Lecture 2 - The Finite Volume Method

Sections 4.1-4.3 (Versteeg)

ME 2256/MEMS 1256 - Applications of Computational Heat and Mass Transfer

Mechanical Engineering and Materials Science Department
University of Pittsburgh

Student Learning Objectives

At the end of the lecture, students should be able to:

- Understand the formulation of the 1D diffusion equation;
- Handle diffusivity at the interface of two materials;
- Code a 1D steady-state diffusion problem considering spatially-dependent and temperature-dependent for one and two-domain systems.

Transport Equation

- Recall the transport equation in vector form:

$$\underbrace{\frac{\partial}{\partial t}(\rho\phi)}_{\text{Accumulation}} + \underbrace{\nabla\cdot(\rho\vec{V}\phi)}_{\text{Advection}} = \underbrace{\nabla\cdot(\Gamma\nabla\phi)}_{\text{Diffusion}} + \underbrace{S}_{\text{Generation}}$$

- $\phi \rightarrow$ conserved quantity;
- $\Gamma \rightarrow$ diffusion coefficient;
- $\rho \rightarrow$ density;
- We are interested in solving the diffusion term first.

Diffusion Equation

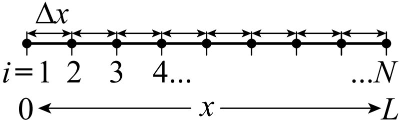
- The steady-state diffusion equation is expressed as:

$$\nabla \cdot (\Gamma \nabla \phi) = 0$$

- Let's take Γ as the thermal conductivity of the material (can be temperature dependent), and ϕ as temperature, T.
- There are two methods to solve this numerically:
  1. Finite Difference Method (FDM);
  2. Finite Volume Method (FVM).

Finite Difference Method

- We first must replace the physical, continuous domain of the problem with a mesh/grid.
- For instance, in 1D, if our domain is bounded between 0 and L, we would construct a grid of N points, separated by a distance Δx:



- Note, for FDM, it is preferable to have constant Δx, otherwise, the expressions for the derivatives becomes cumbersome using Lagrange polynomials.

1D Diffusion Equation

- The diffusion equation, in 1D, is expressed as:

$$\frac{\partial}{\partial x}\left(\lambda(T)\frac{\partial T}{\partial x}\right) = 0$$

- If $\lambda = $ c:

$$\lambda\frac{\partial^2 T}{\partial x^2} = 0$$

- If $\lambda = f(T)$ (see slide 41):

$$\lambda(T)\frac{\partial^2 T}{\partial x^2} + \frac{\partial T}{\partial x}\frac{\partial \lambda(T)}{\partial x} = 0$$

- Either of these can be solved via the FDM—need to decide on a scheme.

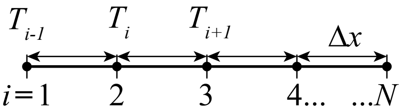
1D Diffusion Equation $\lambda = \text{c} - \text{FDM}$

- Consider the following:

$$\lambda \frac{\partial^2 T}{\partial x^2} = 0$$

- The second derivative can be approximated using a second order central difference scheme:

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} + O(\Delta x^2) = 0$$



1D Diffusion Equation $\lambda = c$ – FDM

- Rearranging for T_i:

$$T_i = \frac{T_{i+1} + T_{i-1}}{2}$$

- This is valid in for $2 \leq i \leq N-1$, where $i = 1$ and $i = N$ are the boundary conditions.
- We see for a constant diffusion coefficient, and constant-temperature boundary conditions (Dirichlet), the steady-solution is trivial.

```matlab
% Number dx:
N = 10;

% Spatial domain definition:
L = 10;
x = linspace(0,L,N);
```

1D Diffusion Equation $\lambda = c$ – FDM

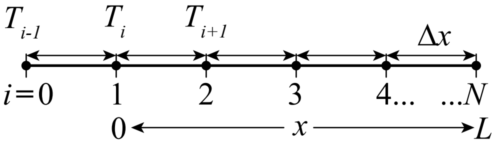
```matlab
% Temperature array pre-allocation:
T_old = zeros(1,N); T_new = zeros(1,N);
% Defining relative error and loop counter:
error = 1; counter = 0;
% Solving for interior temperatures:
while error >= 1e-5
    % Boundary conditions
    T_old(1) = 0; T_new(end) = 100;
    for i = 2:N-1
        T_new(i) = (T_old(i+1) + T_old(i-1))/2;
    end
    % Calculating relative error
    error = max(T_new - T_old);
    % Resetting temperature and updating counter
    T_old = T_new; counter = counter + 1;
end
```

