated temperature at x = 0.1 is 293.000 K.
The solution was obtained in 21 iterations.

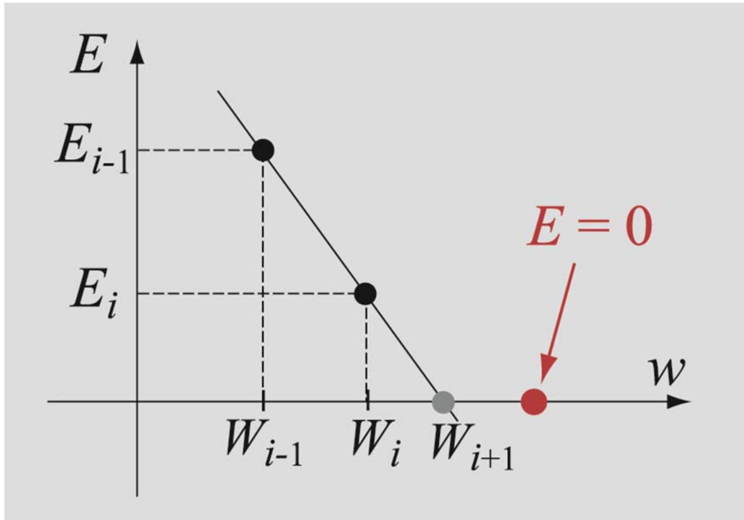


- Shooting method using the secant method

$$W_{i+1} = W_i - \frac{(W_{i-1} - W_i)}{E_{i-1} - E_i} E_i$$

  - $E_{i-1} = y_{b,i-1} - Y_b$
  - $E_i = y_{b,i} - Y_b$
  - Es from the previous two iterations can be both positive, negative, or they can have opposite signs





- Additional comments
  - BVPs with derivative, or mixed, boundary conditions
  - When derivative boundary conditions are prescribed at the endpoint, the calculated value of the derivative must be evaluated numerically.

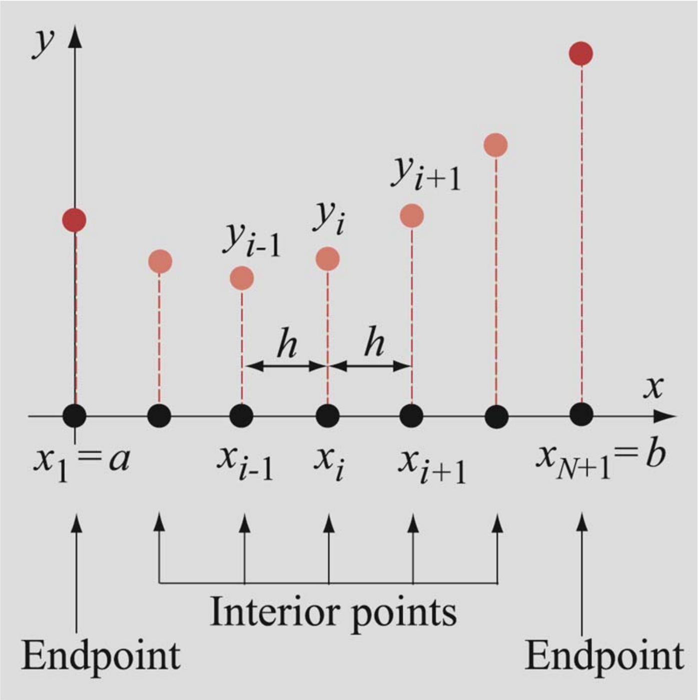
11.3 Finite Difference Method

- Finite difference method for ODE with BCs
  - The derivatives in the differential equation $\Rightarrow$ replaced with finite difference approximations
  - The domain of the solution $[a, b]$ $\Rightarrow$ divided into N subintervals of equal length h
    - Defined by (N + 1) points
    - Grid points
    - In general, subintervals can have unequal length.
  - The length of each subinterval (step size)

    $h = (b - a)/N$Points

    - End points and interior points

  - The differential equation is then written
    at each of the interior points of the domain.
    $\Rightarrow$ a system of linear algebraic equations
    when the differential equation is linear
    $\Rightarrow$ a system of nonlinear algebraic equations
    when the differential equation is nonlinear.



FD method with central difference

$$\frac{dy}{dx} = \frac{y_{i+1} - y_{i-1}}{2h} \quad \text{and} \quad \frac{d^2y}{dx^2} = \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2}$$

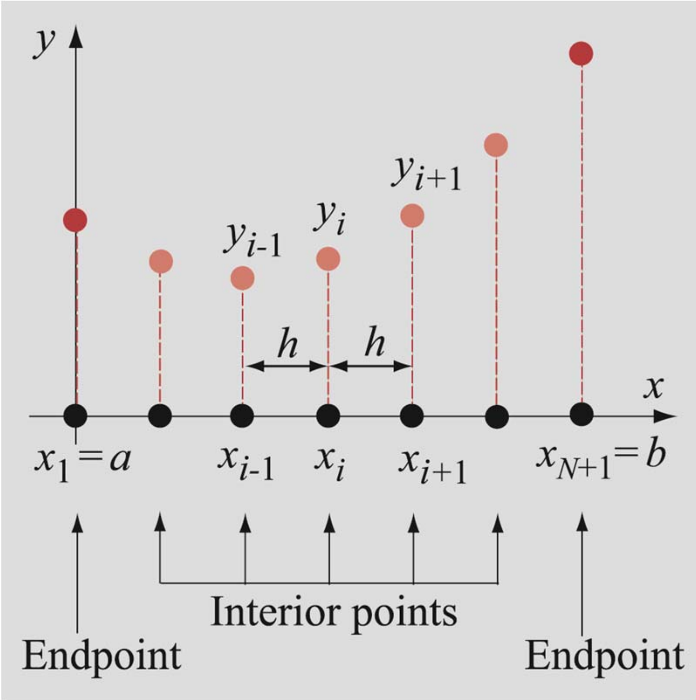
- Finite difference solution of a linear two-point BVP

$$\frac{d^2y}{dx^2} + f(x)\frac{dy}{dx} + g(x)y = h(x)$$

- Discretization

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

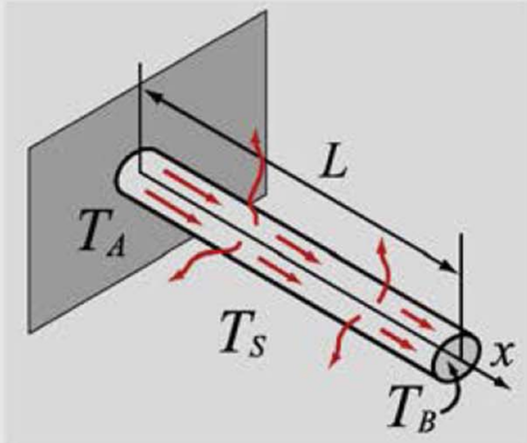
- BCs
  - y_1 and y_{N+1} are known.
- Unknowns
  - $y_2 \cdots y_N$
  - $N - 1$ linear algebraic equations



❖ Example 11-3: Temperature distribution in a pin fin. Solving a second-order linear ODE (BVP) using the finite difference method.

