COMPUTER AIDED ENGINEERING LAB [ME404]

NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

Prof. Krishna Prakash Yadav
Department of Mechanical Engineering
National Institute of Technology Warangal

TOPIC

- Solving Engineering Problems Involving PDEs
- Solving Engineering Problems Involving Vibrations
- Solving Problems Involving Optimization

Partial Differential Equations (PDEs)

- A Partial Differential Equation (PDE) is a type of differential equation that involves partial derivatives of a multivariable function.
- If $u(x, y, t)$ is a function of multiple independent variables (like space x, y and time t), then PDEs describe how u changes with respect to those variables.
- A general second-order PDE in two variables x, y is:

$$A\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} + D\frac{\partial u}{\partial x} + E\frac{\partial u}{\partial y} + Fu = G(x, y)$$

  - Where: $u(x, y)$ = dependent variable
    - x, y= independent variables
    - Coefficients A, B, C, D, E, F can be constants or functions

Types of PDEs

- **Discriminant Rule**

  The type of PDE depends on the **discriminant**:

$$D = B^2 - 4AC$$

1. **Elliptic PDE** if $D < 0$
   - Example: Laplace Equation $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$
   - (Steady-state problems, electrostatics, heat conduction in steady state)
2. **Parabolic PDE** if $D = 0$
   - Example: Heat Equation $\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$
   - (Diffusion, heat transfer problems)
3. **Hyperbolic PDE** if $D > 0$
   - Example: Wave Equation $\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$
   - (Vibration, sound waves, elastic waves)

Solving PDEs Approaches in MATLAB

- MATLAB provides two main approaches:
  1. Analytical approach using **Symbolic Math Toolbox** ($pdepe, dsolve,$ $symbolic\ functions$).

     command : $pdepe(m, pdefun, icfun, bcfun, x, t);$

     Where: m=symmetricity, $pdefun$= pde function, $icfun$= intial condition function, $bcfun$= boundary condition function, x, t= independent variables

  2. Numerical approach using **PDE Toolbox** ($solvepde, finite\ element\ methods$).

General PDE Form

- MATLAB PDE Toolbox solves PDEs in the general scalar form

$$m\frac{\partial^2 u}{\partial t^2} + d\frac{\partial u}{\partial t} - \nabla * (c\nabla u) + au = f$$

Where: $u(x, y, t)$ = dependent variable (temperature, displacement, concentration, etc.)

- m = mass coefficient (wave, structural problems)
- d = damping coefficient (heat, diffusion)
- c = diffusion/conductivity/stiffness matrix
- a = absorption/reaction coefficient
- f = source term

Standard PDEs in MATLAB

1. **Heat Equation (Diffusion)**

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u + Q \ , \qquad m\frac{\partial^2 u}{\partial t^2} + d\frac{\partial u}{\partial t} - \nabla * (c\nabla u) + au = f$$

Where: $u(x,t)$ = temperature

- $\alpha = k/(\rho c_p)$ thermal diffusivity
- In MATLAB PDE Toolbox: $m = 0, d = 1, c = \alpha, a = 0, f = Q$

2. **Wave Equation**

$$\frac{\partial^2 u}{\partial t^2} = c^2 \ \nabla^2 u$$

Where: c = wave speed

- MATLAB form: $m = 1, d = 0, c = c^2, a = 0, f = 0$

3. **Laplace Equation (Steady-State Heat/Static)**

$$\nabla^2 u = 0$$

Where: MATLAB form: $m = 0, d = 0, c = 1, a = 0, f = 0$

4. Poisson Equation

$$\nabla^2 u = -f$$

Where: MATLAB form: $m = 0, d = 0, c = 1, a = 0, f = f(x, y)$

5. Convection–Diffusion Equation

$$\frac{\partial y}{\partial x} + \vec{v} \cdot \nabla u = D\nabla^2 u + Q$$

Where: MATLAB form: $m = 0, d = 1, c = D, a = \vec{v} \cdot \nabla, f = Q$

6. Structural Mechanics (Elasticity PDE)

$$\rho \frac{\partial^2 u}{\partial t^2} - \nabla \cdot \sigma = f$$

with stress tensor $\sigma = C : \varepsilon, \varepsilon = \frac{1}{2}(\nabla u + \nabla u^T)$

