INSULATED BOX WITH INTERNAL HEATING (SÓLIDO AISLADO CON DISIPACIÓN INTERNA)

Statement

Find the steady temperature field in a solid box of side L=0.1 m and thermal conductivity k=10 W/(m·K), with uniform internal heat dissipation of $\dot{W}_{\text{dis}}$ =10 W, thermally insulated all around except at one face, and with T_{m}=100 ºC at the opposite face.

Calcular el campo de temperaturas en un sólido cúbico de 0,1 m de lado y conductividad térmica k=10 W/(m·K), con una disipación de energía interna uniforme de $\dot{W}_{\text{dis}}$ =10 W, estando aislado térmicamente por todas sus caras menos una, y con T_{m}=100 ºC en la cara opuesta.

Solution

This is a one-dimensional (1D) configuration of heat transfer at steady state, with a simple solution. The generic heat equation (a parabolic PDE) becomes a simple second-order ODE, easily integrated to yield the explicit analytical solution:

$$\rho c\frac{\partial T}{\partial t}=k\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}+\frac{\partial^2 T}{\partial z^2}\right)+\phi \xrightarrow{\text{steady-1D}} 0=k\frac{\mathrm{d}^2 T}{\mathrm{d}x^2}+\phi \rightarrow T(x)=T_0-\frac{\phi}{2k}x^2$$

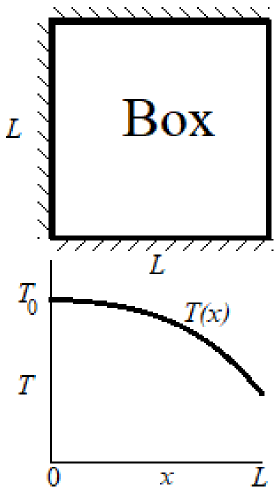

Fig. 1. Temperature profile

i.e. a parabolic T-profile along the heat-transfer direction, $0 \le x \le$ (we chose x for the basic coordinate, as nothing changes along the y-axis and z-axis), with a volumetric heat source, $\phi=\dot{W}_{\text{dis}}/L^3$ (in our case $\phi=\dot{W}_{\text{dis}}/L^3=10/0.1^3=10$ kW/m^3), a maximum temperature at x=0 (in our case $T_0=T_{\text{m}}$=100 ºC), and the two boundary conditions applied: at x=0, both $T=T_0$, and dT/dx=0. The two concrete results that can be extracted are:

1. The temperature at the face where heat flows out of the box is:

$$T_L=T_0-\frac{\phi}{2k}L^2=T_0-\frac{\dot{W}_{\text{dis}}}{2kL}=100-\frac{10}{2{\cdot}10{\cdot}0.1}=95\ \text{ºC}$$

2. The temperature slope at x=L confirms that the whole $\dot{W}_{\text{dis}}$=10 W flows out:

$$\frac{\mathrm{d}T}{\mathrm{d}x}=-\frac{\phi}{k}x=-\frac{\dot{W}_{\text{dis}}}{kL^3}x\overset{x=L}{=}-\frac{\dot{W}_{\text{dis}}}{kL^2}=-\frac{10}{10\cdot 0.1^2}=-100\ \frac{\mathrm{K}}{\mathrm{m}}\quad\rightarrow\quad \dot{W}_{\text{dis}}=-\dot{Q}_L=kA\left.\frac{\mathrm{d}T}{\mathrm{d}x}\right|_{x=L}$$

This simple problem poses, however, some difficulties to be solved by standard numerical methods.