Check out Github for FDM_dirichlet.m

1D Diffusion Equation $\lambda = c$ – FDM

- Now what happens if we have a Neumann boundary condition on the LHS?
- We can do a central difference expansion about $i = 1$:

$$q'' = -\lambda A \frac{\partial T}{\partial x} = -\lambda A \frac{T_2 - T_0}{2\Delta x}$$



- Solving for T_0, i.e. the point on the boundary (ghost point or fictitious point):

$$T_0 = T_2 + \frac{2\Delta x q''}{\lambda A}$$

1D Diffusion Equation $\lambda = c$ – FDM

```matlab
% Spatial domain definition, error and counter:
N = 10; L = 10; x = linspace(0,L,N); dx = L/(N-1);
lambda = 1; A = 1; error = 1; counter = 0;
% Temperature array pre-allocation:
T_old = zeros(1,N); T_new = zeros(1,N);
while error >= 1e-5
    q_dp = 100; T_new(end) = 100;
    T_0 = T_old(2) + (2*dx*q_dp)/(lambda*A);
    for i = 1:N-1
        if i == 1
            T_new(i) = (T_old(i+1) + T_0)/2;
        else
            T_new(i) = (T_old(i+1) + T_old(i-1))/2;
        end
    end
    error=max(T_new-T_old); T_old=T_new; counter=counter+1;
end
```

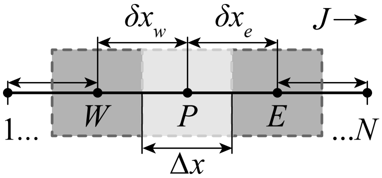
Check out Github for FDM_mixed.m

1D Diffusion Equation $\lambda = c$ – FDM

- How can we check?
$$\frac{d^2T}{dx^2} = 0 \implies \frac{dT}{dx} = c_1 \implies T(x) = c_1 x + c_2$$
- Apply our temperature boundary condition:
$$T(x = L) = 300 \implies c_1 L + c_2 = 300$$
- From Fourier's law of conduction:
$$q'' = -\lambda \frac{dT}{dx} \implies \left.\frac{dT}{dx}\right|_{x=0} = c_1 = -\frac{q''}{\lambda}$$
- Recall:
$$c_1 L + c_2 = 300 \implies c_2 = 300 - c_1 L = 300 + \frac{q''}{\lambda}$$
- Therefore:
$$T(x) = -\frac{q'' x}{\lambda} + 300 + \frac{q''}{\lambda} = 300 + \frac{q''}{\lambda}(L - x)$$
- The analytic solution is plotted against our FDM scheme in FDM_mixed.m, and shows excellent agreement (after 200 iterations).

1D Diffusion Equation – FVM

- As opposed to the FDM, we can look at what is happening within three discrete control volumes using a stencil:



- Recall J is our flux of a conserved variable; for a no flux condition:

$$\frac{dJ}{dx} = 0$$

- We can integrate this over the C.∀. centered at P, recognizing $J_e = J_w$.

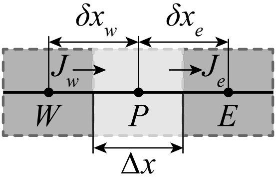
1D Diffusion Equation – FVM

- Integrating the diffusion term:

$$\int_{\text{C.}\forall\text{.}} \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right) = 0 \implies A_e\left(\Gamma \frac{\partial \phi}{\partial x}\right)\bigg|_e = A_w\left(\Gamma \frac{\partial \phi}{\partial x}\right)\bigg|_w$$

- Recalling from FDM, using a backward difference FD scheme:

$$\frac{\partial \phi}{\partial x} = \frac{\phi_i - \phi_{i-1}}{\Delta x}$$



- In terms of cell-center values:

$$A_e\Gamma_e\left(\frac{\phi_E - \phi_P}{\delta x_e}\right) = A_w\Gamma_w\left(\frac{\phi_P - \phi_W}{\delta x_w}\right)$$