Boundary Conditions

1. **Dirichlet Boundary Condition (Essential BC)**

$$u = g \quad \text{on boundary}$$

- Means the value of the solution u is fixed on the boundary.
- Example: Temperature at left edge fixed at $100℃$, $u = 100$.
- Cammand: $applyBoundaryCondition(model,'dirichlet','Edge',1,'u',g);$

2. **Neumann Boundary Condition (Natural BC)**

$$n \cdot (c\nabla u) = q \quad \text{on boundary}$$

- Meaning:
  - Controls the flux (gradient) of u across the boundary.
  - n = outward normal vector to the boundary
  - $c\nabla u$ = flux term (e.g., heat flux, stress, diffusion flux)
  - Q = specified value of flux
- Examples: Heat conduction: $-k\frac{\partial u}{\partial n} = q$ (prescribed heat flux at surface)
  - Fluid diffusion: flux of concentration at boundary
- Cammand: $applyBoundaryCondition(model,'neumann','Edge',1,'g',q);$

3. Robin (Mixed) Boundary Condition

$$n \cdot (c\nabla u) + hu = r$$

- Meaning:
  - A **combination of Dirichlet and Neumann** conditions.
  - First term = flux across boundary $n \cdot (c\nabla u)$
  - Second term = a proportional part of solution hu
  - Right side r = balance term
- Example: Convective heat transfer (Newton's law of cooling):

  $-k\frac{\partial T}{\partial n} = h(T - T_{\infty})$ which matches the Robin form.

  - It means boundary temperature is not fixed, but depends on outside environment.
- Cammand: $applyBoundaryCondition(model,'mixed','Edge',1,'g',h,'q',r);$

- In pdepe, the heat transfer PDE system has the form:

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial t} = x^{-m}\frac{\partial}{\partial x}\left(x^m f\left(x,t,u,\frac{\partial u}{\partial x}\right)\right) + s(x,t,u,\frac{\partial u}{\partial x})$$

Where: $c \rightarrow$ coefficient multiplying time derivative (like thermal capacity)

- $f \rightarrow$ flux term (related to conduction/diffusion/transport)
- $s \rightarrow$ source term (like heat generation, reaction term, etc.

- In MATLAB, the pde function written as:

$$pdefun\ = @(x,t,u,dudx)\, deal(c,f,s);$$

- The boundary conditions must be of the form:

$$p(x,t,u) + q(x,t)f\left(x,t,u,\frac{\partial u}{\partial x}\right) = 0$$

at the left boundary $(x\ =\ xl)$ and right boundary $(x\ =\ xr)$.

- Note:
  powers1234 = @(x) deal(x,x.^2,x.^3,x.^4)
  [X1,X2,X3,X4] = powers1234(2)
  Ans: X1 = 2, X2 = 4, X3 = 8, X4 = 16
  deal command return multiple outputs from an anonymous function.

- In MATLAB, the boundary condition function written as:

