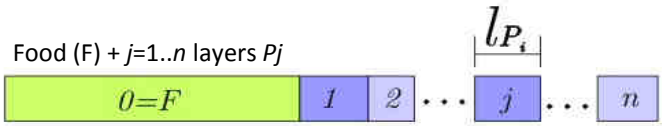
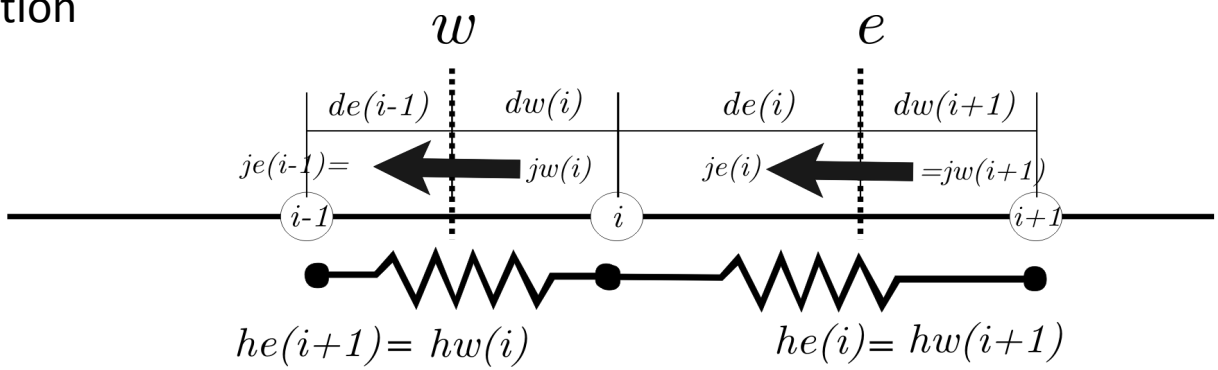
Spatial discretization scheme for *n*-layers materials based on a Finite Volume formulation on a 1D unstructured mesh (N nodes)

Food (F) + *j*=1..*n* layers *Pj*



Details of the discretization scheme: all nodes *i*=1..*N*, where the solution is calculated are separated by interfaces where all properties (*C*, *D*, *k*) can be discontinuous.



The fluxes at each interface between nodes ($0 \le i \le N$) are exactly computed assuming i) a steady transport, ii) the conservation of mass flux and iii) the local thermodynamical equilibrium on both sides of the interface:

$$\begin{cases} k(i-1)\cdot C(w-) = k(i)\cdot C(w+) \\ k(i)\cdot C(e-) = k(i+1)\cdot C(e+) \end{cases}$$

Note that C(w-), C(w+), C(e-) and C(e+) are not calculated explicitly but are included in the equivalent conductances hw and he, defined respectively between *i*-1 and *i*, and between *i* and *i*+1 (weak formulation of transport equations)

Other comments

Discontinuities in *D* or *k* are only handled at interfaces between nodes (*w* and *e)*

i=0 stands for the node in food

je(*N*)=0 (impervious condition)

For further details on FV formulations, see in particular page 45 of Patankar S.P. (1980) Numerical Heat Transfer And Fluid Flow, Hemisphere Publishing Corporation

$$\frac{dw(i+1)}{de(i)} = \frac{D(i+1)}{D(i)}\cdot\frac{k(i)}{k(i+1)}$$ *condition to optimize the mesh*

$$hw(i) = \frac{1}{\frac{de(i-1)}{D(i-1)}\cdot\frac{k(i-1)}{k(i)} + \frac{dw(i)}{D(i)}}$$ *equivalent conductance in the "west" direction*

$$he(i) = \frac{1}{\frac{de(i)}{D(i)}\cdot\frac{k(i)}{k(i+1)} + \frac{dw(i+1)}{D(i+1)}}$$ *equivalent conductance in the "east" direction*

$$jw(i) = hw(i)\cdot\left(\frac{k(i-1)}{k(i)}C(i-1) - C(i)\right)$$ *normal mass flux in the w direction*