1D Diffusion Equation – FVM

- Expanding the expression for the backward difference of ϕ:

$$A_e \frac{\Gamma_e \phi_E}{\delta x_e} - A_e \frac{\Gamma_e \phi_P}{\delta x_e} = A_w \frac{\Gamma_w \phi_P}{\delta x_w} - A_w \frac{\Gamma_w \phi_W}{\delta x_w}$$

- Combining like ϕ:

$$\phi_P \left(A_w \frac{\Gamma_w}{\delta x_w} + A_e \frac{\Gamma_e}{\delta x_e} \right) = \phi_E \left(A_e \frac{\Gamma_e}{\delta x_e} \right) + \phi_W \left(A_w \frac{\Gamma_w}{\delta x_w} \right)$$

- We can specify coefficients to simplify the equation:

$$a_P = A_w \frac{\Gamma_w}{\delta x_w} + A_e \frac{\Gamma_e}{\delta x_e}; \quad a_W = A_w \frac{\Gamma_w}{\delta x_w}; \quad a_E = A_e \frac{\Gamma_e}{\delta x_e}$$

- Thus, the equation becomes:

$$a_P \phi_P = a_W \phi_W + a_E \phi_E = \sum a_{nb} \phi_{nb}$$

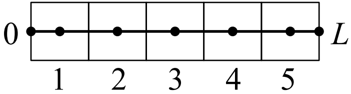
$$\text{with } a_P = a_W + a_E$$

Example #1

- Let us consider 1D conduction with the following:

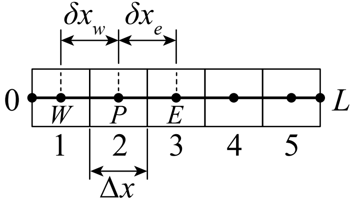
$$
\begin{aligned}
L &= 1\,[\mathrm{m}] \\
\Gamma = \lambda &= 1\,[\mathrm{W/m\text{-}K}] \\
\phi(0) = T(0) &= 100\,\mathrm{K} \\
\phi(L) = T(L) &= 200\,\mathrm{K} \\
A_e = A_w = A &= 1\,[\mathrm{m}^2]
\end{aligned}
$$

- This is schematically shown as:



Example #1

- Denoting distances:



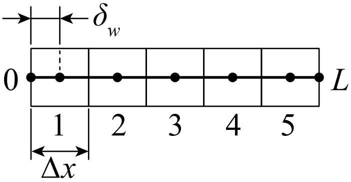
- Starting with the interior C.∀.s (2, 3 and 4):

$$\text{C.}\forall.\,2: \; a_P(2)T_2 - a_E(2)T_3 - a_W(2)T_1 = 0$$

$$\text{C.}\forall.\,3: \; a_P(3)T_3 - a_E(3)T_4 - a_W(3)T_2 = 0$$

$$\text{C.}\forall.\,4: \; a_P(4)T_4 - a_E(4)T_5 - a_W(4)T_3 = 0$$

Example #1



- At C.∀.1:

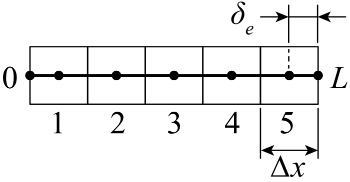
$$\text{C.}\forall.1: \; a_P(1)T_1 - a_E(1)T_2 - a_W(1)T_0 = 0$$

- T_0 is the temperature at the left boundary, but what is $a_W(1)$?

$$a_W = A_w \frac{\Gamma_w}{\delta x_w} = A \frac{\lambda_w}{\delta x_w} = A \frac{\lambda_w}{\left(\dfrac{\Delta x}{2}\right)}$$

$$\implies a_W = A \frac{2\lambda_w}{\Delta x}$$

Example #1



- At C.∀.5:

$$\text{C.}\forall.\,5:\; a_P(5)T_5 - a_E(5)T_6 - a_W(5)T_4 = 0$$

- T_6 is the temperature at the right boundary, but what is $a_E(5)$?