When only convection is included in the analysis, the steady state temperature distribution, $T(x)$, along a pin fin can be obtained from the solution of the equation:

$$\frac{d^2T}{dx^2} - \frac{h_c P}{kA_c}(T - T_S) = 0\ , \qquad 0 \le x \le L \qquad (11.26)$$



with the boundary conditions: $T(0) = T_A$ and $T(L) = T_B$.

In Eq. (11.26), h_c is the convective heat transfer coefficient, P is the perimeter bounding the cross section of the fin, k is the thermal conductivity of the fin material, A_c is the cross-sectional area of the fin, and T_S is the temperature of the surrounding air.

Determine the temperature distribution if $L = 0.1$ m, $T(0) = 473$ K, $T(0.1) = 293$ K, and $T_S = 293$ K. Use the following values for the parameters in Eq. (11.26): $h_c = 40$ W/m^2/K, $P = 0.016$ m, $k = 240$ W/m/K, and $A_c = 1.6 \times 10^{-5}$ m^2.

Solve the ODE using the finite difference method. Divide the domain of the solution into five equally spaced subintervals.

$$\frac{d^2T}{dx^2} - \frac{h_c P}{kA_c}(T - T_S) - \frac{\varepsilon\sigma_{SB}P}{kA_c}(T^4 - T_S^4) = 0\ , \ 0 \le x \le 0.1$$

Why not ?

❖ Example 11-3: Temperature distribution in a pin fin. Solving a second-order linear ODE (BVP) using the finite difference method.

- With the central difference formula

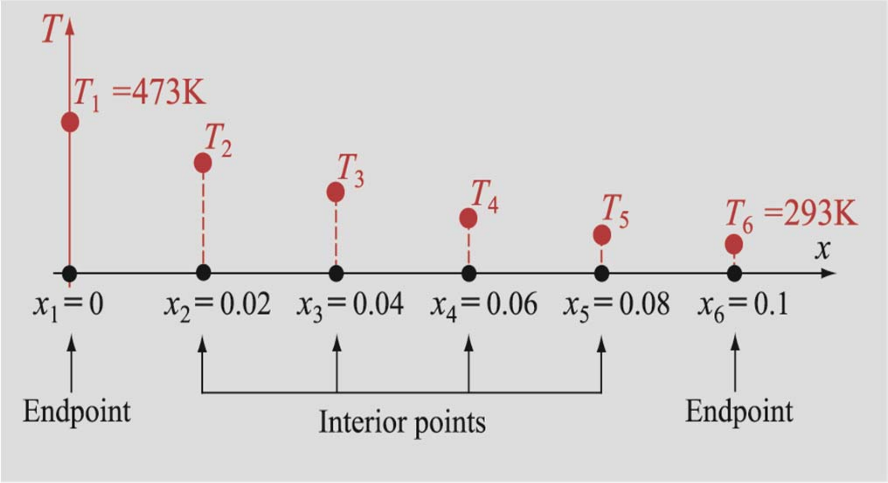
$$\frac{T_{i-1} - 2T_i + T_{i+1}}{h^2} - \beta(T_i - T_S) = 0$$

where $\beta = \dfrac{h_c P}{k A_c}$

$$T_{i-1} - (2 + h^2\beta)T_i + T_{i+1} = -h^2\beta T_S$$

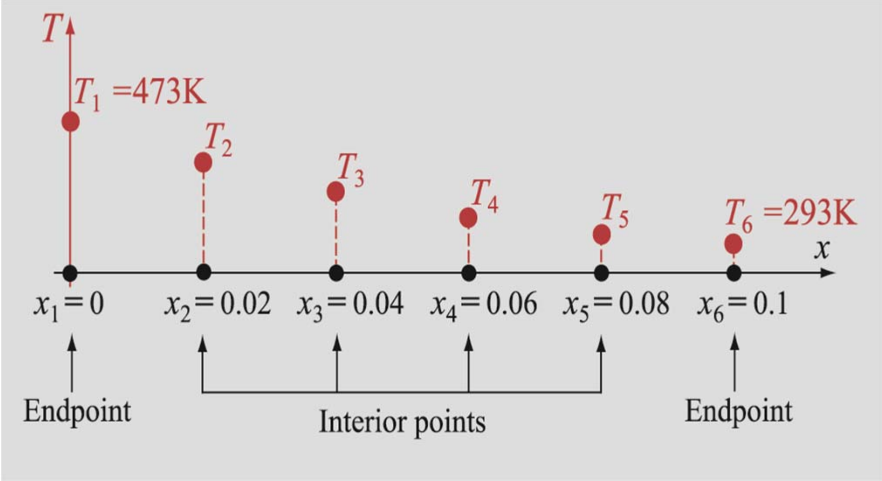
- Discretized equations

$$T_1 - (2 + h^2\beta)T_2 + T_3 = -h^2\beta T_S$$
$$T_2 - (2 + h^2\beta)T_3 + T_4 = -h^2\beta T_S$$
$$T_3 - (2 + h^2\beta)T_4 + T_5 = -h^2\beta T_S$$
$$T_4 - (2 + h^2\beta)T_5 + T_6 = -h^2\beta T_S$$



- Example 11-3: Temperature distribution in a pin fin. Solving a second-order linear ODE (BVP) using the finite difference method.
  - Rearranged discretized equations

$$-(2+h^2\beta)T_2 + T_3 = -(h^2\beta T_S + \boxed{T_1})$$
$$T_2 - (2+h^2\beta)T_3 + T_4 = -h^2\beta T_S$$
$$T_3 - (2+h^2\beta)T_4 + T_5 = -h^2\beta T_S$$
$$T_4 - (2+h^2\beta)T_5 = -(h^2\beta T_S + \boxed{T_6})$$

