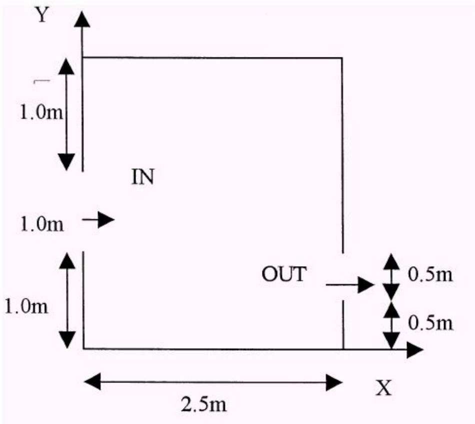

The boundary conditions are:

At the left, right, top and bottom walls:
u=0, v=0, T=0.

At the inlet:
u=1, v=0, T=1.
At the outlet :
Neuman conditions for *u*,*v* and *T*

Fig. 1. The geometry and boundary conditions for the 2D-airflow problem.

U-momentum equation

$$\frac{\partial u}{\partial t} = -\frac{\partial (uu)}{\partial x} - \frac{\partial (vu)}{\partial y} - \frac{\partial p}{\partial x} + \frac{1}{\mathrm{Re}}\nabla^2 u$$

V-momentum equation

$$\frac{\partial v}{\partial t} = -\frac{\partial (uv)}{\partial x} - \frac{\partial (vv)}{\partial y} - \frac{\partial p}{\partial y} + \frac{1}{\mathrm{Re}}\nabla^2 v + \frac{Gr}{\mathrm{Re}^2}T$$

Continuity equation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

Energy equation

$$\frac{\partial T}{\partial t} = -\frac{\partial (uT)}{\partial x} - \frac{\partial (vT)}{\partial y} + \frac{1}{\mathrm{Re}\,\mathrm{Pr}}\nabla^2 T$$

```
%Variables: u1=u; u2=v; u3=p u4=T; u1x = du1/dx etc.
fem.geom=poly2([0 0 0 0 2.5 2.5 2.5 2.5],[0 1 2 3 3   1   0.5 0] ); %Specify geometry
fem.mesh=meshinit(fem); fem.mesh=meshrefine(fem);   %Mesh init and refine
fem.dim=4; fem.form='general';                      %4 variables, general form
Re=1000;Gr=2.5e7; Pr=0.71;                          %Model parameters
etaval=1/Re; betaval=Gr/(Re*Re);alphaval=1/(Re*Pr);
fem.variables={'eta' etaval 'alpha' alphaval 'beta' betaval }; %Model variables

fem.bnd.r={ {'-u1'; '-u2'; 0; '0-u4'} ...           %Specify boundary coefficient r
               {'-u1'; '-u2'; 0; '0-u4'} ...
           {'1-u1'; '-u2'; 0; '1-u4'} ...
               {'-u1'; '-u2'; 0; '0-u4'} ...
               {'-u1'; '-u2'; 0; '0-u4'} ...
               {'-u1'; '-u2'; 0; '0-u4'} ...
               {0; 0; '-u3'; 0} ...
        {'-u1'; '-u2'; 0; '0-u4'} };
fem.equ.da={{1; 1; 0; 1}};                                  %Specify PDE coefficient da
fem.equ.f={{'-(u1.*u1x+u2.*u1y+u3x)';...            %Specify PDE coefficient f
            '-(u1.*u2x+u2.*u2y+u3y)+beta*u4';...
            '-(u1x+u2y)';...
            '-(u1.*u4x+u2.*u4y)'}};
fem.equ.ga={{{'-eta*u1x'; '-eta*u1y'};...           %Specify  PDE coefficient ga
            {'-eta*u2x'; '-eta*u2y'};...
            0;...
            {'-alpha*u4x'; '-alpha*u4y'}}};

fem=femdiff(fem);                               %Calculate divergence gamma on domain
fem.init={{0 0 0 0}};                                          % Intial values
fem.sol=femtime(fem,'atol',{{.1,.1,Inf,.1}},'ode','fldae',...       %Solving
               'sd','on','report','on','tlist',0:.3:30);
```

Fig. 2. The PDE model and the corresponding FemLab code