$$a_E = A_e \frac{\Gamma_e}{\delta x_e} = A\frac{\lambda_e}{\delta x_e} = A\frac{\lambda_e}{\left(\dfrac{\Delta x}{2}\right)}$$

$$\implies a_E = A\frac{2\lambda_E}{\Delta x}$$

Example #1

- We have the following system of equations:

$$
\begin{aligned}
\text{C.}\forall.\,1: \; & a_P(1)T_1 - a_E(1)T_2 - a_W(1)T_0 = 0 \\
\text{C.}\forall.\,2: \; & a_P(2)T_2 - a_E(2)T_3 - a_W(2)T_1 = 0 \\
\text{C.}\forall.\,3: \; & a_P(3)T_3 - a_E(3)T_4 - a_W(3)T_2 = 0 \\
\text{C.}\forall.\,4: \; & a_P(4)T_4 - a_E(4)T_5 - a_W(4)T_3 = 0 \\
\text{C.}\forall.\,5: \; & a_P(5)T_5 - a_E(5)T_6 - a_W(5)T_4 = 0
\end{aligned}
$$

- Assembly the system of equations, in matrix form:

$$
\begin{bmatrix}
a_P(1) & -a_E(1) & 0 & 0 & 0 \\
-a_W(2) & a_P(2) & -a_E(2) & 0 & 0 \\
0 & -a_W(3) & a_P(3) & -a_E(3) & 0 \\
0 & 0 & -a_W(4) & a_P(4) & -a_E(4) \\
0 & 0 & 0 & -a_W(5) & a_P(5)
\end{bmatrix}
\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix}
=
\begin{bmatrix} a_W(1)T_0 \\ 0 \\ 0 \\ 0 \\ a_E(5)T_6 \end{bmatrix}
$$

- This can be solved via inversion, fixed-point iteration, row-reduction, etc., topics of Lecture 4.

Check out Github for L2Ex1.m

Boundary Conditions

- Let's formulate a generalized discretization for 1D diffusion to include boundary conditions, denoted by the coefficient a_b and conserved quantity ϕ_b:

$$a_P\phi_P = a_W\phi_W + a_E\phi_E + a_b\phi_b$$

- Alternatively, if there is a flux boundary condition, denoted by q:

$$a_P\phi_P = a_W\phi_W + a_E\phi_E + a_b\phi_b + q$$

- If cell face not a boundary:

$$a_b = q = 0$$

- If boundaries are at C.∀., for a constant cell size:

$$a_b = A_b\frac{\Gamma_b}{\delta x_b} = A_b\frac{2\Gamma_b}{\Delta x}$$

Boundary Conditions

- There are two types of boundary conditions:

1. Dirichlet:

$$\phi_b = c \text{ i.e. given}$$

$$a_b \phi_b \text{ is known - move to RHS of eqn.}$$

2. Neumann:

$$+(\Gamma \nabla \phi)_B = q_{\text{b,given}}$$

$$J_b = q_{\text{b,given}} \text{ - move to RHS of eqn.}$$

Boundary Conditions

- To summarize:

$$-a_W\phi_W + a_P\phi_P - a_E\phi_E = \underbrace{a_b\phi_b}_{\text{Dirichlet}} + \underbrace{q_{b,\text{given}}A_b}_{\text{Neumann}}$$

- where:

$$a_W = A_w\frac{\Gamma_w}{\delta x_w},\ a_E = A_e\frac{\Gamma_e}{\delta x_e},\ a_b = A_b\frac{2\Gamma_b}{\Delta x}$$

$$a_P = A_w\frac{\Gamma_w}{\delta x_w} + A_e\frac{\Gamma_e}{\delta x_e} + A_b\frac{2\Gamma_b}{\Delta x} = a_W + a_E + a_b$$

- Thus, you can make a more generalized code out of "L2Ex1.m" through the implementation of a_b.