One may be tempted to go directly to a versatile numerical heat-transfer simulation based on the finite element method (FEM); we shall use a generic FEM provided in Matlab's pde-toolbox. Of course, our 1D problem is too simple for such a tool (to begin with, this FEM implementation requires at least 2D geometry, but this is not a difficulty but a waste of computation time). The real problem arises when we try to impose the boundary conditions: we know that the edge at x=0 is adiabatic (edge E4 in Fig. 2a), but we know nothing at x=L (edge E2; edges E1 and E3 are adiabatic). As we know that at the steady state T_0=100 ºC, we might be tempted to fix that instead of $\mathrm{d}T/\mathrm{d}x|_{x=0}$=0, but this yields an absurd result, because the programme assumes that all faces of the box are adiabatic, and the heating is unbounded (yet, one has to think about, because the program gives no clue for the absurd result). Fortunately, a global thermal analysis teaches that, at the steady state, the heat flux (or the temperature gradient) at x=L can be obtained from $\dot{W}_{\text{dis}}=\dot{Q}_L=-kA\left(\mathrm{d}T/\mathrm{d}x\right)_{x=L}$ and now the program yields the right solution. We can see that $\mathrm{d}T/\mathrm{d}x|_{x=0}$=0 in the result. The Matlab code and its output are:

```
% Statement:
L=0.1; A=L*L; Tm=100; Q=10; phi=Q/L^3; k=10; Tp=-Q/(k*A); Nx=10; X=linspace(0,L,Nx);
md=createpde;Box=[3,4,0,L,L,0,0,0,L,L]';gm=[Box];ns=char('Box');ns=ns';sf='Box';
dg=decsg(gm,sf,ns);geometryFromEdges(md,dg);
subplot(2,2,1);pdegplot(md,'EdgeLabels','on','FaceLabels','on');grid;xlabel('X [m]');ylabel('Y [m]');
generateMesh(md);subplot(2,2,3);pdemesh(md);xlabel('X [m]');ylabel('Y [m]');

%State pde-equation and boundary conditions
specifyCoefficients(md,'m',0,'d',0,'c',k*L,'a',0,'f',phi*L);
applyBoundaryCondition(md,'Edge',4,'u',Tm);
applyBoundaryCondition(md,'Edge',2,'g',k*L*Tp);

%Solve and plot result
Sol=solvepde(md);u=Sol.NodalSolution;
subplot(2,2,2);pdeplot(md,'XYData',u,'Contour','on','ColorMap','autumn');grid;xlabel('X [m]');ylabel('Y [m]');
Y=X; T_=interpolateSolution(Sol,X,Y); subplot(2,2,4);plot(X,T_);xlabel('X [m]');ylabel('T [ºC]');
```

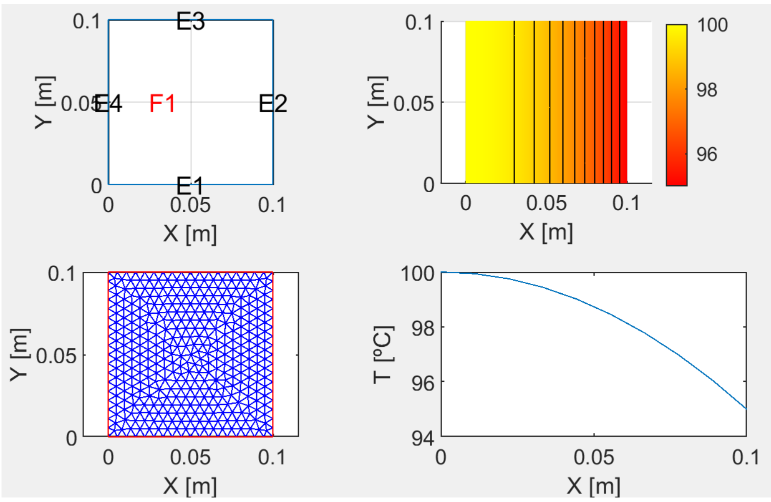

Fig. 2. Temperature profile obtained by FEM, for T=100 ºC at x=0 and $\mathrm{d}T/\mathrm{d}x|_{x=L}$=$-\dot{W}_{\text{dis}}/(kA)$.

