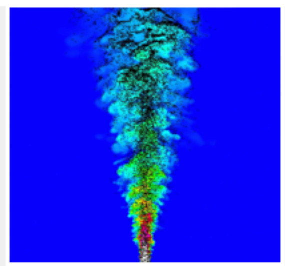
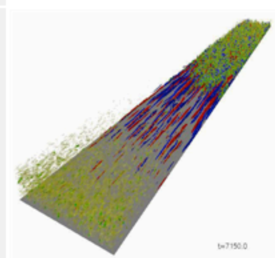
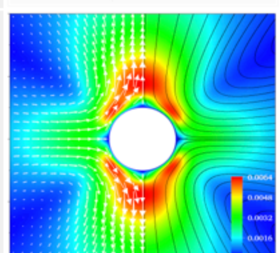
SG2225
Fluid mechanics, continuation

Course information

updated 2012-08-26

Course responsible: Prof Luca Brandt (LB) tel: 7906870 luca@mech.kth.se		Lecturer: Docent Anders Dahlkild (AD) tel: 7909174 ad@mech.kth.se
Book: Lecture notes Fluid Mechanics, P.K. Kundu & I.M. Cohen, 3rd edition **(E-book via KTHB: http://www.lib.kth.se/main/e-resurser.asp** **Referex engineering:** Materials & Mechanical Collection)		Numerical lab: Project description Matlab code
Downloads: Hand-written lecture notes: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ... Links: flow visualizations and educational movies		Oral examination: By appointment with the teacher (e-mail to luca@mech.kth.se)