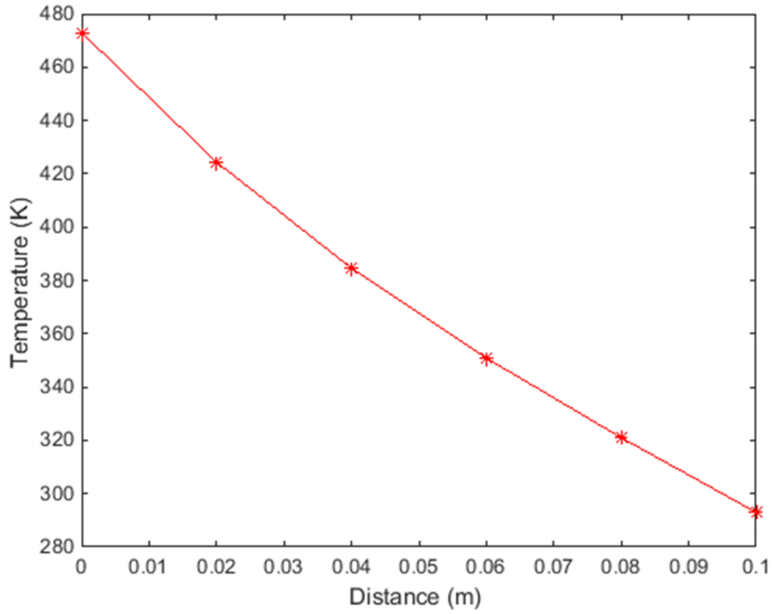


  - In matrix form, $[a][T] = [c]$

$$\begin{bmatrix} -(2+h^2\beta) & 1 & 0 & 0 \\ 1 & -(2+h^2\beta) & 1 & 0 \\ 0 & 1 & -(2+h^2\beta) & 1 \\ 0 & 0 & 1 & -(2+h^2\beta) \end{bmatrix} \begin{bmatrix} T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix} = \begin{bmatrix} -(h^2\beta T_S + T_1) \\ -h^2\beta T_S \\ -h^2\beta T_S \\ -(h^2\beta T_S + T_6) \end{bmatrix}$$

- Example 11-3: Temperature distribution in a pin fin. Solving a second-order linear ODE (BVP) using the finite difference method.

```
clear
hc = 40; P = 0.016; k = 240; Ac=1.6e-5;
h = 0.02; Ts = 293;
x = 0:0.02:0.1;
beta=hc*P/(k*Ac);aDia=-(2+h^2*beta) ;
cele=-h^2*beta*Ts;
T(1)=473; T(6)=293;

a = eye(4,4) *aDia;
for i = 1:3
    a(i,i+1) = 1;
    a(i+1,i) = 1;
end
c = [cele-T(1) ; cele; cele; cele - T(6) ];

T(2:5) = a\c;

fprintf('The temperatures at the grid points are\n')
disp(T)
plot(x,T, '-*r')
xlabel('Distance (m) ') ; ylabel('Temperature (K) ')
```



- Additional comments
  - Application of the finite difference method to an ODE does not always result in a tridiagonal system of equations as in the above illustration.
  - The numerical solution in Example 11-3 yielded a tridiagonal system because the ODE is second order and a central difference scheme is used to approximate the second derivative.

- Finite difference solution of a nonlinear two-point BVP
  - The resulting system of simultaneous equations is nonlinear.
  - The task is much more challenging than that of solving a system of linear equations.
  - The most computationally efficient means of solving a system of nonlinear equations is applying some type of iterative scheme.
  - However, as discussed in Chapter 3, iterative methods run the risk of diverging unless the starting or initial values for the iterations are close enough to the final answer.

  - Application of the finite difference method to a nonlinear ODE results in a system of nonlinear simultaneous equations.
    - One method for solving such nonlinear systems is a variant of the fixed-point iteration.

Finite difference solution of a nonlinear two-point BVP

- Fixed-point iteration
  - One method for solving such nonlinear systems is a variant of the fixed-point iteration.

$$[a][y] + [\Phi] = [b]$$

$[\Phi]$: column vector whose elements are nonlinear functions of the unknowns y

  - Linearized equation

$$[a][y]^{k+1} = [b] - [\Phi]^{k}$$

- For small number of interior points

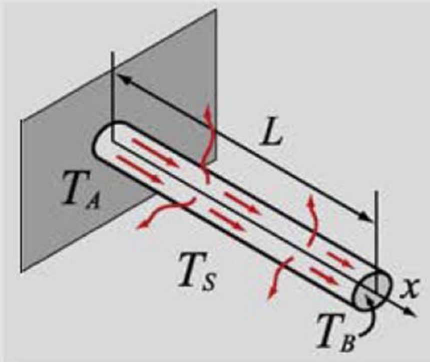
$$[y]^{k+1} = [a]^{-1}([b] - [\Phi]^{k})$$

- Usually, use a linear equation solver

- Example 11-4: Temperature distribution in a pin fin. Solving a second-order nonlinear ODE (BVP) using the finite difference method.

When convection and radiation are included in the analysis, the steady-state temperature distribution, $T(x)$, along a pin fin can be calculated from the solution of the equation:

$$\frac{d^2T}{dx^2} - \frac{h_cP}{kA_c}(T - T_S) - \frac{\varepsilon\sigma_{SB}P}{kA_c}(T^4 - T_S^4) = 0, \quad 0 \le x \le L \qquad (11.37)$$



with the boundary conditions: $T(0) = T_A$ and $T(L) = T_B$.
The definition and values of all the constants in Eq. (11.37) are given in Example 11-2.
Determine the temperature distribution if $L = 0.1\,\text{m}$, $T(0) = 473\,\text{K}$, $T(0.1) = 293\,\text{K}$, and $T_S = 293\,\text{K}$.
Solve the ODE using the finite difference method. Divide the domain of the solution into five equally spaced subintervals.

- Example 11-4: Temperature distribution in a pin fin. Solving a second-order nonlinear ODE (BVP) using the finite difference method.
  - With the central difference formula

$$\frac{T_{i-1} - 2T_i + T_{i+1}}{h^2} - \beta_A(T_i - T_S) - \beta_B(T_i^4 - T_S^4) = 0$$

$$\beta_A = \frac{h_c P}{kA_c} \text{ and } \beta_B = \frac{\varepsilon\sigma_{SB} P}{kA_c}$$

  - General form of a system of equations

$$T_{i-1} - (2 + h^2\beta_A)T_i - h^2 q\beta_B T_i^4 + T_{i+1} = -h^2(\beta_A T_S + \beta_B T_S^4)$$

- Example 11-4: Temperature distribution in a pin fin. Solving a second-order nonlinear ODE (BVP) using the finite difference method.
  - With the central difference formula