However, what if we wanted to simulate the transient from say an initial $T(x)$=0 ºC to that final state? Of course, some thermal capacity is needed; let us assume typical values of ρ=1000 kg/m^3 and c=1000 J/(kg·K). During that transient we know that $dT/dx|_{x=0}=0$; but the solver requires one boundary condition at each edge. If we insist on applying the same boundary conditions as for the steady case, we get the same final result, independently of the initial condition and the thermal capacity assumed, but the intermediate T-profiles are all wrong because $T(0)$=100 ºC and $dT/dx|_{x=L}=-\dot{W}_{dis}/(kA)$=−100 K/m are only valid as $t\to\infty$ but not during the transients. Incidentally, we need to provide a total simulation time. Although we may set a value, and see from the results if we were too short or too long, it is better to have an idea of the proper order of magnitude, which for conduction-limited problems is $\Delta t_c=\rho c L^2/k=10^3\cdot 10^3\cdot 0.1^2/10=10^3$ s. The coding in Matlab language, and its results are:

```
% Statement:
L=0.1; A=L*L; Tm=100; Q=10; phi=Q/L^3; k=10; rho=1000; c=1000; Tp=-Q/(k*A);
Nx=20; Nt=50; Tini=0; X=linspace(0,L,Nx); T=Tini*ones(Nt,Nx); tsim=2e3; tlist=logspace(0,log10(tsim),Nt)';
md=createpde;Box=[3,4,0,L,L,0,0,0,L,L]';gm=[Box];ns=char('Box');ns=ns';sf='Box';
dg = decsg(gm,sf,ns);geometryFromEdges(md,dg);
subplot(2,3,1);pdegplot(md,'EdgeLabels','on','FaceLabels','on');grid;xlabel('X');ylabel('Y');
generateMesh(md);subplot(2,3,4);pdemesh(md);xlabel('X');ylabel('Y');

%Set equation, boundary conditions and initial conditions
specifyCoefficients(md,'m',0,'d',rho*c*L,'c',k*L,'a',0,'f',phi*L);
applyBoundaryCondition(md,'Edge',4,'u',Tm);
applyBoundaryCondition(md,'Edge',2,'g',k*L*Tp);
ic=@(~)Tini;setInitialConditions(md,ic);

%Solve and plot result
Sol=solvepde(md,tlist);u=Sol.NodalSolution;
subplot(2,3,2);pdeplot(md,'XYData',u(:,end),'Contour','on','ColorMap','autumn');grid;xlabel('X');ylabel('Y');
subplot(2,3,5);pdeplot(md,'XYData',u(:,end),'ZData',u(:,end),'ColorMap','autumn');grid;xlabel('X');ylabel('Y');
Y=X; T_=interpolateSolution(Sol,X,Y,[1:Nt]);    %Interpolation from T(tri-nodes) to T(sqr-mesh)
subplot(2,3,3);plot(X,T_);xlabel('X [m]');ylabel('T [ºC]');
subplot(2,3,6);plot(tlist,T_);xlabel('time [s]');ylabel('T [ºC]');
```

