1 main.m

%~~~~~~~~~~~~~~~~~~~~~~~

% This is main module

%~~~~~~~~~~~~~~~~~~~~~~~

% Initialize

init\_all();

% Enter main loop

while t<tend

comp\_delt();

setbcond();

%comp\_temp();

comp\_fg();

comp\_rhs();

p=poisson(RHS,imax,jmax,jI,iB,dx,dy,eps,itermax,omg,p);

adap\_uv();

store\_result();

k=k+1;

t=t+dt;

disp([t, round(k), dt]);

end

post

2 initial\_all.m

tend=30;t=0;k=1; % End time, start time, # of main loop

imax=88;jmax=200; % Mesh Size

xlength=1.1;ylength=2.5; % Base domain

dx=xlength/imax;dy=ylength/jmax;

% Define domain

xB=0.3;xC=0.6;xD=0.8;xG=0.15;yI=1.2;

iB=round(xB/dx);iC=round(xC/dx);iD=round(xD/dx);iG=round(xG/dx);jI=round(yI/dy);

% computation control

tau=0.5;itermax=100;eps=0.001;omg=1.7;

gamma=0.9;Re=17000;gx=0;gy=0;

ui=0; vi=0; pi=0;

Pr=0.7;beta=0.00034;

% Initialize U,V,P,T

ui=0;vi=0;pi=0;uin=-1;vin=1;

T\_room=298; % Room temperature 25

T\_inlet=293; % Inlet air temperature

T\_svTop=318; % Server top temperature

T\_svRight=318; % Server right temperature

init\_uvpt();

% Initialize t,k

t0=0;k=1;

% Store solution

u\_solution=zeros(202,90,1000);

v\_solution=zeros(202,90,1000);

3 init\_uvp.m

u=ui\*ones(jmax+2,imax+2);

v=vi\*ones(jmax+2,imax+2);

p=pi\*ones(jmax+2,imax+2);

4 init\_uvpt.m

% This module initialize U,V,P and T

u=ui\*ones(jmax+2,imax+2);

v=vi\*ones(jmax+2,imax+2);

p=pi\*ones(jmax+2,imax+2);

T=T\_room\*ones(jmax+2,imax+2);

5 setbcond.m

%~~~~~~~~~~~~~~~~~~~~~~~~~~

% Set u,v boundary

%~~~~~~~~~~~~~~~~~~~~~~~~~~

% BC, no-slip

u(1,iB+2:iC+1)=-u(2,iB+2:iC+1);

v(1,iB+2:iC+1)=0;

% CD, inlet

u(1,iC+2:iD+1)=2\*uin-u(2,iC+2:iD+1);

v(1,iC+2:iD+1)=vin;

% DE, no-slip

u(1,iD+2:imax+1)=-u(2,iD+2:imax+1);

v(1,iD+2:imax+1)=0;

% EF, no-slip

u(2:jmax+1,imax+1)=0;

v(2:jmax+1,imax+2)=-v(2:jmax+1,imax+1);

% GF, no-slip

u(jmax+2,iG+2:imax+1)=-u(jmax+1,iG+2:imax+1);

v(jmax+1,iG+2:imax+1)=0;

% HG, outflow

u(jmax+2,2:iG+1)=u(jmax+1,2:iG+1);

v(jmax+1,2:iG+1)=v(jmax,2:iG+1);

% HI, symmerty

u(jI+2:jmax+1,1)=0;

v(jI+2:jmax+1,1)=v(jI+2:jmax+1,2);

% IJ, no-slip (server top)

u(jI+1,2:iB)=u(jI+2,2:iB);

v(jI+1,2:iB)=0;

% Corner J, no-slip on N,E

u(jI+1,iB+1)=0;

v(jI+1,iB+1)=0;

% JB, no-slip (server right)

u(2:jI,iB+1)=0;

v(2:jI,iB+1)=-v(2:jI,iB+2);

%~~~~~~~~~~~~~~~~~~~~~~~~~~

% Set temperature boundary

%~~~~~~~~~~~~~~~~~~~~~~~~~~

%{

% BC, adiabatic

T(1,iB+2:iC+1)=2\*T\_room-T(2,iB+2:iC+1);

