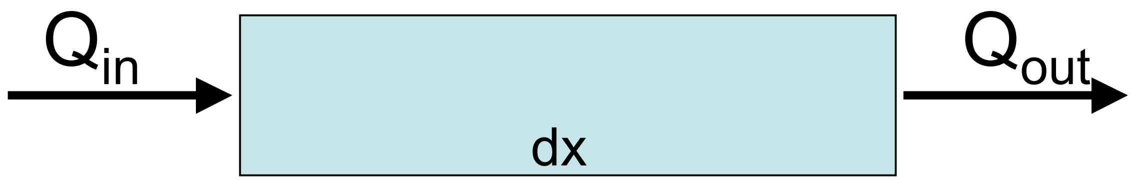
Components of Scientific Programming

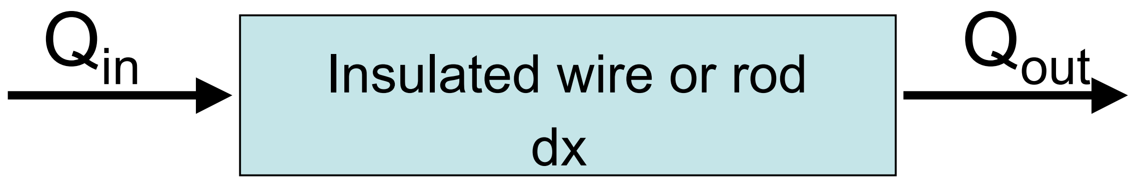
- Definition of problem
- Physical/mathematical formulation (Focus today)
- Development of computer code (Focus to date)
  - Development of logic (e.g., flowchart)
  - Assembly of correct lines of code
  - Testing and troubleshooting
  - Visualization of results
  - Optimization of code
- Analysis of results
- Synthesis

Steady-State
Heat Conduction Equation



In this exercise, no major Matlab concepts need to be implemented.
The focus is on a small change in the coding and mathematics as a result of the physics extending from 1-D to 2-D.

Steady-State Heat Conduction Equation (1-D)



Change in thermal energy in time = Heat flow out - Heat-flow in

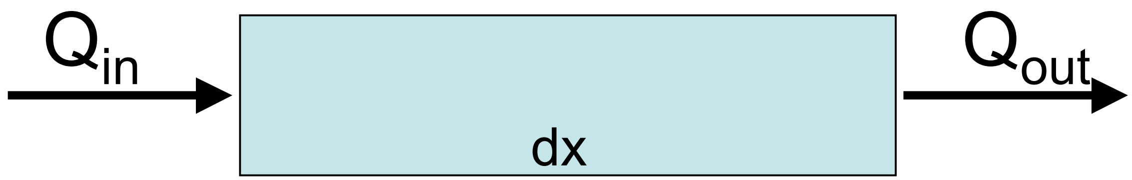
$$\frac{\Delta E}{\Delta t} = E^* = Q_{out} - Q_{in} = -c\left(\frac{dT_{out}}{dx} - \frac{dT_{in}}{dx}\right)$$

T = temperature, and x = position
We will represent dT/dx as T'.

At steady state, the rate of energy change (E*) is zero:

$$E^* = 0 = -c\left(T'_{out} - T'_{in}\right)$$

Steady-State Heat Conduction Equation (1-D)



At steady state,the rate of energy change (E*) is zero everywhere, so E* does not change as a function of position (x):

$$\frac{dE*}{dx} = 0 = -c\frac{(T'_{out} - T'_{in})}{dx} = -cT''$$

$$T'' = \frac{d^2T}{dx^2} = \nabla^2 T = 0 \quad (1 - D \;\; heat \;\; flow)$$

This is the Laplace equation,
one of the most important equations in physics

Steady-State Heat Conduction Equation

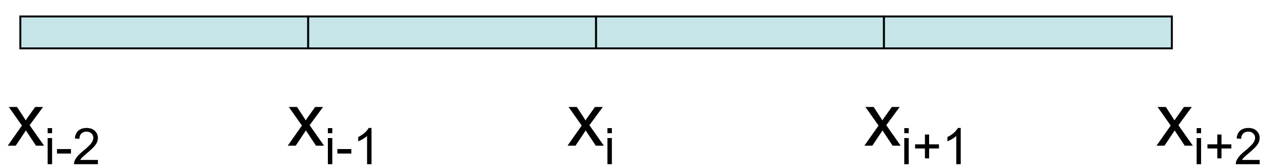
$$\nabla^2 T = \frac{d^2 T}{dx^2} = 0 \quad (1-D \;\; steady\, state \;\; heat \;\; flow)$$

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \quad (2-D\, state \;\; heat \;\; heat \;\; flow)$$

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = 0 \quad (3-D\, state \;\; heat \;\; heat \;\; flow)$$

What does T" = 0 mean?