$$bcfun\ = @(xl,ul,xr,ur,t)\, deal(pl,ql,pr,qr);$$

Where: $xl, ul \rightarrow$ left boundary position and solution value

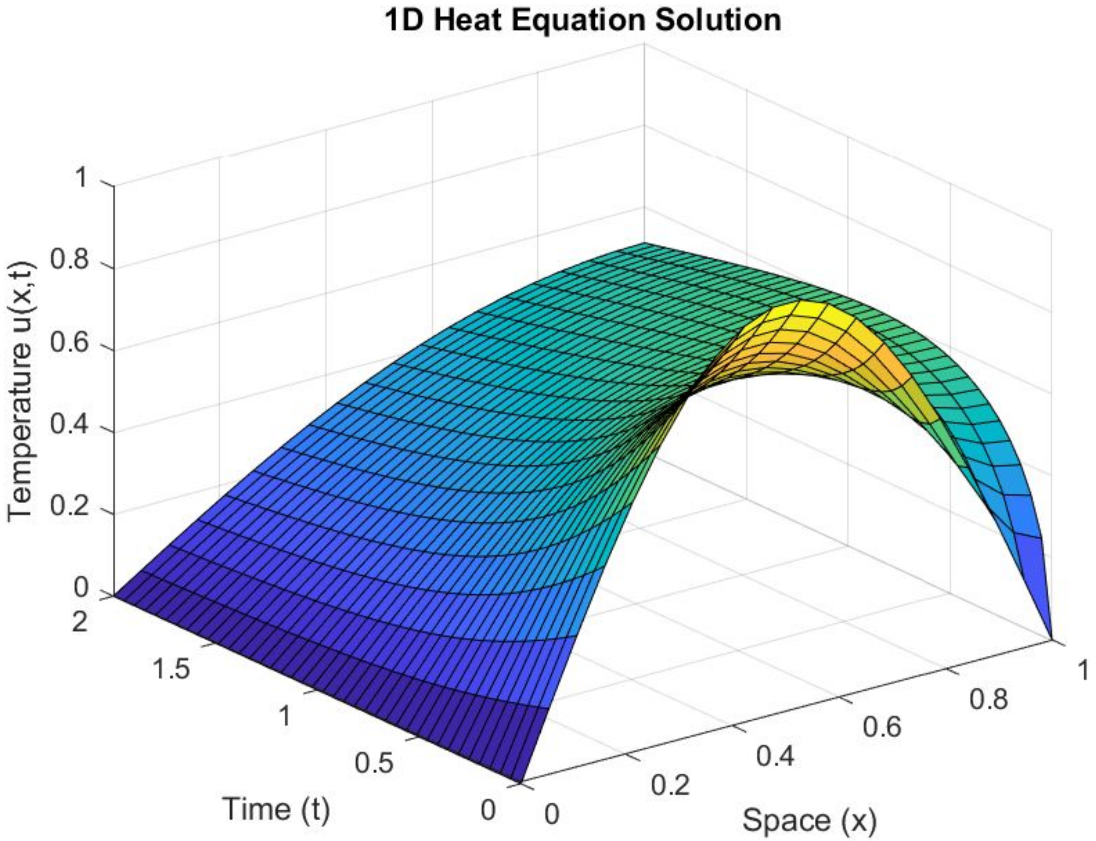
- $xr, ur \rightarrow$ right boundary position and solution value
- $t \rightarrow$ time
- $pl, ql \rightarrow$ define left BC: $pl\ +\ ql * f\ =\ 0$
- $pr, qr \rightarrow$ define right BC: $pr\ +\ qr * f\ =\ 0$

Solving PDEs in MATLAB

- **Example**: Solve the given heat equation in MATLAB and plot the heat diffusion process using Symbolic Math Toolbox $\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, 0 < x < 1, t > 0, u(x,0) = \sin(\pi x), u(0,t) = 0, u(1,t) = 0$
- **Solution**: $\%c\left(x,t,u,\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial t} = x^{-m}\frac{\partial}{\partial x}\left(x^m f\left(x,t,u,\frac{\partial u}{\partial x}\right)\right) + s(x,t,u,\frac{\partial u}{\partial x}), c = 1, s = 0$

```
clc;clear;
m = 0; % Symmetry (slab geometry)
alpha = 0.1;
% PDE function for pdepe
pdefun = @(x,t,u,dudx) deal(1, alpha*dudx, 0);
% Initial condition
icfun = @(x) sin(pi*x);
% Boundary condition
bcfun = @(xl,ul,xr,ur,t) deal(0, ul, 0, ur);      %pl=0,ql=ul,pr=0,qr=ur
```

```
% Solve PDE
x = linspace(0,1,20);
t = linspace(0,2,40);
sol = pdepe(m,pdefun,icfun,bcfun,x,t)
% Plot results
surf(x,t,sol)
xlabel('Space (x)')
ylabel('Time (t)')
zlabel('Temperature u(x,t)')
title('1D Heat Equation Solution’)
```



Steps to Solve PDEs in MATLAB (PDE Toolbox)