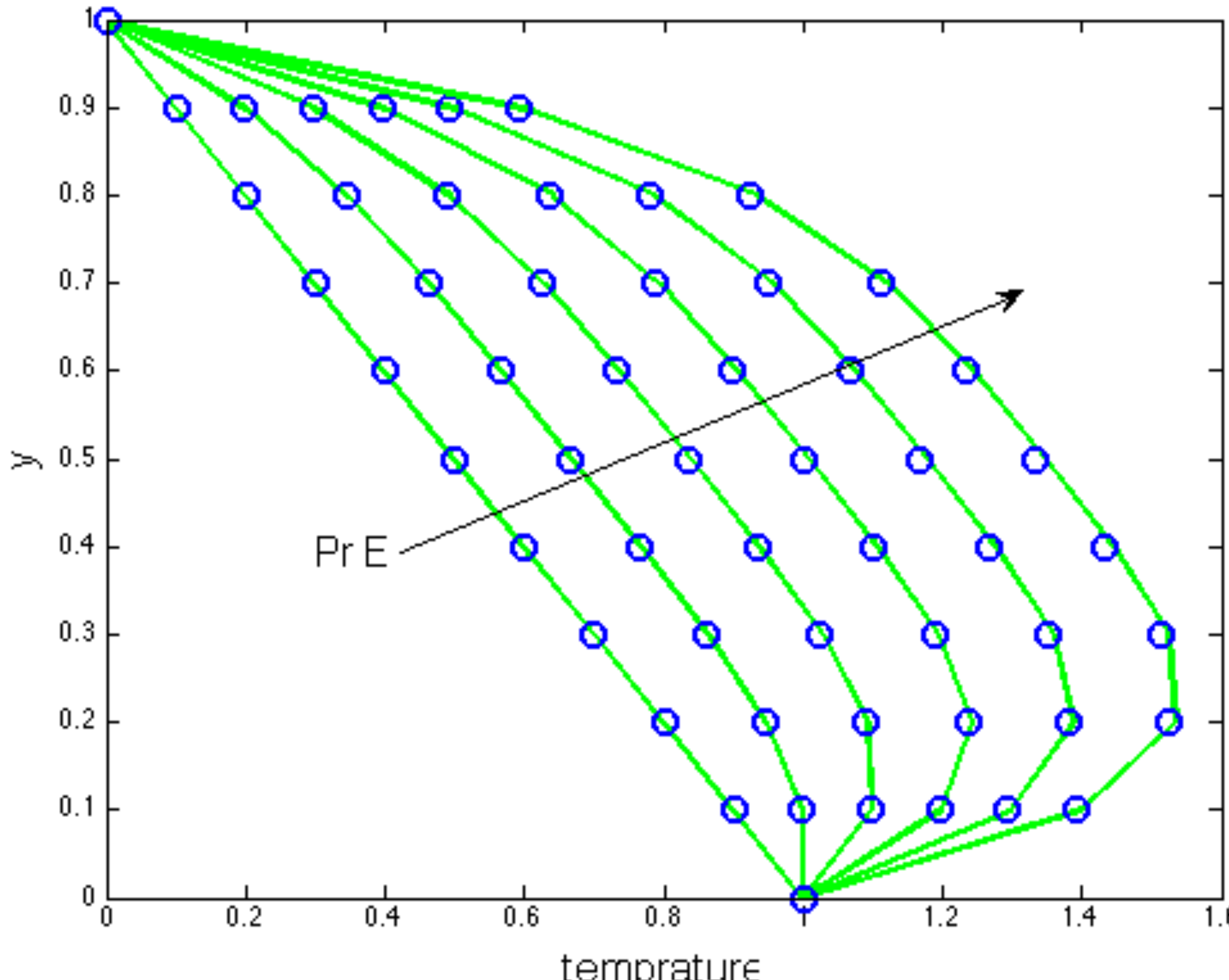
SG2225
Fluid mechanics, continuation

Homepage: http://www2.mech.kth.se/~luca/SG2225.html

Numerical Project: Matlab code

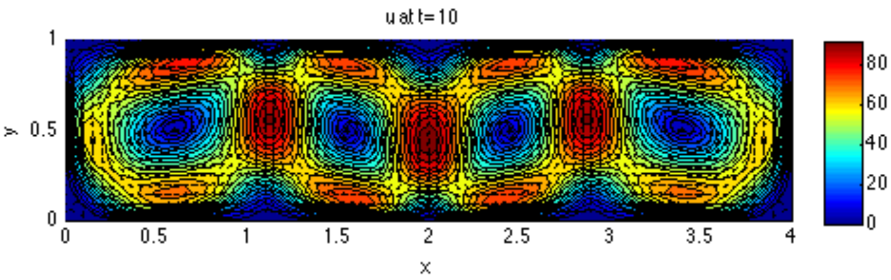
```
% Navier-Stokes solver, based on version 07/2007 by Benjamin Seibold
%           http://www-math.mit.edu/~seibold/
%
% Adapted for course SG2225
% KTH Mechanics
%
% This version: 20120830 PS
%
% Features:
% - Navier-Stokes for velocity and scalar fields
% - Boussinesq approximation for convection problems
% - explicit time integration for advection and viscous terms
% - central differencing for advection term
% - sparse matrices
%
% Depends on avg.m and DD.m
%
```

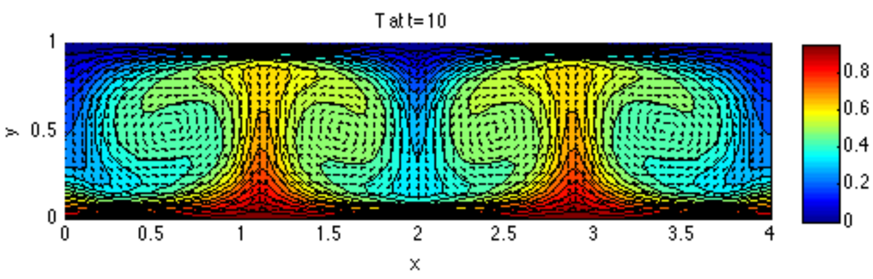
Poiseuille flow and passive scalar equation – effect of Dissipation – circles: analytical solution, lines: N-S solver



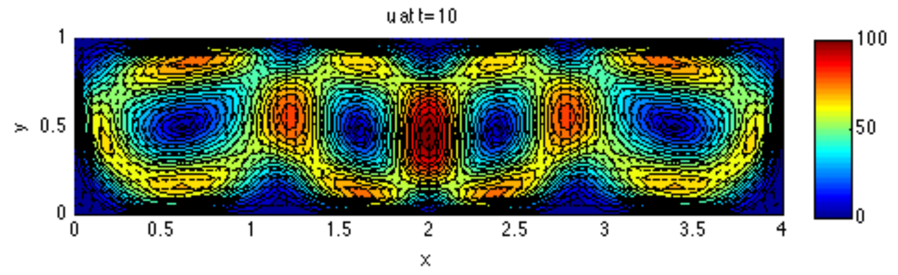
Rayliegh Benard instability– effect of Dissipation – Ra=50000