Example #2

- Let us reconsider Example #1 with the modified boundary conditions:

$$q''(0) = 100\,[\mathrm{W/m^2}]$$
$$\phi(L) = T(L) = 300\,\mathrm{K}$$

- Solve for the temperature distribution within the system. We will start by updating our boundary conditions, updating a_P, and then modify our RHS matrix. Updates only shown:

```
% Defining our boundary conditions:
q_b = 100; % [W/m^2]
% Updating the first a_P entry:
coeff(1,1) = a_E;
% Updating the RHS matrix:
b(1) = A_w*q_b;
```

Example #2

- The trickiest part is solving for the temperature on the boundary. We will use a basic finite-difference scheme after we obtain our solution:

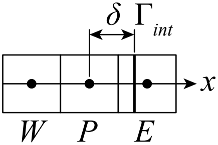
```
% Obtaining the solution through matrix inversion:
temp = coeff\b;
% Solving boundary temperature:
T(1) = temp(1) + (q_b/Gamma_w)*(delta_x_w/2);
```

- Compare the solutions obtained analytically, through FDM and through FVM. We see they are all in agreement, with the FVM being the most computationally expedient of the numerical methods.

Check out Github for L2Ex2.m

Interfacial Diffusivity

- Lastly, how do we determine the diffusivity at the interface? One method is linear interpolation (i.e. arithmetic mean):



$$\Gamma_{int} = \delta \Gamma_E + (1-\delta)\Gamma_P$$

- where:

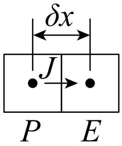
$$\delta = \frac{x - x_P}{x_E - x_P} \qquad \begin{array}{l} x = x_P,\ \delta = 0 \implies \Gamma_P \\ x = x_E,\ \delta = 1 \implies \Gamma_E \end{array}$$

- At the midpoint:

$$\delta = 0.5 \implies \Gamma_{int} = \frac{\Gamma_E + \Gamma_P}{2}$$

Interfacial Diffusivity

- Now what happens when two materials exists on opposite sides of the interface?



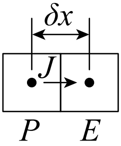
- Let's assume $\lambda_E = 0$, i.e. a perfect insulator. Using linear interpolation:

$$\lambda_{int} = \frac{\lambda_P + \lambda_E}{2} = 0.5\lambda_P$$

- This leads to flux entering said perfect insulator, which means linear interpolation is incorrect.

Interfacial Diffusivity

- Let's try to formulate an interface diffusivity on the basis of flux conservation (composite slab):



$$J = -\lambda_{int}\frac{\partial T}{\partial x} = \frac{\lambda_{int}(T_E - T_P)}{\delta x_e}$$

$$\implies J = \frac{T_E - T_P}{\dfrac{1-\delta}{\lambda_P} + \dfrac{\delta}{\lambda_E}}$$

- Let's define the denominator as the effective diffusivity:

$$\lambda_{int} = \left[\frac{1-\delta}{\lambda_P} + \frac{\delta}{\lambda_E}\right]^{-1}$$

Interfacial Diffusivity

- Assuming the interface is equidistant from both C.∀. centroids (i.e. $\delta = 0.5$):

$$\lambda_{int} = \left[\frac{0.5}{\lambda_P} + \frac{0.5}{\lambda_E}\right]^{-1}$$

$$\lambda_{int} = \frac{2\lambda_P \lambda_E}{\lambda_P + \lambda_E}$$

- Let's check for consistency:

$$\lambda_E \to 0 \text{ or } \lambda_P \to 0 \implies J = 0$$

Interfacial Diffusivity

- Considering the other extreme case of $\lambda_P \to \infty$:

$$\frac{2\lambda_P\lambda_E}{\lambda_P + \lambda_E} \to \frac{2\lambda_P\lambda_E}{\lambda_P\left(1 + \frac{\lambda_E}{\lambda_P}\right)} \to \frac{2\lambda_E}{1 + \frac{\lambda_E}{\lambda_P}}$$

- This implies that the interface diffusivity tends to:

$$\lambda_{int} = 2\lambda_E$$

- This would mean that the rate of diffusion is limited by the least diffusive material.
- Note: the harmonic average always yields a value less than that determined by the arithmetic average.