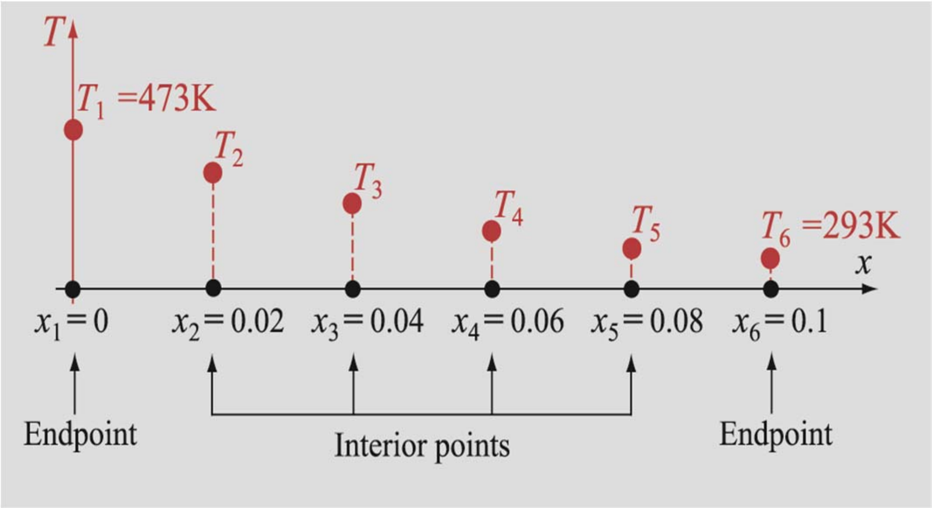
$$\frac{T_{i-1} - 2T_i + T_{i+1}}{h^2} - \beta_A(T_i - T_S) - \beta_B(T_i^4 - T_S^4) = 0$$

$$\beta_A = \frac{h_c P}{k A_c} \text{ and } \beta_B = \frac{\varepsilon \sigma_{SB} P}{k A_c}$$

  - General form of a system of equations

$$T_{i-1} - (2 + h^2\beta_A)T_i - h^2 q \beta_B T_i^4 + T_{i+1}$$

$$= -h^2(\beta_A T_S + \beta_B T_S^4)$$



- Example 11-4: Temperature distribution in a pin fin. Solving a second-order nonlinear ODE (BVP) using the finite difference method.
  - With the central difference formula

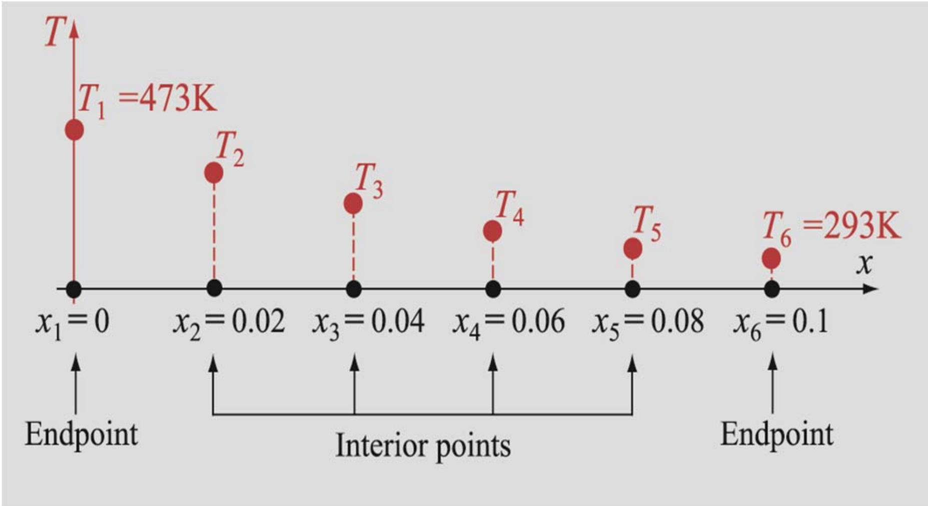
$$\frac{T_{i-1} - 2T_i + T_{i+1}}{h^2} - \beta_A(T_i - T_S) - \beta_B(T_i^4 - T_S^4) = 0$$

$$\beta_A = \frac{h_c P}{kA_c} \text{ and } \beta_B = \frac{\varepsilon\sigma_{SB} P}{kA_c}$$

  - General form of a system of equations

$$T_{i-1} - (2 + h^2\beta_A)T_i - h^2 q\beta_B T_i^4 + T_{i+1}$$

$$= -h^2(\beta_A T_S + \beta_B T_S^4)$$

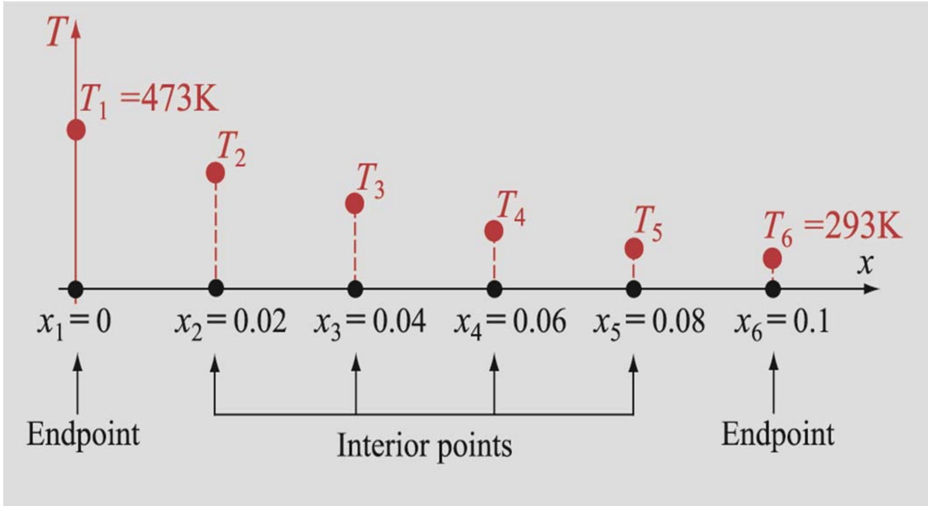


- Example 11-4: Temperature distribution in a pin fin. Solving a second-order nonlinear ODE (BVP) using the finite difference method.

$$T_{i-1} - (2 + h^2\beta_A)T_i - h^2 q\beta_B T_i^4 + T_{i+1} = -h^2(\beta_A T_S + \beta_B T_S^4)$$

- Discretized non-linear equations

$$-(2 + h^2\beta_A)T_2 - h^2\beta_B T_2^4 + T_3 = -h^2(\beta_A T_S + \beta_B T_S^4) - T_1$$
$$T_2 - (2 + h^2\beta_A)T_3 - h^2\beta_B T_3^4 + T_4 = -h^2(\beta_A T_S + \beta_B T_S^4)$$
$$T_3 - (2 + h^2\beta_A)T_4 - h^2\beta_B T_4^4 + T_5 = -h^2(\beta_A T_S + \beta_B T_S^4)$$
$$T_4 - (2 + h^2\beta_A)T_5 - h^2\beta_B T_5^4 = -h^2(\beta_A T_S + \beta_B T_S^4) - T_6$$