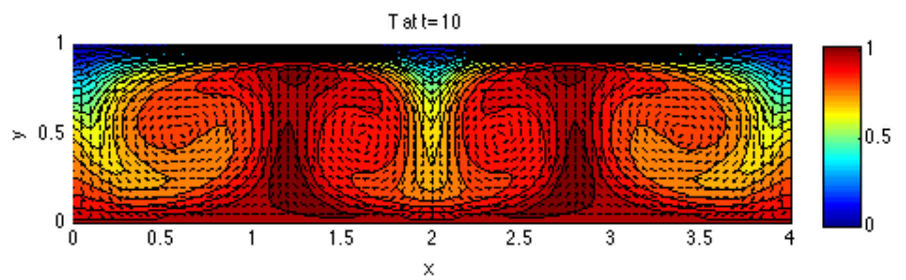
Pr * E = 0



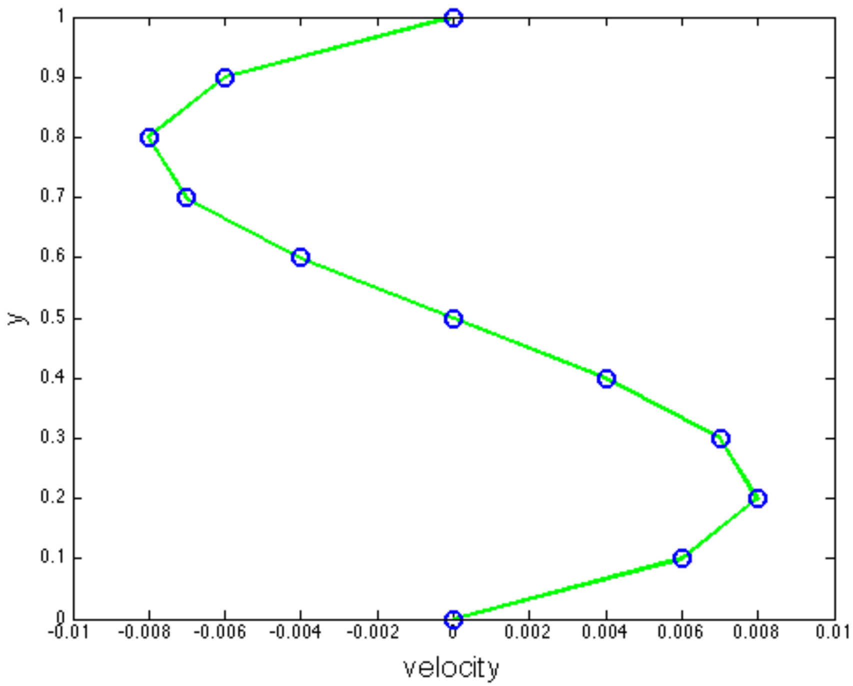


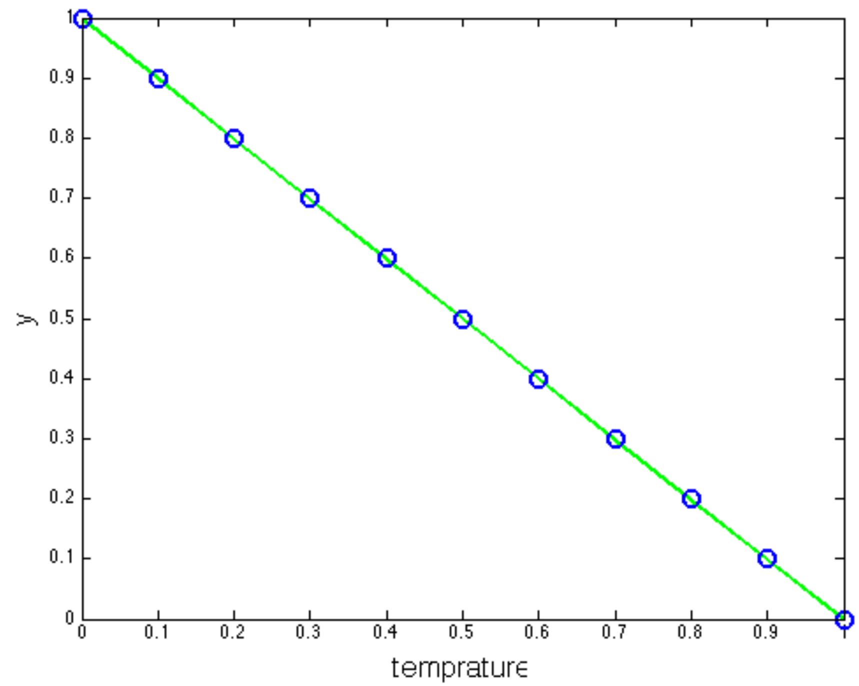
Pr* E = 1e-4



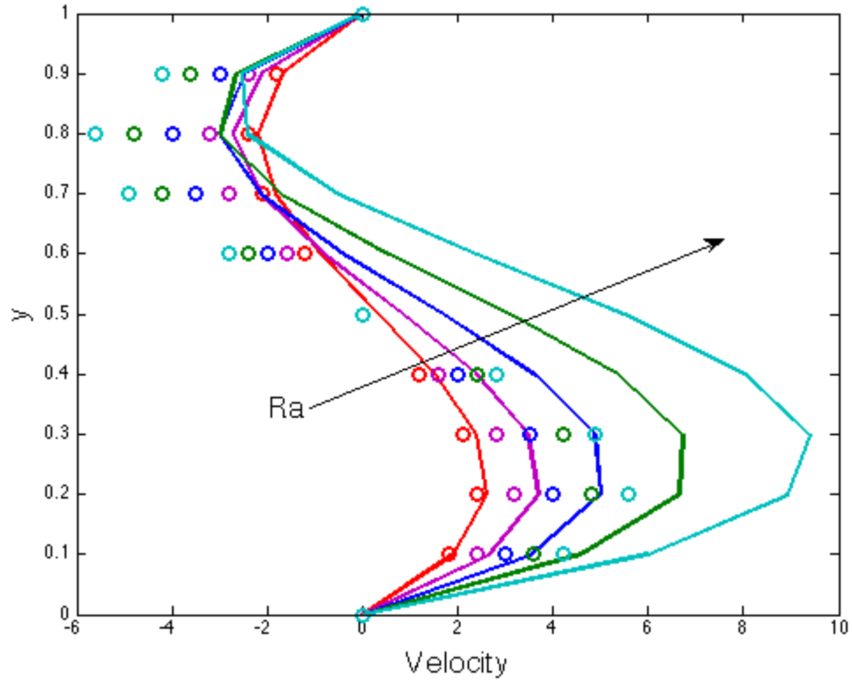