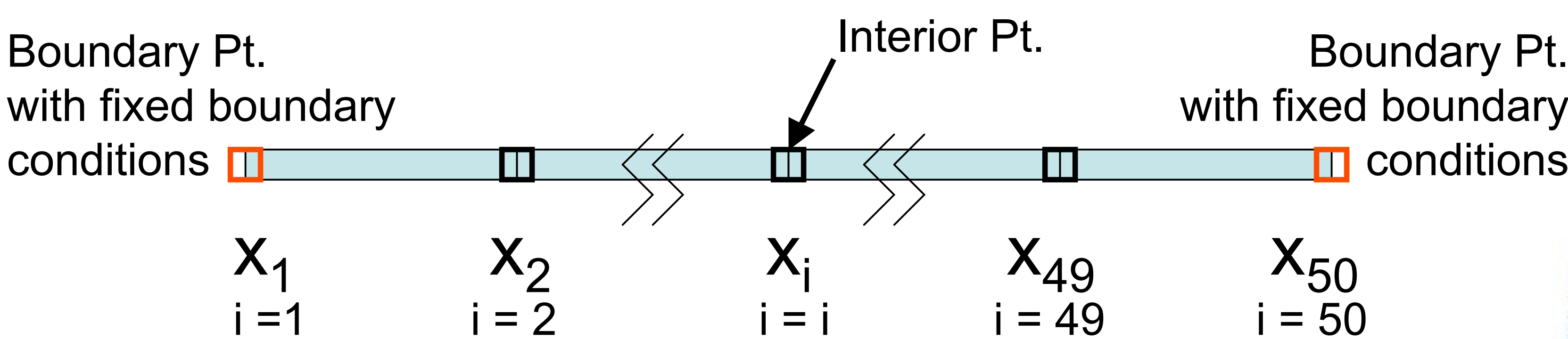
- T' = constant
- A plot of T vs. x must be a line.
- T must vary linearly between any two points along the rod at steady state.
- If T is known at positions x_{i-1} and x_{i+1}, then T(i) = [T(i-1) + T(i+1)]/2
- T(i) = average of Tat the nearby equidistant points (see Appendix for more detail)



1-D Steady State Heat Conduction (a)

$$T(x_i) \approx \frac{T(x_{i+1}) + T(x_{i-1})}{2}$$

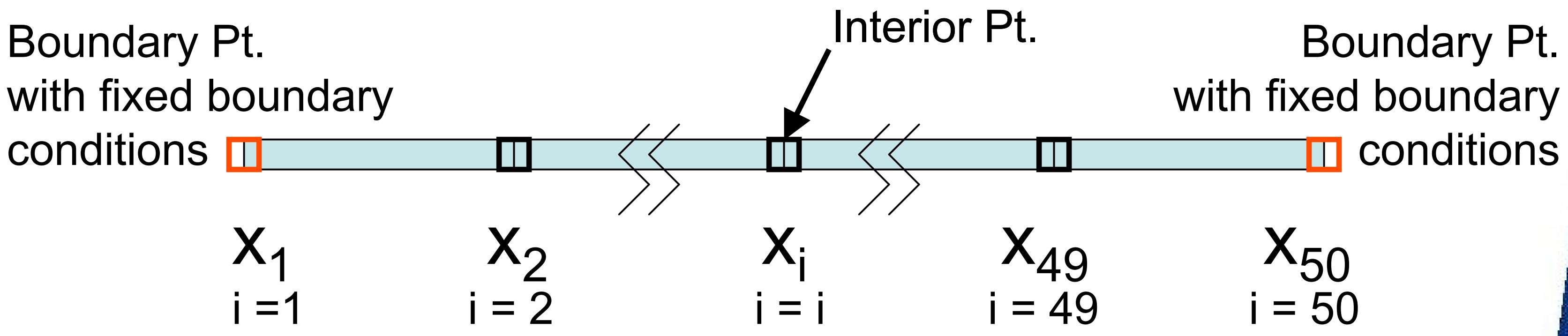
```
num_iterations = 500;
x = 1:50;
T = 10.*rand(size(x));          % initial temperature distribution
n = length(x);
for j = 1:num_iterations        % a "for loop" is used here
   for i=2:n-1;          % Don't change T at the ends of the rod!
            T(i) = (T(i+1) + T(i-1))./2;
   end
figure(1); clf; plot(x,T); axis([0 n 0 10]); xlabel('x'); ylabel('T');
end
```