- Example 11-4: Temperature distribution in a pin fin. Solving a second-order nonlinear ODE (BVP) using the finite difference method.

$$[a][T]+[\Phi] = [b]$$

- Discretized non-linear equations

$$\begin{bmatrix} -(2+h^2\beta_A) & 1 & 0 & 0 \\ 1 & -(2+h^2\beta_A) & 1 & 0 \\ 0 & 1 & -(2+h^2\beta_A) & 1 \\ 0 & 0 & 1 & -(2+h^2\beta_A) \end{bmatrix} \begin{bmatrix} T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix} + \begin{bmatrix} -h^2\beta_B T_2^4 \\ -h^2\beta_B T_3^4 \\ -h^2\beta_B T_4^4 \\ -h^2\beta_B T_5^4 \end{bmatrix} = \begin{bmatrix} -h^2(\beta_A T_S + \beta_B T_S^4) - T_1 \\ -h^2(\beta_A T_S + \beta_B T_S^4) \\ -h^2(\beta_A T_S + \beta_B T_S^4) \\ -h^2(\beta_A T_S + \beta_B T_S^4) - T_6 \end{bmatrix}$$

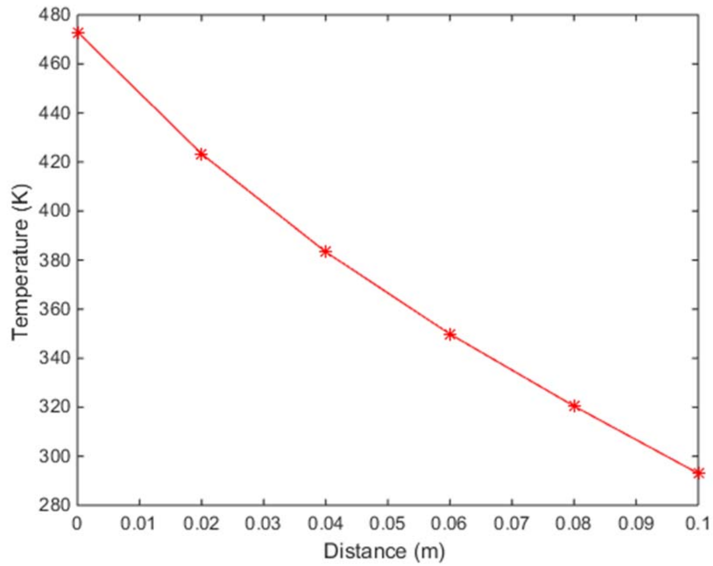
$$[T]^{k+1} = [a]^{-1}([b]-[\Phi]^k)$$

❖ Example 11-4: Temperature distribution in a pin fin. Solving a second-order nonlinear ODE (BVP) using the finite difference method.

```
clear
hc = 40; P = 0.016; k = 240; Ac=1.6e-5; epsln = 0.4; seg = 5.67E-8;
betaA = hc*P/ (k*Ac) ; betaB = epsln*seg*P/ (k*Ac) ; Ts = 293;
N = 5; h = 0.1/N;
x = 0:h:0.1;
aDia = - (2 + h^2*betaA) ;
bele = -h^2* (betaA*Ts + betaB*Ts^4) ;
h2betaB = h^2*betaB;
Ti(1) = 473; Ti (N + 1) = 293;
Tnext(1) = Ti (1) ; Tnext (N + 1) = Ti (N + 1) ;
a = eye (N - 1,N - 1) *aDia;

for i = 1:N - 2
    a (i,i + 1) = 1;
    a (i + 1,i) = 1;
end

ainv = inv (a) ;
```

- Example 11-4: Temperature distribution in a pin fin. Solving a second-order nonlinear ODE (BVP) using the finite difference method.

```
b (1) = bele - Ti (1) ; b (N-1) = bele - Ti (N+1) ;
b (2:N - 2) = bele;
Ti (2:N) = 400;

for i = 1:4
    phi = -h2betaB*Ti(2:N).^4';
    Tnext (2:N) = ainv*(b' - phi) ;
    Ti = Tnext;
    fprintf('Iteration number%d, Temperatures:\n'...
        ,i )
    fprintf('%10.2f',Tnext); fprintf('\n')
end

plot (x,Tnext,'-*r')
xlabel ('Distance (m) ') ; ylabel ('Temperature (K) ')
```




```
Iteration number1, Temperatures:
  473.00   423.23   382.83   349.11   319.82   293.00
Iteration number2, Temperatures:
  473.00   423.35   383.32   349.85   320.45   293.00
Iteration number3, Temperatures:
  473.00   423.34   383.31   349.84   320.45   293.00
Iteration number4, Temperatures:
  473.00   423.34   383.31   349.84   320.45   293.00
```

- Finite difference solution of a linear BVP with mixed boundary conditions
  - A constraint that involves the derivative is prescribed at one or both of the endpoints
  - The system of algebraic equations cannot be solved since the solution at the endpoints is not given
  - The additional equations needed for solving the problem are obtained by discretizing the boundary conditions using finite differences, and incorporating the resulting equations into the algebraic equations for the interior points.

$$c_1 \left.\frac{dy}{dx}\right|_{x=a} + c_2\, y(a) = C_a \qquad c_3 \left.\frac{dy}{dx}\right|_{x=b} + c_4\, y(b) = C_b$$

- Final notes
  - Neither the finite difference method nor the shooting method has a clear advantage
  - The FDM requires the solution of a nonlinear system of equations
  - Shooting method requires information regarding the higher derivatives of the dependent variable at the leftmost boundary.
  - The choice of which method to use is therefore problem-dependent, depending either on how easily initial guesses can be generated for the derivatives of the dependent variable at a boundary (shooting method) or on how well a particular fixed-point iteration scheme converges (finite difference method)

- Example 11-5: Solving a BVP with mixed boundary conditions.

Use the finite difference method to solve the following mixed boundary condition BVP.

$$-2\frac{d^2y}{dx^2} + y = e^{-0.2x}, \quad \text{for} \quad 0 \le x \le 1 \tag{11.46}$$

with the boundary conditions: $y(0) = 1$ and $\left.\frac{dy}{dx}\right|_{x=1} = -y|_{x=1}$.