% CD, inlet

T(1,iC+2:iD+1)=2\*T\_inlet-T(2,iC+2:iD+1);

% DE, adiabatic

T(1,iD+2:imax+1)=2\*T\_room-T(2,iD+2:imax+1);

% EF, adiabatic

T(2:jmax+1,imax+2)=2\*T\_room-T(2:jmax+1,imax+1);

% GF, adiabatic

T(jmax+1,iG+2:imax+1)=2\*T\_room-T(jmax+1,iG+2:imax+1);

% HG, outflow

T(jmax+1,2:iG+1)=T(jmax,2:iG+1);

% HI, symmerty

T(jI+2:jmax+1,1)=T(jI+2:jmax+1,2);

% IJ, server top

T(jI+1,2:iB)=2\*T\_svTop-T(jI+2,2:iB);

% JB, server right

T(2:jI,iB+1)=2\*T\_svRight-T(2:jI,iB+2);

% Corner J

T(jI+1,iB+1)=(T(jI+1,iB+2)+T(jI+2,iB+1))/2;

%}

6 comp\_fg.m

% Initialize F and G

F=zeros(jmax+2,imax+2); G=zeros(jmax+2,imax+2);

% Calculate F,G interior values

for i=1:imax-1

for j=1:jmax

if (i<iB+1)&&(j<jI+1)

F(j,i)=F(j,i); % Cell is in solid domain

else

a=u(j+1,i+1)+u(j+1,i+1+1);

b=u(j+1,i-1+1)+u(j+1,i+1);

c=u(j+1,i+1)-u(j+1,i+1+1);

d=u(j+1,i-1+1)-u(j+1,i+1);

e=u(j+1,i+1)+u(j+1+1,i+1);

f=u(j-1+1,i+1)+u(j+1,i+1);

g=u(j+1,i+1)-u(j+1+1,i+1);

h=u(j-1+1,i+1)-u(j+1,i+1);

va=v(j+1,i+1)+v(j+1,i+1+1);

vb=v(j-1+1,i+1)+v(j-1+1,i+1+1);

u2x=1/dx\*((a/2)^2-(b/2)^2)...

+gamma\*1/dx\*(abs(a)/2\*c/2-abs(b)/2\*d/2);

uvy=1/dy\*(va/2\*e/2-vb/2\*f/2)...

+gamma\*1/dy\*(abs(va)/2\*g/2-abs(vb)/2\*h/2);

u2x2=(u(j+1,i+1+1)-2\*u(j+1,i+1)+u(j+1,i-1+1))/(dx^2);

u2y2=(u(j+1+1,i+1)-2\*u(j+1,i+1)+u(j-1+1,i+1))/(dy^2);

F(j+1,i+1)=u(j+1,i+1)+dt\*(1/Re\*(u2x2+u2y2)-(u2x)-(uvy)+gx)...

-beta\*dt/2\*(T(j,i)+T(j,i+1))\*gx;

end

end

end

for i=1:imax

for j=1:jmax-1

if (i<iB+1)&&(j<jI+1)

G(j,i)=G(j,i); % Cell is in solid domain

else

a=v(j+1,i+1)+v(j+1,i+1+1);

b=v(j+1,i-1+1)+v(j+1,i+1);

c=v(j+1,i+1)-v(j+1,i+1+1);

d=v(j+1,i-1+1)-v(j+1,i+1);

e=v(j+1,i+1)+v(j+1+1,i+1);

f=v(j-1+1,i+1)+v(j+1,i+1);

g=v(j+1,i+1)-v(j+1+1,i+1);

h=v(j-1+1,i+1)-v(j+1,i+1);

ua=u(j+1,i+1)+u(j+1+1,i+1);

ub=u(j+1,i-1+1)+u(j+1+1,i-1+1);

uvx=1/dx\*(ua/2\*a/2-ub/2\*b/2)...

+gamma\*1/dx\*(abs(ua)/2\*c/2-abs(ub)/2\*d/2);

v2y=1/dy\*((e/2)^2-(f/2)^2)...