1-D Steady State Heat Conduction (b)

$$T(x_i) \approx \frac{T(x_{i+1}) + T(x_{i-1})}{2}$$

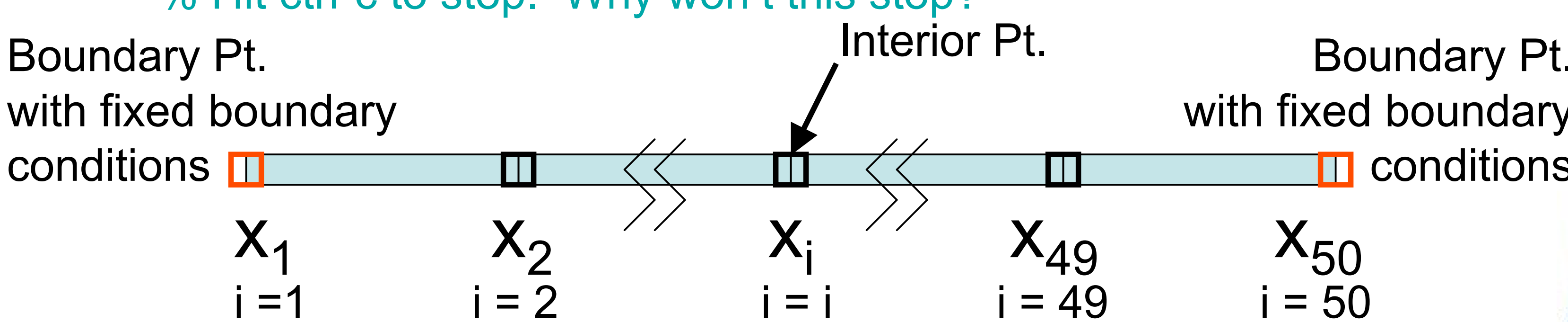
```
num_iterations = 500;
x = 1:50;
T = 10.*rand(size(x));         % initial temperature distribution
n = length(x);
for j = 1:num_iterations       % a "for loop" is used here
  T(2:n-1) = (T (3:n) + T(1:n-2))./2;
  figure(1); clf; plot(x,T); axis([0 n 0 10]); xlabel('x'); ylabel('T');
end
% Shorter, and it runs, but it introduces “noise”
```



1-D Steady State Heat Conduction (c)

$$T(x_i) \approx \frac{T(x_{i+1}) + T(x_{i-1})}{2}$$

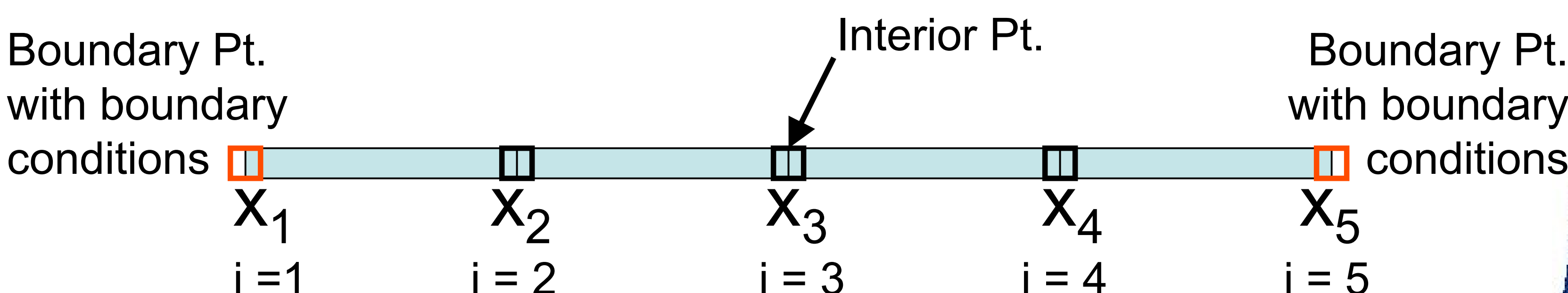
```
x = 1:50;
T = 10.*rand(size(x));        % initial temperature distribution
n = length(x);
tol = 0.1;
dT = tol.*2.*ones(size(T)); % Initialize dT
while max(abs(dT)) > tol     % a "while loop" is used here
  for i=2:n-1 ;
        dT(i) = ((T(i+1) + T(i-1))./2) -T(i); % Change in T
        T(i) = T(i) + dT(i); % New T = old T + change in T
  end
figure(1); clf; plot(x,T); axis([0 n 0 10]);figure(2) ; plot(x,dT);
end
% Hit ctrl-c to stop.  Why won't this stop?
```