Divide the solution domain into eight subintervals, and use the central difference approximation for all derivatives. Compare the numerical solution with the exact solution:

$$y = -0.2108e^{x/(\sqrt{2})} + 0.1238e^{-x/(\sqrt{2})} + \frac{e^{-0.2x}}{0.92} \tag{11.47}$$

- Example 11-5: Solving a BVP with mixed boundary conditions.
  - Central difference

$$-2\left(\frac{y_{i-1}-2y_i+y_{i+1}}{h^2}\right)+y_i = e^{-0.2x_i}$$

  - General form of discretized equation

$$-2y_{i-1}+(4+h^2)y_i-2y_{i+1} = h^2e^{-0.2x_i}$$

  - At each grid point

$$-2\boxed{y_1}+(4+h^2)y_2-2y_3 = h^2e^{-0.2x_2}$$
$$-2y_2+(4+h^2)y_3-2y_4 = h^2e^{-0.2x_3}$$
$$-2y_3+(4+h^2)y_4-2y_5 = h^2e^{-0.2x_4}$$
$$-2y_4+(4+h^2)y_5-2y_6 = h^2e^{-0.2x_5}$$

$$-2y_5+(4+h^2)y_6-2y_7 = h^2e^{-0.2x_6}$$
$$-2y_6+(4+h^2)y_7-2y_8 = h^2e^{-0.2x_7}$$
$$-2y_7+(4+h^2)y_8-2\boxed{y_9} = h^2e^{-0.2x_8}$$

$$y_1 = y(0) = 1$$

- Example 11-5: Solving a BVP with mixed boundary conditions.
  - The other boundary condition

$$\left.\frac{dy}{dx}\right|_{x=1} = -y$$

    - One-sided backward formula that uses values at the previous points: first-order accuracy
    - Three-point backward difference formula: second-order accuracy

  - Discretized boundary condition

$$\frac{dy}{dx} = \frac{y_{i-2} - 4y_{i-1} + 3y_i}{2h} \Rightarrow \frac{y_7 - 4y_8 + 3y_9}{2h} = -y_9 \Rightarrow y_9 = \frac{-1}{3+2h}y_7 + \frac{4}{3+2h}y_8$$

  - Modified linear equation

$$-2y_7 + (4+h^2)y_8 - 2y_9 = h^2 e^{-0.2x_8} \Rightarrow$$

$$\left(\frac{2}{3+2h} - 2\right)y_7 + \left(4 + h^2 - \frac{8}{3+2h}\right)y_8 = h^2 e^{-0.2x_8}$$

❖ Example 11-5: Solving a BVP with mixed boundary conditions.

$$-2y_{i-1} + (4+h^2)y_i - 2y_{i+1} = h^2 e^{-0.2x_i}$$

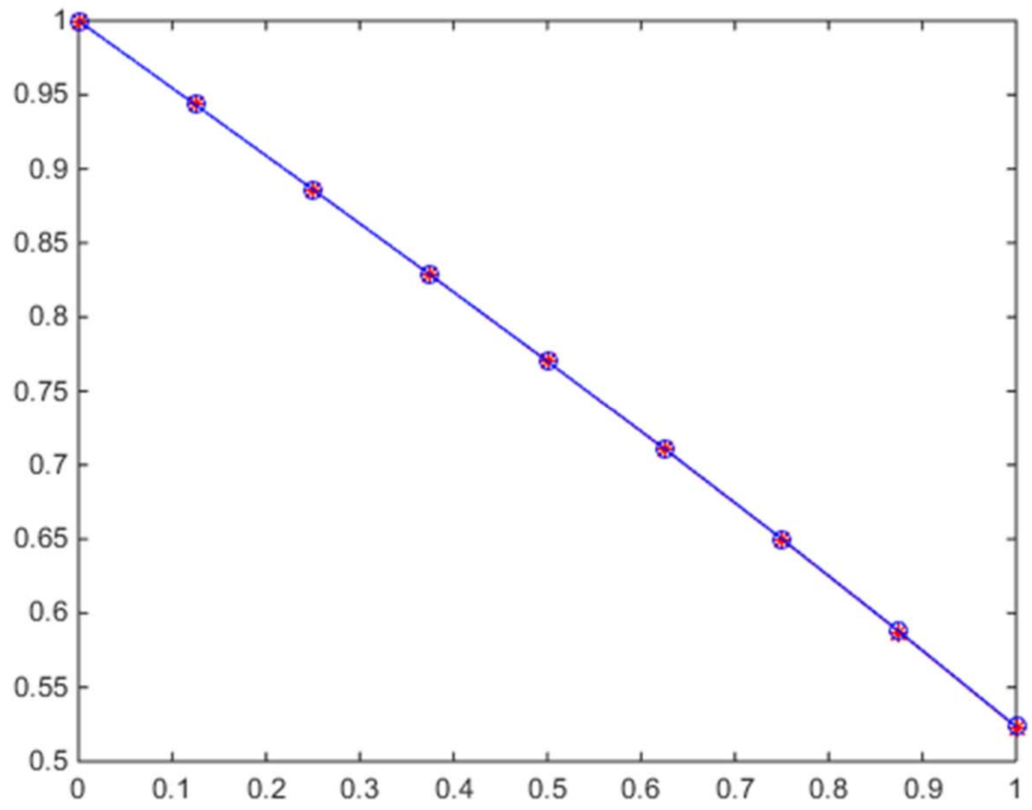
$$\left(\frac{2}{3+2h} - 2\right)y_7 + \left(4 + h^2 - \frac{8}{3+2h}\right)y_8 = h^2 e^{-0.2x_8}$$

- System of the linear equations

$$\begin{bmatrix} (4+h^2) & -2 & 0 & 0 & 0 & 0 & 0 \\ -2 & (4+h^2) & -2 & 0 & 0 & 0 & 0 \\ 0 & -2 & (4+h^2) & -2 & 0 & 0 & 0 \\ 0 & 0 & -2 & (4+h^2) & -2 & 0 & 0 \\ & & & -2 & (4+h^2) & -2 & 0 \\ & & & & -2 & (4+h^2) & -2 \\ & & & & & \left(\frac{2}{3+2h} - 2\right) & \left(4 + h^2 - \frac{8}{3+2h}\right) \end{bmatrix} \begin{bmatrix} y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} 2 + h^2 e^{-0.2x_2} \\ h^2 e^{-0.2x_3} \\ h^2 e^{-0.2x_4} \\ h^2 e^{-0.2x_5} \\ h^2 e^{-0.2x_6} \\ h^2 e^{-0.2x_7} \\ h^2 e^{-0.2x_8} \end{bmatrix}$$

- Example 11-5: Solving a BVP with mixed boundary conditions.

```
% Solution of Chapter 11 Example 5
clear
a = 0; b = 1;
N = 8; h = (b - a)/N;
x = a:h:b;
hDenom = 3 + 2*h;
aDia = (4 + h^2);
y(1) = 1;

a = eye(N - 2,N - 2)*aDia;
a(N - 1,N - 1) = aDia - 8/hDenom;
for i=1:N-2
    a(i,i+1)=-2;
    a(i+1,i)=-2;
End

a(N-1,N-2) = 2/hDenom-2;
c(1)=2+h^2*exp(-0.2*x(2));
c(2:N-1)=h^2*exp(-0.2*x(3:N));
y(2:N)=Tridiagonal(a,c);
y(N+1)=-1*y(N-1)/hDenom+4*y(N)/hDenom;
yExact=-0.2108.*exp(x./sqrt(2))+0.1238.*exp(-x./sqrt(2))+exp(-0.2.*x)./0.92;

plot(x,y,'-*r',x,yExact,'-ob')
```