+gamma\*1/dy\*(abs(e)/2\*g/2-abs(f)/2\*h/2);

v2x2=(v(j+1,i+1+1)-2\*v(j+1,i+1)+v(j+1,i-1+1))/(dx^2);

v2y2=(v(j+1+1,i+1)-2\*v(j+1,i+1)+v(j-1+1,i+1))/(dy^2);

G(j+1,i+1)=v(j+1,i+1)+dt\*(1/Re\*(v2x2+v2y2)-uvx-v2y+gy)...

-beta\*dt/2\*(T(j,i)+T(j+1,i))\*gy;

end

end

end

% Update F and G at boundaries

F(2:jmax+1,imax+1)=u(2:jmax+1,imax+1); % Room right, EF

F(2:jI+1,iB+1)=u(2:jI+1,iB+1); % Server right, JB

F(jI+2:jmax+1,1)=u(jI+2:jmax+1,1); % Room left up, HI

G(1,iB+2:imax+1)=v(1,iB+2:imax+1); % Room bottom, BE

G(jI+1,2:iB+1)=v(jI+1,2:iB+1); % Server top, IJ

G(jmax+1,2:imax+1)=v(jmax+1,2:imax+1); % Room top

7 comp\_rhs.m

RHS=zeros(jmax+2,imax+2);

for i=1:imax

for j=1:jmax

RHS(j+1,i+1)=1/dt\*((F(j+1,i+1)-F(j+1,i-1+1))/dx+(G(j+1,i+1)-G(j-1+1,i+1))/dy);

end

end

8 possion.m

function p = poisson(RHS,imax,jmax,jI,iB,dx,dy,eps,itermax,omg,p)

for it=1:itermax

% Copy pressure for boundary cells

p(2:jI+1,1)=p(2:jI+1,2); % Server right, JB

P(jI+2:jmax+1,1)=p(jI+2:jmax+1,2); % Room left, HI

P(2:jmax+1,imax+2)=p(2:jmax+1,imax+1); % Room right, EF

P(1,iB+2:imax+1)=P(2,iB+2:imax+1); % Room bottom, BE

P(jI+1,2:iB+1)=P(jI+2,2:iB+1); % Server top, IJ

P(jmax+1,2:imax+1)=P(jmax,2:imax+1); % Room top, HF

P(jI+1,iB+1)=(P(jI+2,iB+1)+P(jI+1,iB+2))/2; % Server NE corner, J

% Used to compute residual

rr=0; % Sum of r(j,i)

count=0; % Count number of fluid cells

for i=1:imax

for j=1:jmax

if (i<iB+1)&&(j<jI+1)

p(j,i)=p(j,i);

else

% Compute P(n+1)

eiw=1;eie=1;ejs=1;ejn=1;

p(j+1,i+1)=(1-omg)\*p(j+1,i+1)+...

omg/((eie+eiw)/(dx^2)+(ejn+ejs)/(dy^2))...

\*((eie\*p(j+1,i+1+1)+eiw\*p(j+1,i-1+1))/(dx^2)+...

(ejn\*p(j+1+1,i+1)+ejs\*p(j-1+1,i+1))/(dy^2)-RHS(j+1,i+1));

% Compute residual

r(j+1,i+1)=(eie\*(p(j+1,i+1+1)-p(j+1,i+1))-eiw\*(p(j+1,i+1)-p(j+1,i-1+1)))/(dx^2)...

+(ejn\*(p(j+1+1,i+1)-p(j+1,i+1))-ejs\*(p(j+1,i+1)-p(j-1+1,i+1)))/(dy^2)-RHS(j+1,i+1);

count=count+1;

rr=rr+r(j+1,i+1)^2;

end

end

end

Ls\_norm=(rr/count)^0.5;

if (Ls\_norm<eps)

disp('Converged...')

disp(Ls\_norm);

break;

end

end

9 adap\_uv.m

for i=1:imax-1

for j=1:jmax

if (i<iB+1)&&(j<jI+1)

u(j,i)=u(j,i);

else

u(j+1,i+1)=F(j+1,i+1)-dt/dx\*(p(j+1,i+1+1)-p(j+1,i+1));

end

end

end

for i=1:imax

for j=1:jmax-1

if (i<iB+1)&&(j<jI+1)

v(j,i)=v(j,i);

else

v(j+1,i+1)=G(j+1,i+1)-dt/dy\*(p(j+1+1,i+1)-p(j+1,i+1));

end

end

end

10 post.m

X=0:dx:xlength;