Example #3

- Repeat Example #1 with the following modified values for the thermal conductivity, which is now a function of temperature ($p-$type Bi_2Te_3):

$$\begin{aligned}\Gamma = \lambda = &(5.238086549608868e-17)T^6 + \ldots\\ &(-2.927636770231909e-13)T^5 + \ldots\\ &(5.844390241944433e-10)T^4 + \ldots\\ &(-5.642804450717544e-7)T^3 + \ldots\\ &(0.0002909446395983974)T^2 + \ldots\\ &(-0.0806341803814208 3)T + \ldots\\ &(11.00293123390308)\,[\mathrm{W/m\text{-}K}]\end{aligned}$$

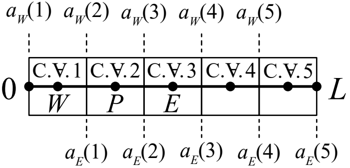
- Consider the updated geometry and B.C.s:

$$\phi(0) = T(0) = 300\,\mathrm{K} \quad \phi(L) = T(L) = 650\,\mathrm{K}$$

$$L = 0.01\,[\mathrm{m}] \quad A_e = A_w = A = 2.5e-5\,[\mathrm{m}^2]$$

Example #3

- We will use linear interpolation to determine the interfacial diffusivity. Recall that our temperature is defined at the cell-center, but we need the diffusivity at the interface. We will denote the following augmentations to our stencil:



$$\text{C.}\forall.2: \ a_P(2)T_2 - a_E(2)T_3 - a_W(2)T_1 = 0$$

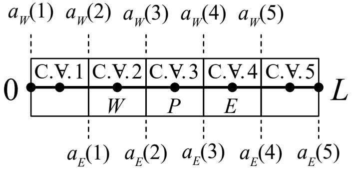
- We will define the interfacial diffusivities as:

$$a_E(2) = A_e\frac{\Gamma_{int}}{\delta x_e} = A_e\frac{\Gamma(T_2)+\Gamma(T_3)}{2\delta x_e}$$

$$a_W(2) = A_w\frac{\Gamma_{int}}{\delta x_w} = A_W\frac{\Gamma(T_1)+\Gamma(T_2)}{2\delta x_w}$$

Example #3

- Moving to the right:



$$\text{C.}\forall.3: \; a_P(3)T_3 - a_E(3)T_4 - a_W(3)T_2 = 0$$

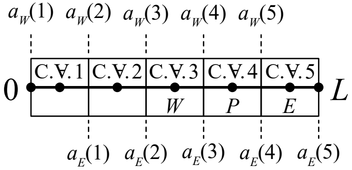
- We will define the interfacial diffusivities as:

$$a_E(3) = A_e \frac{\Gamma_{int}}{\delta x_e} = A_e \frac{\Gamma(T_3) + \Gamma(T_4)}{2\delta x_e}$$

$$a_W(3) = A_w \frac{\Gamma_{int}}{\delta x_w} = A_W \frac{\Gamma(T_2) + \Gamma(T_3)}{2\delta x_w}$$

Example #3

- Moving to the right:



$$\text{C.}\forall.\,4: \; a_P(4)T_4 - a_E(4)T_5 - a_W(4)T_3 = 0$$

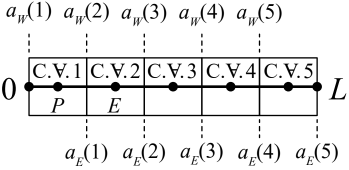
- We will define the interfacial diffusivities as:

$$a_E(4) = A_e \frac{\Gamma_{int}}{\delta x_e} = A_e \frac{\Gamma(T_4) + \Gamma(T_5)}{2\delta x_e}$$

$$a_W(4) = A_w \frac{\Gamma_{int}}{\delta x_w} = A_W \frac{\Gamma(T_3) + \Gamma(T_4)}{2\delta x_w}$$

Example #3

- At the first control volume:



$$\text{C.}\forall\text{.}\,1 : \; a_P(1)T_1 - a_E(1)T_2 - a_W(1)T_0 = 0$$

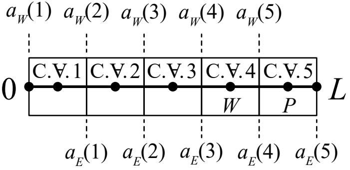
- We will define the interfacial diffusivities as:

$$a_E(1) = A_e \frac{\Gamma_{int}}{\delta x_e} = A_e \frac{\Gamma(T_1) + \Gamma(T_2)}{2\delta x_e}$$

$$a_W(1) = A_w \frac{\Gamma_w}{\left(\dfrac{\delta x_w}{2}\right)} = A_W \frac{2\Gamma(T_0)}{\delta x_w}$$