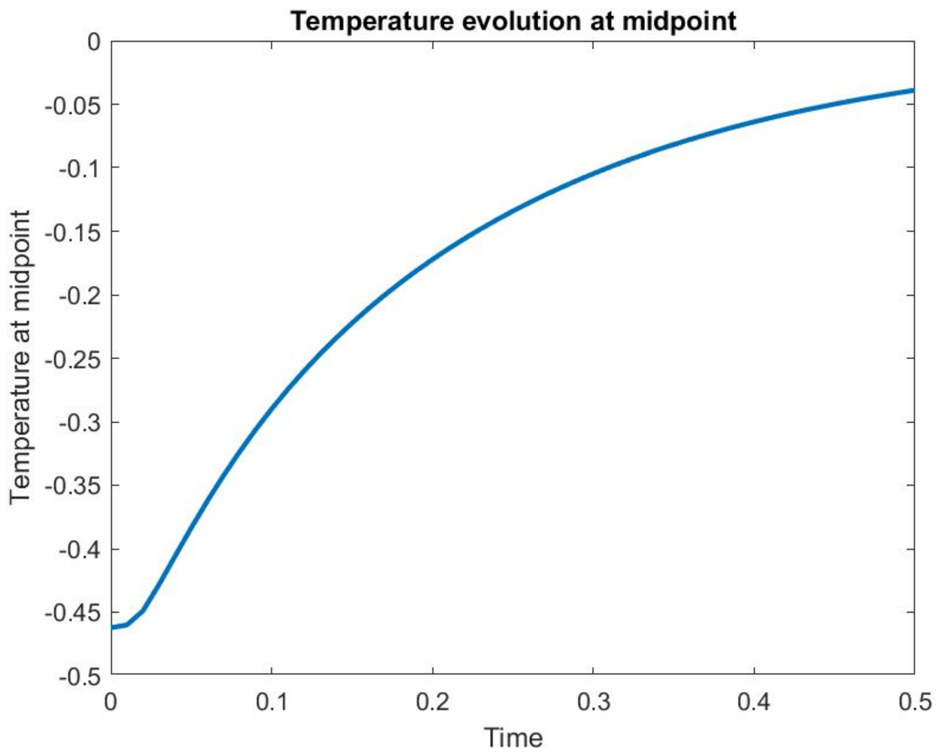
1. Define the PDE problem type
2. Create a PDE model : $createpde();$
3. Define the geometry : $geometryFromEdges();$
4. Generate the mesh : $generateMesh();$
5. Specify PDE coefficients : $applyProperties();$
6. Apply boundary conditions : $applyBoundaryCondition();$
7. Set initial conditions (for time-dependent PDEs) : $applyInitialCondition();$
8. Define the time steps : $time = i{:}sl{:}f;$
9. Solve the PDE : $solution = solve();$
10. Visualize results : $pdeplot();$

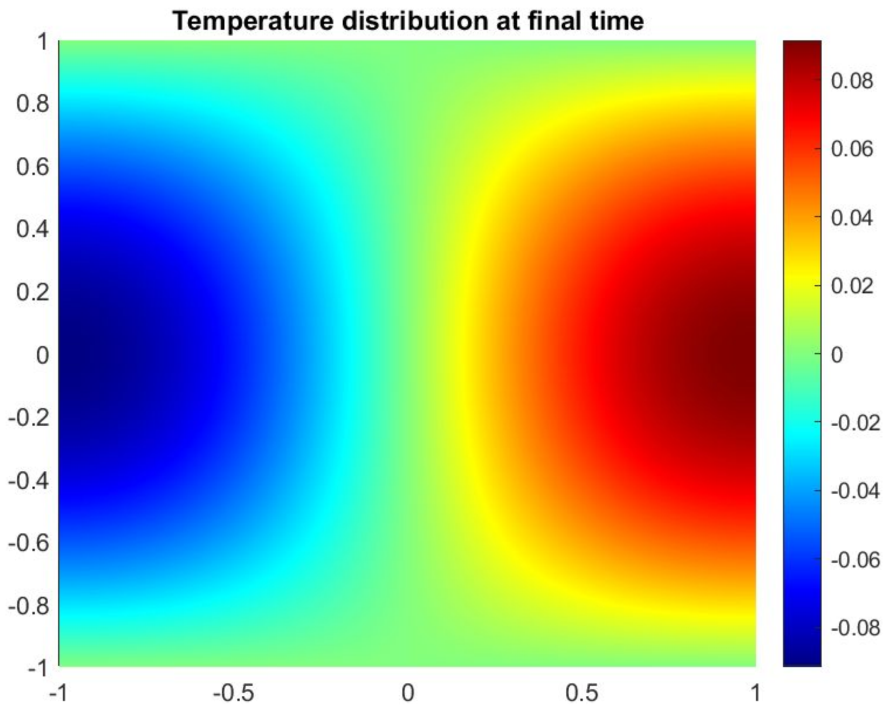
- Example: Solve the given heat equation in MATLAB and plot the heat diffusion process using PDE Toolbox $\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, 0 < x < 1, t > 0, u(x,0) = \sin(\pi x), u(0,t) = 0, u(1,t) = 0$
- Solution: clc;clear;

```
% Heat Equation using PDE Toolbox
model = createpde('thermal','transient');
% Geometry: 1D interval [0,1]
geometryFromEdges(model,@squareg);   % squareg is unit square geometry
model.Geometry.Edges;                % edges of geometry
% Assign thermal properties
thermalProperties(model,'ThermalConductivity',1,'MassDensity',1,'SpecificHeat',1);
% Apply boundary conditions: u=0 at both ends
thermalBC(model,'Edge',1,'Temperature',0); % left boundary
thermalBC(model,'Edge',3,'Temperature',0); % right boundary
```

```
% Initial condition: u(x,0) = sin(pi*x)
thermalIC(model,@(location) sin(pi*location.x));
% Generate mesh
generateMesh(model,'Hmax',0.05);
% Time stepping
tlist = 0:0.01:0.5;  % time vector
% Solve PDE
result = solve(model,tlist)
% Extract solution
u = result.Temperature
% Plot temperature distribution at different times
figure
pdeplot(model,'XYData',u(:,end),'ColorMap','jet');
title('Temperature distribution at final time');
```

```
% Plot evolution at mid-point
midNode = round(size(u,1)/2);
figure
plot(tlist,u(midNode,:),'LineWidth',2);
xlabel('Time'); ylabel('Temperature at midpoint');
title('Temperature evolution at midpoint');
```