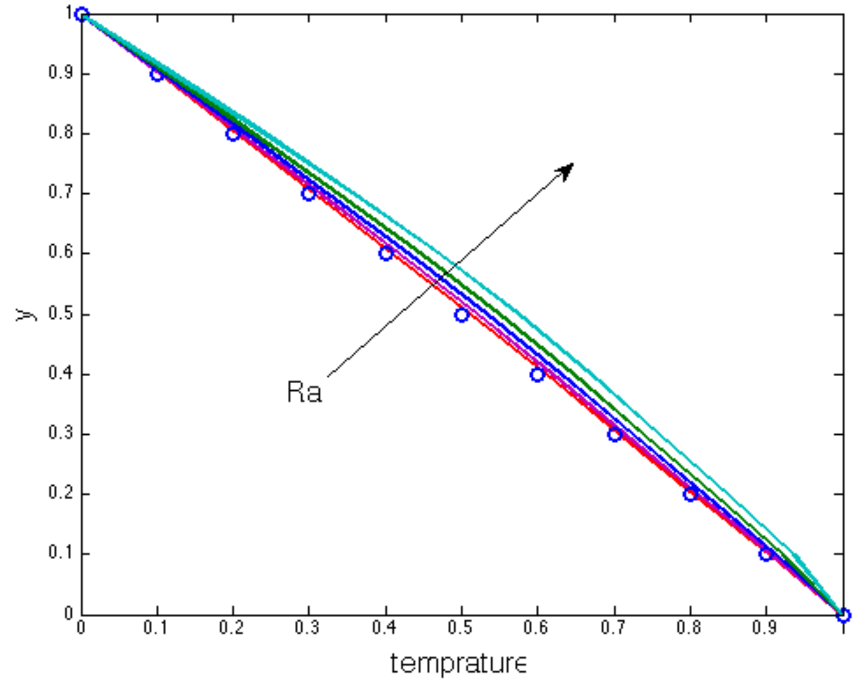
Natural convection in vertical channel flow– circles: analytical solution, lines: N-S solver – Pr=1, Ra =1, E=0