Example #3

- At the last control volume:



$$\mathrm{C.\forall.}\,5: \; a_P(5)T_5 - a_E(5)T_6 - a_W(5)T_4 = 0$$

- We will define the interfacial diffusivities as:

$$a_E(5) = A_e \frac{\Gamma_e}{\left(\frac{\delta x_e}{2}\right)} = A_e \frac{2\Gamma(T_6)}{\delta x_e}$$

$$a_W(5) = A_w \frac{\Gamma_{int}}{\delta x_w} = A_W \frac{\Gamma(T_4) + \Gamma(T_5)}{2\delta x_w}$$

Example #3

- Programmatically, we have to make minor updates to the definitions of a_E and a_W. This starts with updating the definitions of thermal diffusivity:

```matlab
% Functionalizing our thermal conductivity.  Our
input is the variable T. Units are [W/m-K]:
Gamma = @(T) (5.238086549608868e-17).*T.^6 + ...
(-2.927636770231909e-13).*T.^5 + ...
(5.844390241944433e-10).*T.^4 + ...
(-5.642804450717544e-7).*T.^3 + ...
(0.0002909446395983974).*T.^2 + ...
(-0.08063418038142083).*T + (11.00293123390308);
```

Example #3

- Defining the coefficients a_E and a_W considering temperature-dependent diffusivity:

```
% Predefining our coefficients arrays:
a_W = zeros(1,N); a_E = zeros(1,N); a_P = zeros(1,N);

% Defining our coefficients for the interior of our
domain (CVs 1:N):
for i = 1:N-1
   a_E(i) = A_e*(Gamma(T(i)) + Gamma(T(i+1)))/(2*delta_x_e);
end
a_E(end) = A_e*(2*Gamma(T(end)))/(delta_x_e);
a_W(1) = A_w*(2*Gamma(T(1)))/(delta_x_w);
for i = 2:N
   a_W(i) = A_w*(Gamma(T(i-1)) + Gamma(T(i)))/(2*delta_x_w);
end
for i = 1:N
   a_P(i) = a_E(i) + a_W(i);
end
```

Example #3

- We can proceed with populating our coefficient matrix:

```
% Populating our coefficient matrix
for i = 1:N
   for j = 1:N
     if i == j
       coeff(i,j) = a_P(i);
     end
     if i + 1 == j
       coeff(i,j) = -a_E(i);
     end
     if j + 1 == i
       coeff(i,j) = -a_W(i);
     end
   end
end
```

Example #3

- Populating the RHS matrix, obtaining a solution, calculating the relative error, and then re-initializing temperatures:

```
% Next we populate the RHS matrix
b(1) = a_W(1)*T(1);
b(end) = a_E(end)*T(end);

% Obtaining the solution through matrix inversion
temp = coeff\b;

% Defining maximum relative error
error = abs(max(temp - transpose(T(2:end-1))));

% Redefining our temperatures
T(2:end-1) = temp;
```

- Iterate until error reaches desired level!

Check out Github for L2Ex3.m

1D Diffusion Equation $\lambda = f(T) - \text{FDM}$

- Consider the following

$$\lambda(T)\frac{\partial^2 T}{\partial x^2} + \frac{\partial T}{\partial x}\frac{\partial \lambda(T)}{\partial x} = 0$$

- Using a second-order central differencing scheme:

$$\lambda_i\left(\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2}\right) + \left(\frac{T_{i+1} - T_{i-1}}{2\Delta x}\right)\left(\frac{\lambda_{i+1} - \lambda_{i-1}}{2\Delta x}\right)$$

$$... = 0$$

- Rearranging for T_i:

$$T_i = \frac{T_{i+1} + T_{i-1}}{2} + \frac{(T_{i+1} - T_{i-1})(\lambda_{i+1} - \lambda_{i-1})}{8\lambda_i}$$