APPLICATIONS

- **Applications of PDEs :**
  1. **Mechanical Engineering**: stress analysis, vibrations, fluid flow, heat conduction
  2. **Electrical Engineering**: electromagnetics, wave propagation
  3. **Civil Engineering**: structural analysis, soil mechanics
  4. **Aerospace**: aerodynamics, thermal analysis in aircraft/drone design

Vibrations in MATLAB

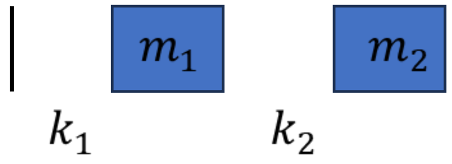
- Vibrations are oscillations of a system about an equilibrium position. In mechanical engineering, vibration analysis is critical for machines, vehicles, and structures.
- **Types of Vibrations :**
- **Free vibration** – System oscillates with its natural frequency (no external force).
  Example: Mass-spring-damper system when released from rest.
- **Forced vibration** – System is subjected to an external periodic force.
  Example: A car suspension under road disturbances.
- **Damped vibration** – Oscillations decrease with time due to energy loss (friction, air resistance).
- Example: Mathematical model for a single degree of freedom (SDOF) system:

$$m\ddot{x} + c\dot{x} + kx = F(t)$$

Where: m: mass, c: damping coefficient, k: stiffness, $F(t)$: external force

- Example :Solve the 2-DOF spring–mass system:
  - Masses: $m1 = 1\,kg,\ m2 = 1\,kg$
  - Stiffnesses: $k1 = 2000\,N/m,\ k2 = 1000\,N/m$
  - Equations of motion: $M\ddot{x} + Kx = 0$
  - Where: M=$\begin{pmatrix} m_1 & 0 \\ 0 & m_2 \end{pmatrix}$, $K = \begin{pmatrix} k_1+k_2 & -k_2 \\ -k_2 & k_2 \end{pmatrix}$
  - We solve for natural frequencies, mode shapes, and response with initial condition:

    $x(0) = \begin{pmatrix} 0.01 \\ 0 \end{pmatrix}, \dot{x}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$



- Solution:

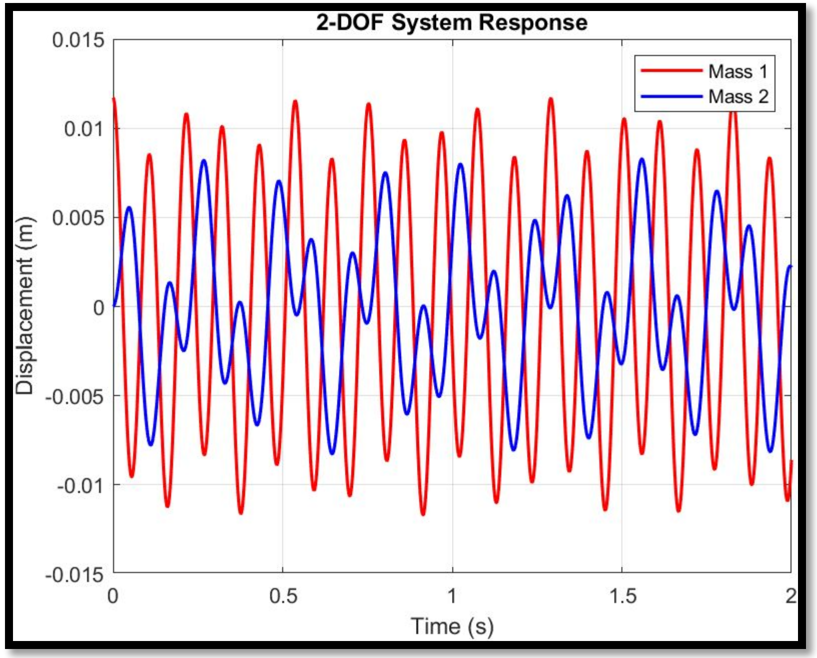
```
clc; clear; close all;
% Parameters
m1 = 1; m2 = 1;
k1 = 2000; k2 = 1000;
M = [m1 0; 0 m2];
K = [k1+k2 -k2; -k2 k2];
% Eigenvalue problem
[phi, D] = eig(K, M);
wn = sqrt(diag(D));   % natural frequencies (rad/s)
```

```
% Normalize mode shapes
phi = phi ./ max(abs(phi));
disp('Natural frequencies (Hz):')
disp(wn/(2*pi))
disp('Mode shapes:')
disp(phi)
% Initial conditions
x0 = [0.01; 0];   % initial displacement
xd0 = [0; 0];     % initial velocity
% Modal coordinates
q0 = phi' * M * x0;
qd0 = phi' * M * xd0;
% Time vector
t = linspace(0, 2, 1000);
% Compute response
x = zeros(2, length(t));
```

```
for i = 1:length(wn)
    x = x + phi(:,i) * ( q0(i)*cos(wn(i)*t) + (qd0(i)/wn(i))*sin(wn(i)*t) );
end
% Plot response
figure;
plot(t, x(1,:), 'r', 'LineWidth', 1.5); hold on;
plot(t, x(2,:), 'b', 'LineWidth', 1.5);
xlabel('Time (s)'); ylabel('Displacement (m)');
legend('Mass 1', 'Mass 2');
title('2-DOF System Response');
grid on;
```

- Ans:

```
Natural frequencies (Hz): 3.8520
                          9.2996
Mode shapes: -0.4142   -1.0000
             -1.0000    0.4142
```



Optimization in MATLAB

- Optimization is the process of finding the **best solution** under given constraints, usually by minimizing or maximizing an objective function.
- **Types of Optimization Problems :**

1. **Unconstrained Optimization**

$$\min f(x)$$

Example: $Minimize\ f(x) = x^2 + 4x + 5$

2. **Constrained Optimization**

$$\min f(x)\ \ subject\ to\ g(x) \leq 0,\ h(x) = 0$$

Example: Minimize cost while satisfying design limits.

MATLAB Functions for Optimization

Problem Type	MATLAB Command
Unconstrained Optimization	
Constrained Optimization	
Linear Programming	
Quadratic Programming	
Nonlinear Least Squares	
Global Optimization	
Multiobjective Optimization	
Integer/Mixed Integer	

MATLAB Implementation of Optimization

- **Example 1: Unconstrained Optimization**

  Minimize function $f(x) = x^2 + 4x + 5$

- Solution:

```
f = @(x) x.^2 + 4*x + 5;
x0 = 2;   % initial guess
[x_opt,fval] = fminunc(f,x0);
disp(['Optimal x = ', num2str(x_opt)])            Ans: -2
disp(['Minimum value = ', num2str(fval)])          Ans:  1
```

- **Example 2: Constrained Optimization**

  Minimize $f(x) = x_1^2 + x_2^2$
  Subject to: $x_1 + x_2 \geq 1, x_1, x_2 \geq 0$