1-D Steady State Heat Conduction (d)

$$T(x_i) \approx \frac{T(x_{i+1}) + T(x_{i-1})}{2}$$

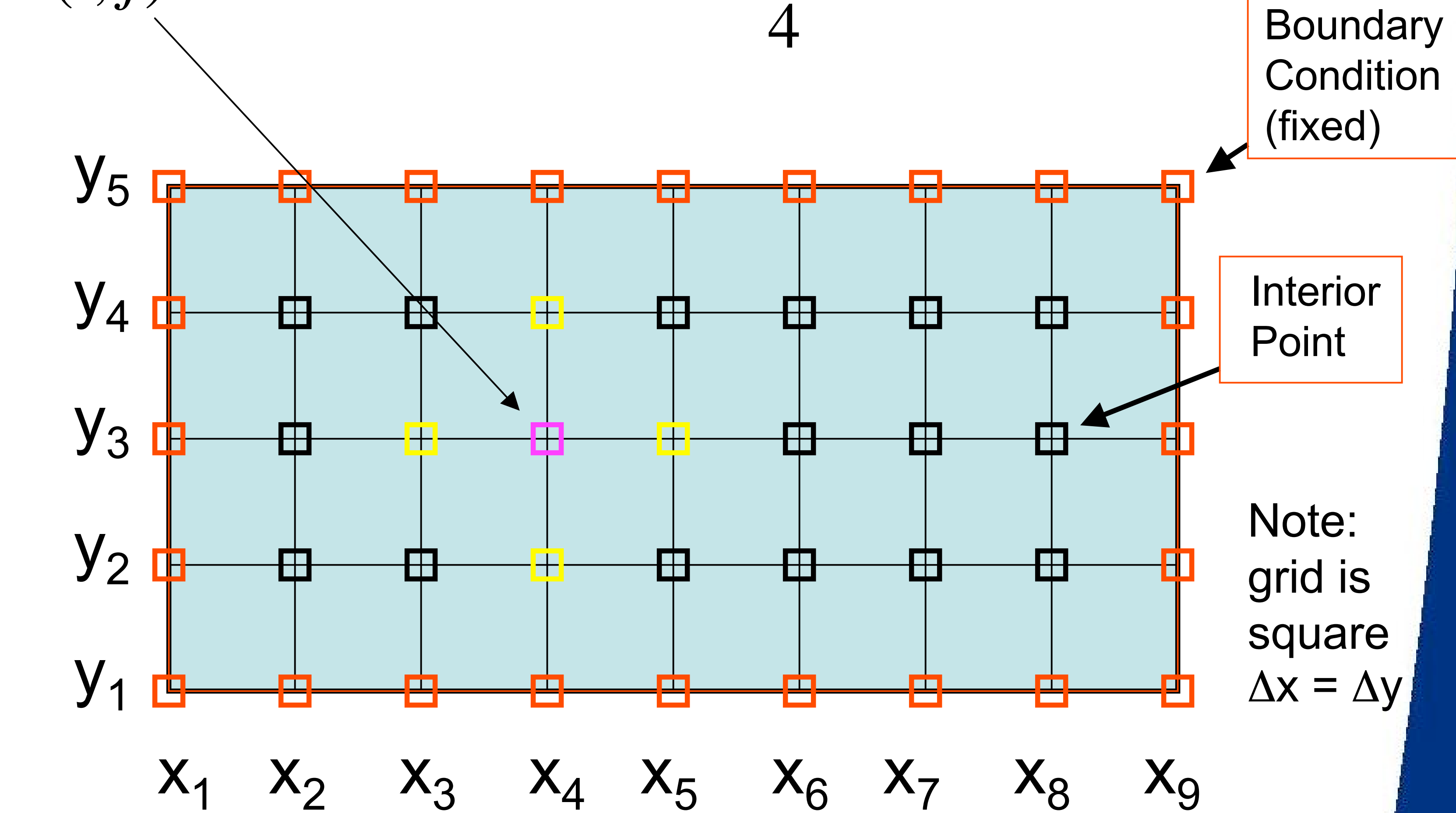
```
x = 1:50;
T = 10.*rand(size(x));        % intial temperature distribution
n = length(x);
tol = 0.0001;
dT = tol.*2.*ones(size(T)); % Initialize dT
while max(abs(dT(2:n-1))) > tol        % a "while loop" is used here
  for i=2:n-1 ;
          dT(i) = ((T(i+1) + T(i-1))./2) -T(i); % Change in T
          T(i) = T(i) + dT(i); % New T = old T + change in T
  end
figure(1); clf; plot(x,T); axis([0 n 0 10]); xlabel('x'); ylabel('T');
end
```