11.4 Use of MATLAB Built-in Functions for Solving BVPs

- Built-in function for ODEs

$$\frac{d^2y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right) \text{ for } a \le x \le b \text{ with } y(a) = Y_a \text{ and } y(b) = Y_b$$

- System of the first-order ODEs

$$\frac{dy}{dx} = w \quad \text{and} \quad \frac{dw}{dx} = f(x, y, w)$$

```
sol = bvp4c(odefun,bcfun,solinit)
```

- 'odefun'
  - Name of the user-defined function
  - String (i.e., 'odefun') or by using a handle (i.e.,@odefun)

$$\frac{dy}{dx} = w \quad \text{and} \quad \frac{dw}{dx} = f(x, y, w)$$

```
dydx = odefun(x,yw)
```

11.4 Use of MATLAB Built-in Functions for Solving BVPs

- Built-in function for ODEs

```
sol = bvp4c(odefun,bcfun,solinit)
```

- 'bcfun'
  - The name of the user-defined function (function file) that computes the residual in the boundary condition.
  - The residual is the difference between the numerical solution and the prescribed boundary conditions.
  - 'bcfun' or @bcfun

```
res = bcfun(ya,yb)
```

  - ya and yb : column vectors corresponding to the numerical solution at $x = a$ and at $x = b$
  - $ya(1)$ and $yb(1)$, are the values of y at $x = a$ and $x = b$.
  - $ya(2)$ and $yb(2)$, are the values of dy/dx at $x = a$ and $x = b$.

Dirichlet BC $\begin{bmatrix} ya(1) - Y_a \\ yb(1) - Y_b \end{bmatrix}$ Neumann BC $\begin{bmatrix} ya(2) - D_a \\ yb(2) - D_b \end{bmatrix}$

11.4 Use of MATLAB Built-in Functions for Solving BVPs

- Built-in function for ODEs
  - 'bcfun'
    - For mixed BCs

Boundary condition:

$$y(a) = Y_a \quad \text{and} \quad \left.\frac{dy}{dx}\right|_{x=b} = D_b$$

vector `res` is: $\begin{bmatrix} ya(1) - Y_a \\ yb(2) - D_b \end{bmatrix}$

Boundary condition:

$$\left.\frac{dy}{dx}\right|_{x=a} = D_a \quad \text{and} \quad y(b) = Y_b$$

vector `res` is: $\begin{bmatrix} ya(2) - D_a \\ yb(1) - Y_b \end{bmatrix}$

Boundary condition (general case):

$$c_1 \left.\frac{dy}{dx}\right|_{x=a} + c_2\, y(a) = C_a \quad \text{and}$$

$$c_3 \left.\frac{dy}{dx}\right|_{x=b} + c_4\, y(b) = C_b$$

vector `res` is (for $c_1, c_3 \neq 0$):

$$\begin{bmatrix} ya(2) - \dfrac{C_a}{c_1} + \dfrac{c_2}{c_1} ya(1) \\ yb(2) - \dfrac{C_b}{c_3} + \dfrac{c_4}{c_3} yb(1) \end{bmatrix}$$

11.4 Use of MATLAB Built-in Functions for Solving BVPs

- Built-in function for ODEs

```
sol = bvp4c(odefun,bcfun,solinit)
```

- ‘solinit’

```
solinit = bvpinit(x,yinit)
```

  - A structure containing the initial guess for the solution.
  - ‘solinit’ is created by a built-in MATLAB function named ‘bvpinit’.
  - The input argument x: a vector that specifies the initial interior points.
    - For a BVP with the domain $[a, b]$, the first element of x is a, and the last element is b.
    - Often an initial number of ten points is adequate: x=linspace(a, b, 10)

  - The input argument $yinit$: initial guess for the solution (interior points)
  - $yinit$: vector that has one element for each of the dependent variables
    - In the case of two equations,
    - The first element is the initial guess for the value of y.
    - The second element is the initial guess for the value of w.

11.4 Use of MATLAB Built-in Functions for Solving BVPs

- Built-in function for ODEs

```
sol = bvp4c(odefun,bcfun,solinit)
```

- 'sol'
  - A structure containing the solution.
  - sol.x
    - The x coordinate of the interior points.
    - The number of interior points is determined during the solution process by MATLAB.
    - It is, in general, not the same as was entered by the user in bvpinit.
  - sol.y
    - The numerical solution, $y(x)$, which is the y value at the interior points.
  - sol.yp
    - The value of the derivative, dy/dx at the interior points.

11.4 Use of MATLAB Built-in Functions for Solving BVPs

- Example 11-6: Solving a two-point BVP using MATLAB's built-in function bvp4c

Use MATLAB's built-in function `bvp4c` to solve the following two-point BVP.

$$\frac{d^2y}{dx^2} + 2x\frac{dy}{dx} + 5y - \cos(3x) = 0, \quad \text{for} \quad 0 \le x \le \pi \tag{11.65}$$

with the boundary conditions: $y(0) = 1.5$ and $y(\pi) = 0$.

- Rewrite the ODE

$$\frac{d^2y}{dx^2} = -2x\frac{dy}{dx} - 5y + \cos(3x)$$

- Transform into a system of the first-order ODEs

$$\frac{dy}{dx} = w$$

$$\frac{dw}{dx} = -2xw - 5y + \cos(3x)$$

11.4 Use of MATLAB Built-in Functions for Solving BVPs

❖ Example 11-6: Solving a two-point BVP using MATLAB's built-in function bvp4c

```
sol = bvp4c(odefun,bcfun,solinit)
```

- odefun

```
function dydx = odefunExample6(x,yw)
dydx = [yw(2)
   -2*x*yw(2) - 5*yw(1) + cos(3*x)];
```

$$\frac{dy}{dx} = w$$

$$\frac{dw}{dx} = -2xw - 5y + \cos(3x)$$

- bcfun

```
function res = bcfunExample6(ya,yb)
BCa = 1.5; BCb = 0;
res = [ya(1) - BCa
     yb(1) - BCb];
```