- Solution:

```
f = @(x) x(1)^2 + x(2)^2;   % Objective function
x0 = [0.5, 0.5];             % Initial guess
A = [-1 -1];   % Inequality constraint (x1 + x2 >= 1)
```

- ```
  b = -1;
  lb = [0 0];   % Lower bounds
  [x_opt, fval] = fmincon(f,x0,A,b,[],[],lb);
  disp(['Optimal x = ', num2str(x_opt)])        Ans: 0.5    0.5
  disp(['Minimum value = ', num2str(fval)])     Ans:  0.5
  ```
- **Example 3**: Find the optimum solution and minimum value of given global optimization problem $x_1^2 + x_2^2 + 10 \cdot \sin(x_1) + 5 \cdot \sin(x_2)$.
- **Solution**:

```
f = @(x) x(1)^2 + x(2)^2 + 10*sin(x(1)) + 5*sin(x(2));
[x_opt, fval] = ga(f,2);
disp(['Optimal solution = ', num2str(x_opt)])   Ans: -1.3157    -1.121
disp(['Minimum value = ', num2str(fval)])       Ans: -11.1914
```

- **Example 4**: Find the maximum of the function $f(x) = \sin(x)$ in interval $[0,2\pi]$.
- **Solution:**

```
f=@(x) sin(x);
sol=fminbnd(f,0,2*pi)
Ans:  sol= 4.7124
```
```

EXERCISE

1. Solve the heat equation $\frac{\partial u}{\partial t} = c^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$ with $c = \frac{1}{\pi}$. The boundary conditions are $u(0, y, t) = u(1, y, t) = 0, 0 < y < 1$ and $u(x, 0, t) = u(x, 1, t) = 0, 0 < x < 1$. The initial temperature distribution is $u(x, y, 0) = 100$ for $0 < \mathrm{x} < 1, 0 < \mathrm{y} < 1$.
2. Solve following heat equation $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$. Initial conditions is $u(0, x) = \frac{2x}{1+x^2}$ and boundary conditions are $u(t, 0) = 0,\ u(t, 1) = 1$.
3. Solve the following heat equation

$$
\begin{gathered}
u_t = u_{xx} + u_{yy} + \sin(t) \\
u(t, 0, y) = 0; \quad u_x(t, \pi, y) = 1 \\
u_y(t, x, 0) = 0; \quad u(t, x, 2\pi) = x \\
u(0, x, y) = 0
\end{gathered}
$$

4. Solve the 1-D wave equation $u_{tt} = c^2 u_{xx}, x \epsilon (0,1), t > 0$. Boundary Conditions $u(0, t) = u(1, t) = 0$. Initial Conditions are $u(x, 0) = \sin(\pi x)\,, u_t(x, 0) = 0$.

5. Solve the optimization problem, a cantilever beam vibrates with natural frequency: $\omega_n = \sqrt{\frac{3EI}{mL^3}}$

E = modulus of elasticity, I = moment of inertia ($I = \frac{bh^3}{12}$)

m = mass per unit length, L = beam length

Find beam dimensions b, h that maximize natural frequency (less vibration, more stable).

Constraint: maximum material volume $V = b \cdot h \cdot L \leq V_{max}$

6. Solve the optimization problem, equation of motion: $m\ddot{x} + c\dot{x} + kx = F_0 sin(\omega t)$, steady-state amplitude: $X = \frac{F_0}{\sqrt{(k-m\omega^2)^2+(c\omega)^2}}$. Choose damping coefficient c that minimizes vibration amplitude X. Constraint: $0 < c < c_{max}$.

7. Find the minimum of function $f\,(x) = x^2 - 2sinx$ in the interval from $x1 = 0\ to\ x2 = 4$.

8. As electric current moves through a wire, heat generated by resistance is conducted through a layer of insulation and then convected to the surrounding air. The steady-state temperature of the wire can be computed as

$$T = T_{air} + \frac{q}{2\pi}\left[\frac{1}{k}\ln\left(\frac{r_w + r_i}{r_w}\right) + \frac{1}{h}\left(\frac{1}{r_w + r_i}\right)\right]$$

Determine the thickness of insulation $r_i(m)$ that minimizes the wire's temperature given the following parameters: $q = \text{heat generation rate} = 75\, W/m, r_w = \text{wire radius} = 6\, mm, k = \text{thermal conductivity of insulation} = 0.17\, W/(m\, K), h = \text{convective heat transfer coefficient} = 12\, W/(m^2\, K), \text{and } Tair = \text{air temperature} = 293\, K.$

THANK YOU!