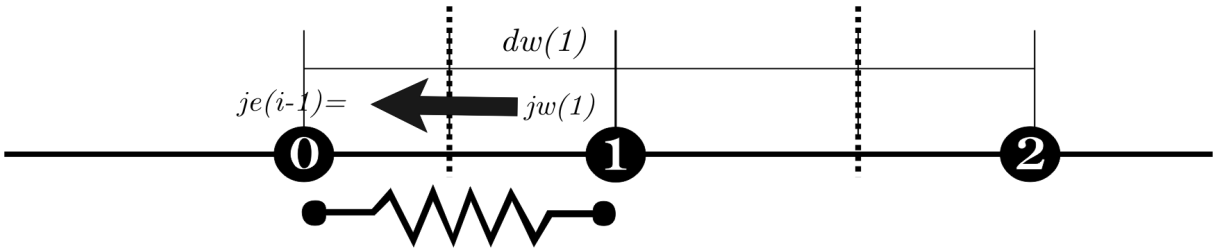
$$je(i) = he(i)\cdot\left(\frac{k(i)}{k(i+1)}C(i) - C(i+1)\right)$$ *normal mass flux in the e direction*

$$\frac{dC(i)}{dt} = \frac{jw(i) - je(i)}{dw(i) + de(i)} = \frac{1}{dw(i) + de(i)}\cdot\begin{bmatrix} hw(i)\cdot\frac{k(i-1)}{k(i)} \\ -hw(i) - he(i)\cdot\frac{k(i)}{k(i+1)} \\ he(i) \end{bmatrix}'\cdot\begin{bmatrix} C(i-1) \\ C(i) \\ C(i+1) \end{bmatrix}$$

Local mass balance (a line in Matrix A, see next page)

Robin boundary condition between P and F

$$\begin{cases} hw(1) = \dfrac{1}{\dfrac{1}{h}\cdot\dfrac{k(0)}{k(1)} + \dfrac{dw(1)}{D(1)}} \\ jw(1) = hw(1)\cdot\left(\dfrac{k(0)}{k(1)}CF - C(1)\right) \\ \dfrac{dCF}{dt} = -\dfrac{jw(1)}{l_F} = \dfrac{1}{l_F}\cdot\begin{bmatrix} -hw(1)\cdot\dfrac{k(0)}{k(1)} \\ hw(1) \end{bmatrix}'\cdot\begin{bmatrix} CF \\ C(1) \end{bmatrix} \end{cases}$$



Use the equivalent code below, to generate your own code in your own language. An implementation in Excel is even possible even if time integration is not an easy task.

Full assembled problem as a (N+1)x(N+1) triband matrix

(the equivalent Matlab® code to include the 2 boundaries and Mass balance is detailed on the right)

$$\begin{cases} \dfrac{d\underline{C}}{dt} = \underline{\underline{A}}\cdot\underline{C} \\ \underline{C}(t=0) = \underline{C}_0 \end{cases}$$

best integrated with an implicit method (Maxorder=2) and a dimensionless formulation.

```
% equivalent conductances
he = zeros(N,1); hw = he;
hw(1) = 1/( (k0/k(1))/h + dw(1)/(D(1))   ); % pervious contact
for i=2:N
    hw(i) = 1/( (de(i-1)/D(i-1))*(k(i-1)/k(i)) + dw(i)/D(i)  );
end
he(1:end-1) = hw(2:end); %flux continuity
he(end) = 0; % impervious BC
% Assembling
A = zeros(N+1,N+1);
% fluid phase = node 0 (row 1 in A)
A(1,1:2) = hw(1)/lF * [-k0/k(1)
                        1          ]';
% node 1 (row 2 in A)
A(2,1:3) = 1/(dw(1)+de(1)) * [ hw(1)*k0/k(1)
                               -hw(1)-he(1)*k(1)/k(2)
                                he(1) ]';

% node i<N (row i+1 in A)
for i=2:N-1
    A(i+1,i:i+2) = 1/(dw(i)+de(i)) * [ hw(i)*k(i-1)/k(i)
                                       -hw(i)-he(i)*k(i)/k(i+1)
                                        he(i) ]';

end
% node n (row N+1)
A(end,end-1:end) = 1/(dw(N)+de(N)) * [ hw(N)*k(N-1)/k(N)
                                       -hw(N)]';
```