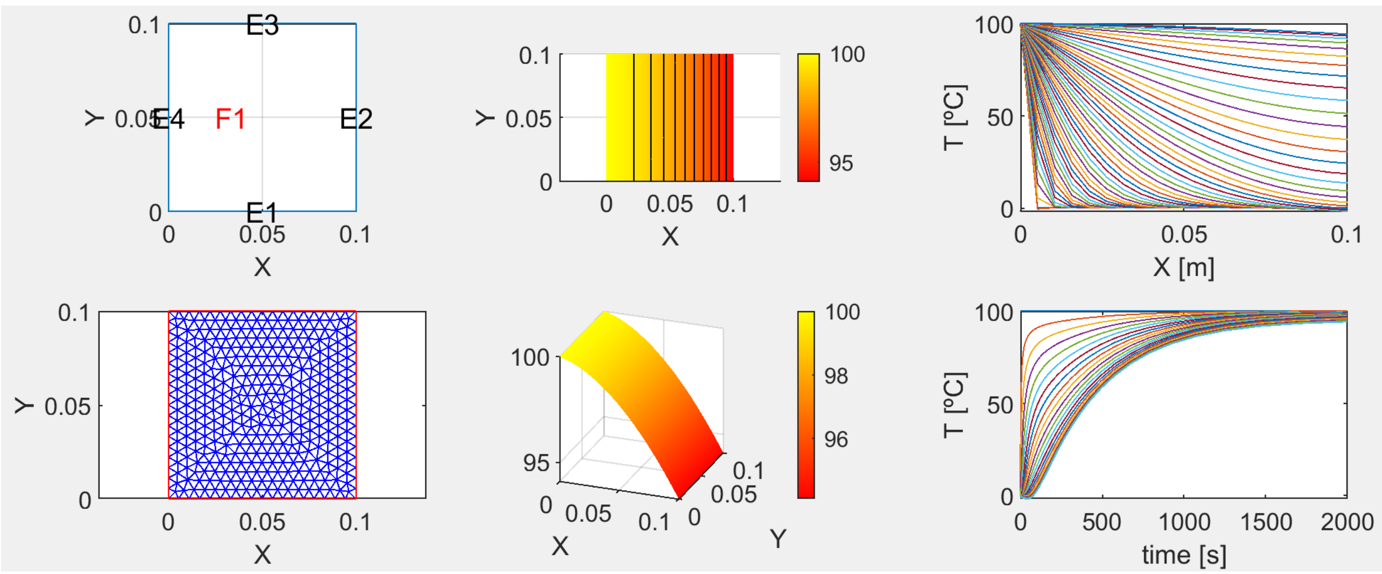

Fig. 3. Temperature evolution by FEM, from T_{ini}=0 ºC, for fix boundary conditions: $T|_{x=0}$=100 ºC and $dT/dx|_{x=L}=-hA(T_L-T_{ini})$.

The final $T(x)$-profile in Fig. 3 is correct (a parabolic drop from $T(0)$=100 ºC to $T(1)$=95 ºC, coincident with Fig. 2), but the transients are all wrong; we expect an almost flat $T(x)$-profile growing with time. The heat-transfer problem is ill posed (but it may take time to realise why).

Consider now the finite difference method (FDM) in the explicit form, which is simpler and easily followed step by step, contrary to the FEM. We starts from an initial profile, e.g. $T(x,t)|_{t=0}$=0 ºC as with the FEM, and build new $T(x)$-profiles as time advances, using the nodal points shown in Fig. 4 and the only known boundary condition during the transient, namely the adiabatic condition at x=0. The discretization to be used (Fig. 4) is:

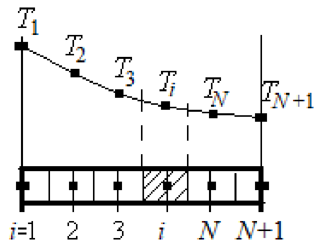

Fig. 4. Nodal scheme used in our FDM.

- At intermediate nodes ($2 \le i \le N$ in Fig. 4), the energy balance is:

$$\rho c \frac{\partial T}{\partial t} = k\frac{\partial^2 T}{\partial x^2} + \phi \quad \rightarrow \quad \rho A \Delta x c \frac{T_i^{j+1} - T_i^j}{\Delta t} = kA\left(\frac{T_{i+1}^j - T_i^j}{\Delta x} - \frac{T_i^j - T_{i-1}^j}{\Delta x}\right) + \phi A \Delta x \quad \rightarrow$$

$$\rightarrow T_i^{j+1} = T_i^j + \frac{k\Delta t}{\rho c (\Delta x)^2}\left(T_{i+1}^j - 2T_i^j + T_{i-1}^j\right) + \frac{\phi \Delta t}{\rho c}$$

- At boundary nodes (note that their mass is half of the generic ones), on the left (i=1):

$$\rho A \frac{\Delta x}{2} c \frac{T_1^{j+1} - T_1^j}{\Delta t} = kA\frac{T_2^j - T_1^j}{\Delta x} + \phi A \frac{\Delta x}{2} - \dot{Q}_0 \xrightarrow{\dot{Q}_0 = 0} T_1^{j+1} = T_1^j + \frac{\Delta t}{\rho c}\left[2k\frac{T_2^j - T_1^j}{(\Delta x)^2} + \phi\right]$$