2-D Steady State Heat Conduction Equation

For a 2-D problem, the temperature at a point is the average of T at the four nearest neighbors

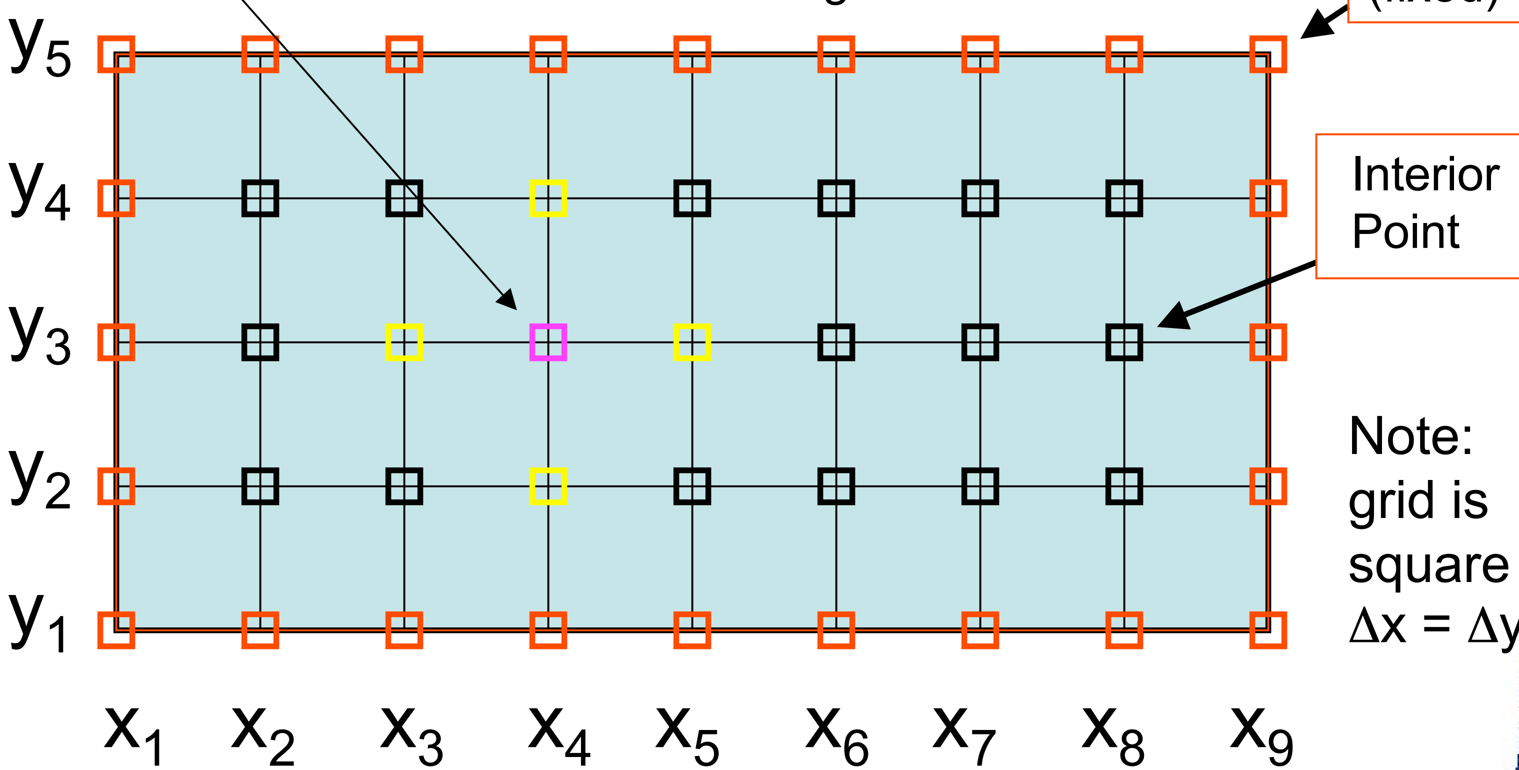
$$T(i,j) \approx \frac{T(i-1,j)+T(i+1,j)+T(i,j+1)+T(i,j-1)}{4}$$