Y=0:dy:ylength;

[xx,yy]=meshgrid(X,Y);

uu=zeros(jmax+1,imax+1);

vv=zeros(jmax+1,imax+1);

for m=1:(imax+1)

for n=1:(jmax+1)

x=xx(n,m);

y=yy(n,m);

i=floor(x/dx)+1;

j=floor((y+dy/2)/dy)+1;

x1=(i-1)\*dx;

y1=((j-1)-0.5)\*dy;

x2=i\*dx;

y2=(j-1/2)\*dy;

u1=u(j+1-1,i+1-1);

u2=u(j+1-1,i+1);

u3=u(j+1,i+1-1);

u4=u(j+1,i+1);

uu(n,m)=1/(dx\*dy)\*((x2-x)\*(y2-y)\*u1+(x-x1)\*(y2-y)\*u2+(x2-x)\*(y-y1)\*u3+(x-x1)\*(y-y1)\*u4);

end

end

for m=1:imax

for n=1:jmax

x=xx(n,m);

y=yy(n,m);

j=floor(y/dy)+1;

i=floor((x+dx/2)/dx)+1;

y1=(j-1)\*dy;

x1=((i-1)-0.5)\*dx;

y2=j\*dy;

x2=(i-1/2)\*dx;

v1=v(j+1-1,i+1-1);

v2=v(j+1-1,i+1);

v3=v(j+1,i+1-1);

v4=v(j+1,i+1);

vv(n,m)=1/(dx\*dy)\*((x2-x)\*(y2-y)\*v1+(x-x1)\*(y2-y)\*v2+(x2-x)\*(y-y1)\*v3+(x-x1)\*(y-y1)\*v4);

end

end

axis([0,1.1,0,2.5]);

quiver(xx,yy,uu,vv,5);

axis([0,1.1,0,2.5]);

figure;

streamslice(xx,yy,uu,vv);

axis tight

11 store\_result

% Save u and v at certain k

k=round(k);

if k==1 || mod(k,5)==0

disp('Saved..');

index=k/5;

u\_solution(:,:,index)=u;

v\_solution(:,:,index)=v;

end

12 comp\_temp

% Compute Temperture for next timestep

Tnew=zeros(jmax+2,imax+2);

for i=2:imax+1

for j=2:jmax+1

if (i<iB+1)&&(j<jI+1)

Tnew(j,i)=T\_room;

else

AA=1/dx\*(u(j,i)\*(T(j,i)+T(j,i+1))/2-u(j,i-1)\*(T(j,i-1)+T(j,i))/2)...

+gamma/dx\*(abs(u(j,i))\*(T(j,i)-T(j,i+1))/2-abs(u(j,i-1))\*(T(j,i-1)-T(j,i))/2);

BB=1/dy\*(v(j,i)\*(T(j,i)+T(j,i+1))/2-v(j-1,i)\*(T(j-1,i)+T(j,i))/2)...

+gamma/dy\*(abs(v(j,i))\*(T(j,i)-T(j+1,i))/2-abs(v(j-1,i))\*(T(j-1,i)-T(j,i))/2);

CC=1/(Re\*Pr)\*((T(j,i+1)-2\*T(j,i)+T(j,i-1))/(dx.^2)+(T(j+1,i)-2\*T(j,i)+T(j-1,i))/(dy.^2));

Tnew(j,i)=T(j,i)+dt\*(CC-AA-BB);

end

end

end

T(2:jmax+1,2:imax+1)=Tnew(2:jmax+1,2:imax+1);

13 comp\_delt.m

if k==1

dt=0.02;

else

delta=1/dx^2+1/dy^2;

first=Re\*Pr/2/delta;

second=Re/2/delta;

third=dx/abs(max(max(u)));

fourth=dy/abs(max(max(v)));

dt=tau\*min([first,second,third,fourth]);

end