- and on the right ($i=N+1$):

$$\rho A \frac{\Delta x}{2} c \frac{T_{N+1}^{j+1} - T_{N+1}^j}{\Delta t} = -k\frac{T_{N+1}^j - T_N^j}{\Delta x} + \phi A \frac{\Delta x}{2} + \dot{Q}_L$$

  where now the heat going outside at the right-hand side during the transients, $-\dot{Q}_L(t)$, is unknown; hence, we cannot solve for updating T_{N+1}^j as done for T_1^j above.

We now realise that we must impose an additional boundary condition at $x=L$; thinking on the real physics (often forgotten when immersed in the interpretation of massive results from numerical schemes and the effect of discretization parameters), we realise that we are free to fix $\dot{Q}_L$, or to fix T_{N+1}^j, or a convective coefficient, h, to an ambient fluid at T_∞, such that $\dot{Q}_L = hA\left(T_\infty - T_{N+1}^j\right)$, with a presumable $T_\infty = T_{ini}$=0 ºC in our case, or thermal radiation cooling in vacuum within a large enclosure, $\dot{Q}_L = \varepsilon A \sigma\left(T_\infty^4 - T_{N+1}^{4,j}\right)$, where ε is the emissivity of the right face. For instance, if we choose the convective boundary condition with in calm air with a typical value of h=10 W/(m^2·K), the Biot number $Bi=hL/k$=10·0.1/10=0.1 being Bi<<1 indicates that now the heat transfer is limited by convection, and the characteristic relaxation times is $\Delta t_c=\rho cL/h=10^3\cdot10^3\cdot0.1/10=10^4$ s, an order of magnitude larger than in the conduction-limited case above. The Matlab code and its results are:

```
%Statement:
L=0.1; A=L*L; Q=10; phi=Q/L^3; k=10; rho=1000; c=1000; tsim=1e5; Tini=0; Tinf=Tini; h=10;
N=20; M=10e4; a=k/(rho*c); Dx=L/N; Dt=tsim/M; Fo=a*Dt/(Dx*Dx)
X=linspace(0,L,N+1); t=linspace(0,tsim,M+1)'; T=Tini*ones(M+1,N+1);

%Iteration:
for j=2:M+1
  T(j,1)=T(j-1,1)+(2*Dt/(rho*c))*(k*(T(j-1,2)-T(j-1,1))/Dx^2+phi/2);
  for i=2:N
    T(j,i)=T(j-1,i)+(Dt/(rho*c*A))*((k*A*(T(j-1,i+1)-T(j-1,i))-k*A*(T(j-1,i)-T(j-1,i-1)))/Dx^2+phi*A);
  end
  T(j,N+1)=T(j-1,N+1)+(2*Dt/(rho*c))*(k*(T(j-1,N)-T(j-1,N+1))/Dx^2+phi/2-h*(T(j-1,N+1)-Tinf)/Dx);
end

%Results
subplot(2,2,1);plot(X,T(1:M/25:M+1,:));xlabel('X [m]'),ylabel('T [K]');title('T(t,x) vs. X at several times');
subplot(2,2,3);plot(t,T(:,1:N/10:N+1));xlabel('t [s]'),ylabel('T [K]');title('T(t,x) vs. t at several locations');
subplot(2,2,2);plot(X,T(1:M/25:M+1,:));xlabel('X [m]'),ylabel('T [K]');title('T(t,x) vs. X at several times');axis([0 L 95 105]);
```