2-D Steady State Heat Conduction Equation

$$T(i,j) \approx \frac{T(i-1,j)+T(i+1,j)+T(i,j+1)+T(i,j-1)}{4}$$

*Add a loop to account for 2-D grid
*Reformulate $T(x_i)$ to account for 2-D
*Check for convergence



Note: grid is square $\Delta x = \Delta y$

Appendix

- This appendix shows in more detail how the second derivative d^2T/dx^2 is evaluated numerically using a finite difference approximation method.

Solution of 1-D Steady State Heat Conduction Equation

$$\nabla^2 T = \frac{d^2T}{dx^2} = T'' = \lim_{\Delta x \to 0} \frac{T'\left(x + \frac{\Delta x}{2}\right) - T'\left(x - \frac{\Delta x}{2}\right)}{\Delta x} = 0$$

$$\nabla^2 T = \lim_{\Delta x \to 0} \frac{\left(\frac{T(x+\Delta x) - T(x)}{\Delta x}\right) - \left(\frac{T(x) - T(x-\Delta x)}{\Delta x}\right)}{\Delta x} = 0$$

$$\nabla^2 T = \lim_{\Delta x \to 0} \frac{[T(x+\Delta x) - T(x)] - [T(x) - T(x-\Delta x)]}{(\Delta x)^2} = 0$$

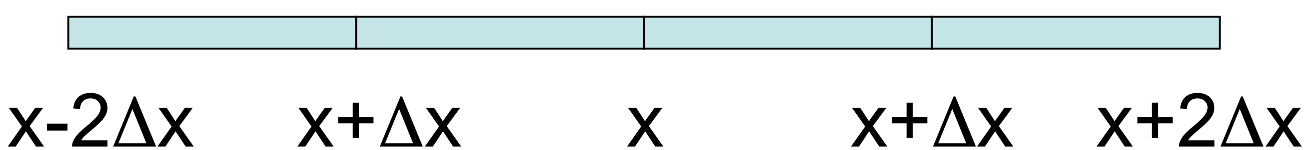
1-D Steady State Heat Conduction Equation

$$\nabla^2 T = \lim_{\Delta x \to 0} \frac{[T(x+\Delta x) - T(x)] - [T(x) - T(x-\Delta x)]}{(\Delta x)^2} = 0$$

$$[T(x+\Delta x) - T(x)] - [T(x) - T(x-\Delta x)] \approx 0$$

$$T(x+\Delta x) + T(x-\Delta x) - 2T(x) \approx 0$$

$$-2T(x) \approx -[T(x+\Delta x) + T(x-\Delta x)]$$



1-D Steady State Heat Conduction Equation

$$-2T(x) \approx -[T(x+\Delta x) + T(x-\Delta x)]$$

$$T(x) \approx \frac{T(x+\Delta x) + T(x-\Delta x)}{2}$$

$$T(x_i) \approx \frac{T(x_{i+1}) + T(x_{i-1})}{2}$$

In **words**, what does this mean?