- solinit

```
solinit=bvpinit(linspace(0,pi,20),[0.2, 0.2]);
```

11.4 Use of MATLAB Built-in Functions for Solving BVPs

- Example 11-6: Solving a two-point BVP using MATLAB's built-in function bvp4c

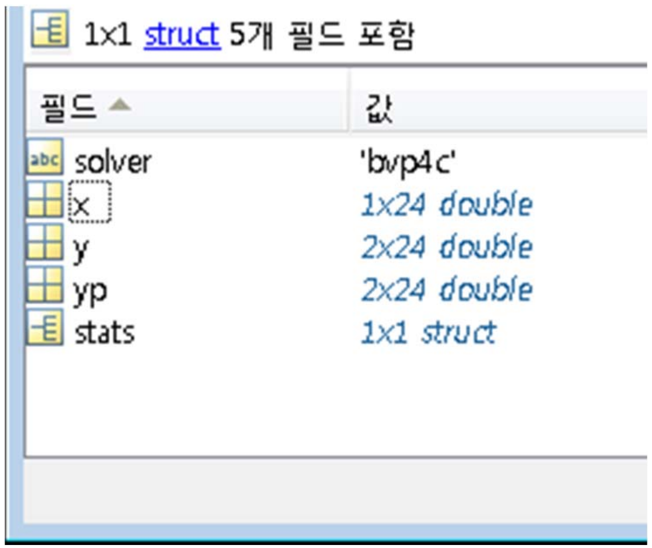
```
sol = bvp4c(odefun,bcfun,solinit)
```

```
function Example11_6_Script
    clear all;
    solinit=bvpinit(linspace(0, pi, 20),[0.2, 0.2]);
    sol=bvp4c(@odefunExample6,@bcfunExample6,solinit);
    plot(sol.x, sol.y(1,:),'r')
    xlabel ('x'); ylabel ('y')
end

function dydx = odefunExample6(x,yw)
    dydx = [yw(2); -2*x*yw(2) - 5*yw(1) + cos(3*x)];
end

function res = bcfunExample6(ya,yb)
    BCa = 1.5; BCb = 0;
    res = [ya(1) - BCa; yb(1) - BCb];
end
```

11.4 Use of MATLAB Built-in Functions for Solving BVPs

- Example 11-6: Solving a two-point BVP using MATLAB's built-in function bvp4c

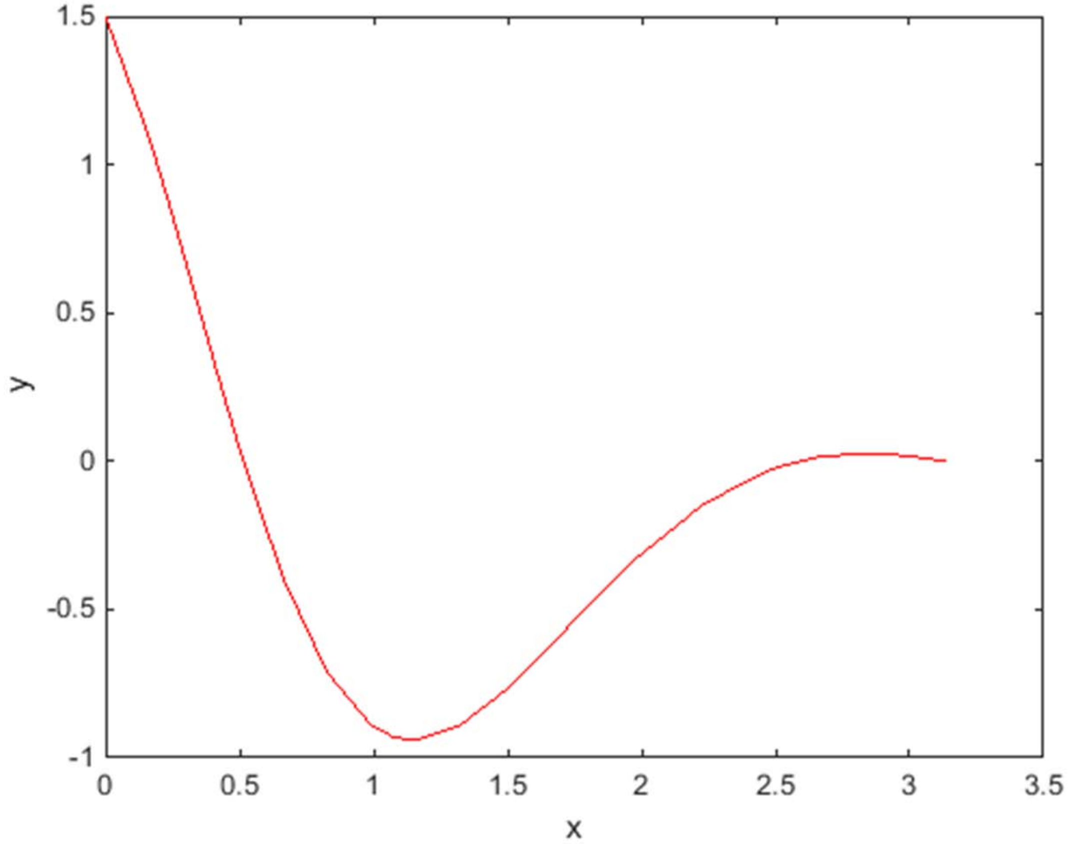
```
sol = bvp4c(odefun,bcfun,solinit)
```



sol.x(:)

sol.y(1,:)

sol.y(2,:)



11.5 Error and Stability in Numerical Solution of BVPs

- Numerical error
  - For the shooting method
    - The numerical error is the same as for the initial value problem and depends on the method used.
    - BVP $\Rightarrow$ a series of IVPs with guesses for the leftmost boundary condition
  - In the case of the finite difference method
    - The error is determined by the order of accuracy of the numerical scheme used.
    - The truncation errors of the different approximations used for the derivatives
    - The accuracy of the solution by the finite difference method is determined by the larger of the two truncation errors:
      - That of the difference scheme used for the differential equation
      - That of the difference scheme used to discretize the boundary conditions
    - An effort must therefore be made to ensure that the order of the truncation error is the same for the boundary conditions and the differential equation.

11.5 Error and Stability in Numerical Solution of BVPs

- Stability
  - In an IVP, the instability was associated with error that grew as the integration progressed.
  - In contrast, in BVPs, the growth of numerical error as the solution progresses is limited by the boundary conditions.
  - For shooting method
    - In some cases, there may be valid multiple solutions to the BVP so that when it is solved as an IVP, small changes in the initial constraint (i.e., leftmost boundary condition) can produce one solution or the other for a small change in the leftmost boundary condition.
    - In some cases, the differential equation itself may be unstable to small perturbations in the boundary conditions, in which case the problem formulation has to be examined.
    - In other cases, multiple, valid solutions to the ODE exist for different rightmost boundary conditions.
  - For FDM,
    - Stability of solving a BVP by finite differences rests on stability of the scheme used to solve the resulting set of simultaneous equations.
    - For non-linear systems , stability is determined by the type of method used to solve the system as well as the proximity of the initial guess to the solution.