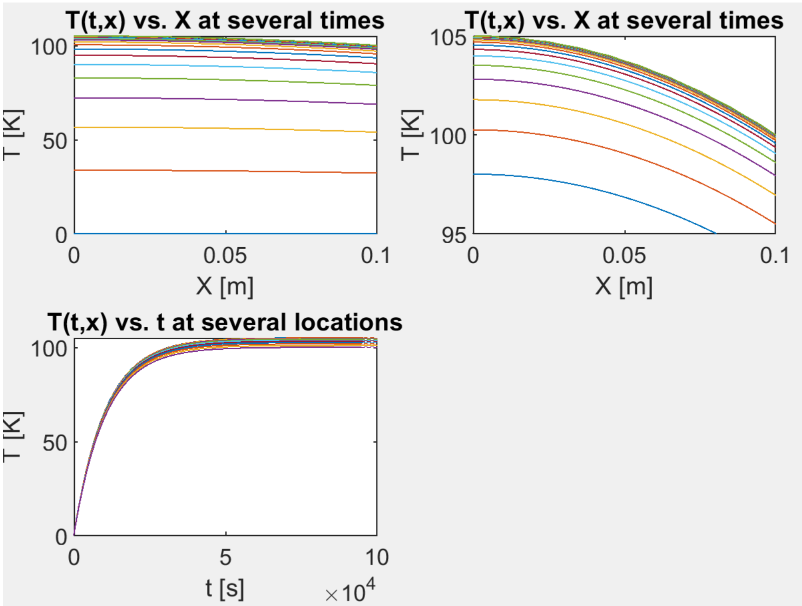

Fig. 5. Temperature evolution by FDM, from T_{ini}=0 ºC at t=0, with $dT/dx|_{x=0}=0$, and $dT/dx|_{x=L}=-hA(T_L-T_{ini})$. The details near the steady state are shown at the right.

The results in Fig. 5a show an almost uniform warming of the material, as expected, but with the final state (better seen in Fig. 5b) in disagreement with what we expected, the parabolic $T(x)$ starting with T_0=100 ºC and ending with T_L=95 ºC seen in Fig. 2; in fact, we got the same profile but shifted upwards 5 ºC. We can also see that, doubling the convective coefficient, yields a similar end result but shifted downwards to T_0=55 ºC and T_L=50 ºC. In each case, we may check that at the steady state all the power dissipated inside goes out.

In conclusion: the steady-state solution required in the statement cannot be directly solved as the limit of a transient problem if we ignore the boundary conditions during the transient, and fixing them (e.g. $\partial T(t,x)/\partial x|_{x=0}=0$ and $-k\partial T(t,x)/\partial x|_{x=L}=hA(T(t,x)-T_\infty)$ with given h and T_∞) usually do not produce the desired end result (here T_0=100 ºC). Fortunately, in many cases there is no need to iterate on the parameter (e.g. h) to match the end result: at the steady state, the internal solution yielded $T_0-T_L=\dot{W}_{dis}/(2kL)$=10/(2·10·0.1)=5 K, i.e. T_L=95 ºC, while the heat balance at the external surface, $\dot{W}_{dis}=hA(T_L-T_\infty)$, yields the appropriate value of $h=\dot{W}_{dis}/\left(A\left(T_L-T_\infty\right)\right)$=10/(0.1^2·(95-0))=10.5 W/(m^2·K). In the case of pure radiative cooling the heat balance at the external surface, $\dot{W}_{dis}=\varepsilon A\sigma\left(T_L^4-T_\infty^4\right)$, would yield the maximum value of T_∞ to cope with this heat, $T_\infty=\left[T_L^4-\dot{W}_{dis}/\left(A\sigma\right)\right]^{1/4}$=[(95+273)^4–10/(0.1^2·5.67·10^{-8})]=162 K (smaller for emissivity ε<1). Finally, the general heat transfer formulation $\dot{Q}=KA\Delta T$, teaches that, to evacuate the stated power $\dot{Q}$, instead of increasing the heat transmittance K (in this case increasing the convection coefficient, e.g. by blowing air with a fan), or increasing ΔT (in this case decresing the environmental temperature), the simplest solution may be to increase the cooling area A (e.g. by attaching a larger conductive plate).

Comments

This exercise teachs how embarrassing numerical simulation may become if we dive into discretization details without paying proper attention to the physical problem.

Back to Heat and mass transfer

Back to Thermodynamics