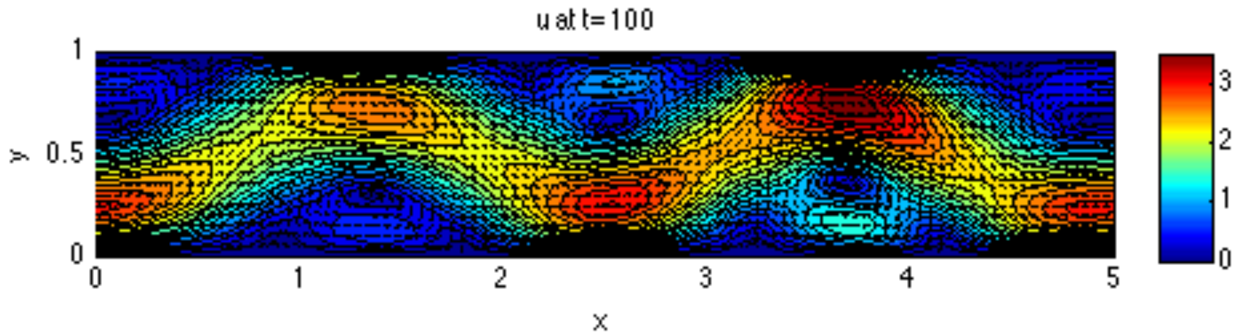


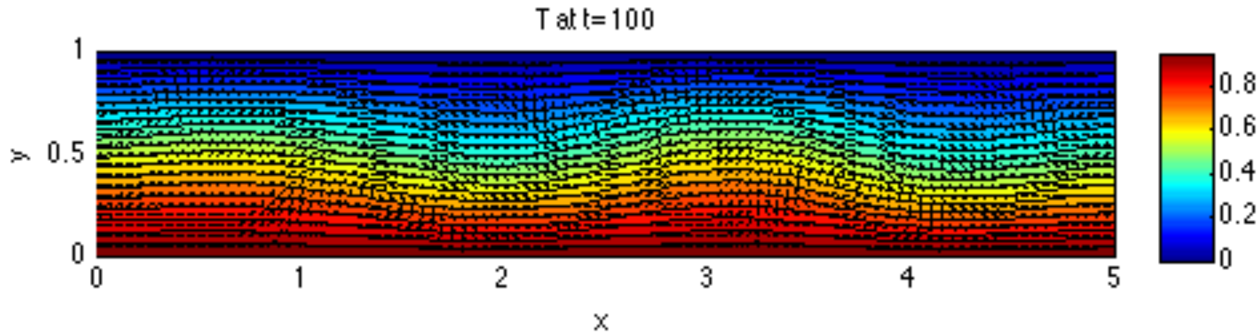
Natural convection in vertical channel flow + dissipation – circles: analytical solution, lines: N-S solver – Pr=1, E=1e-3, Ra= [300:700]

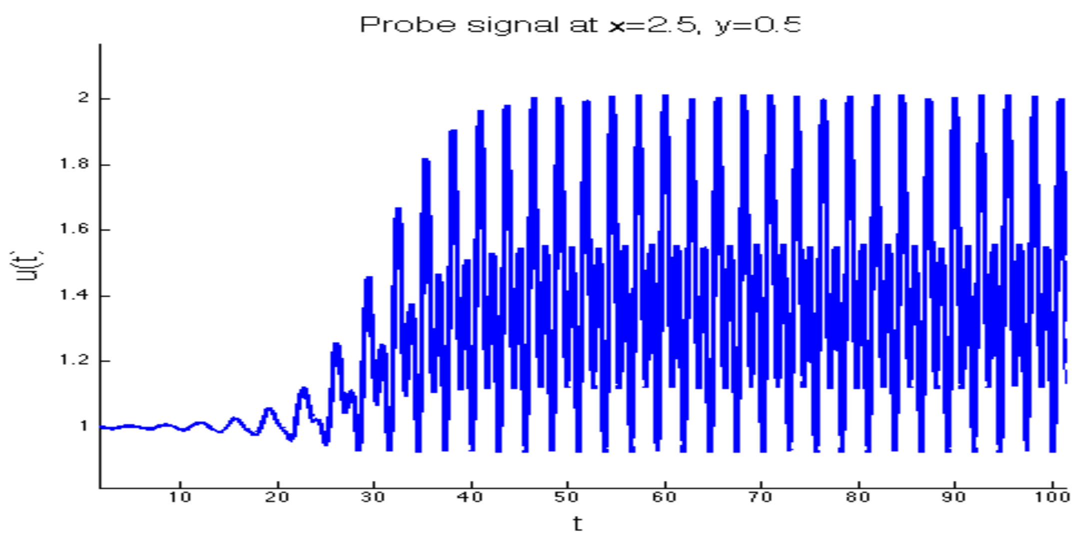




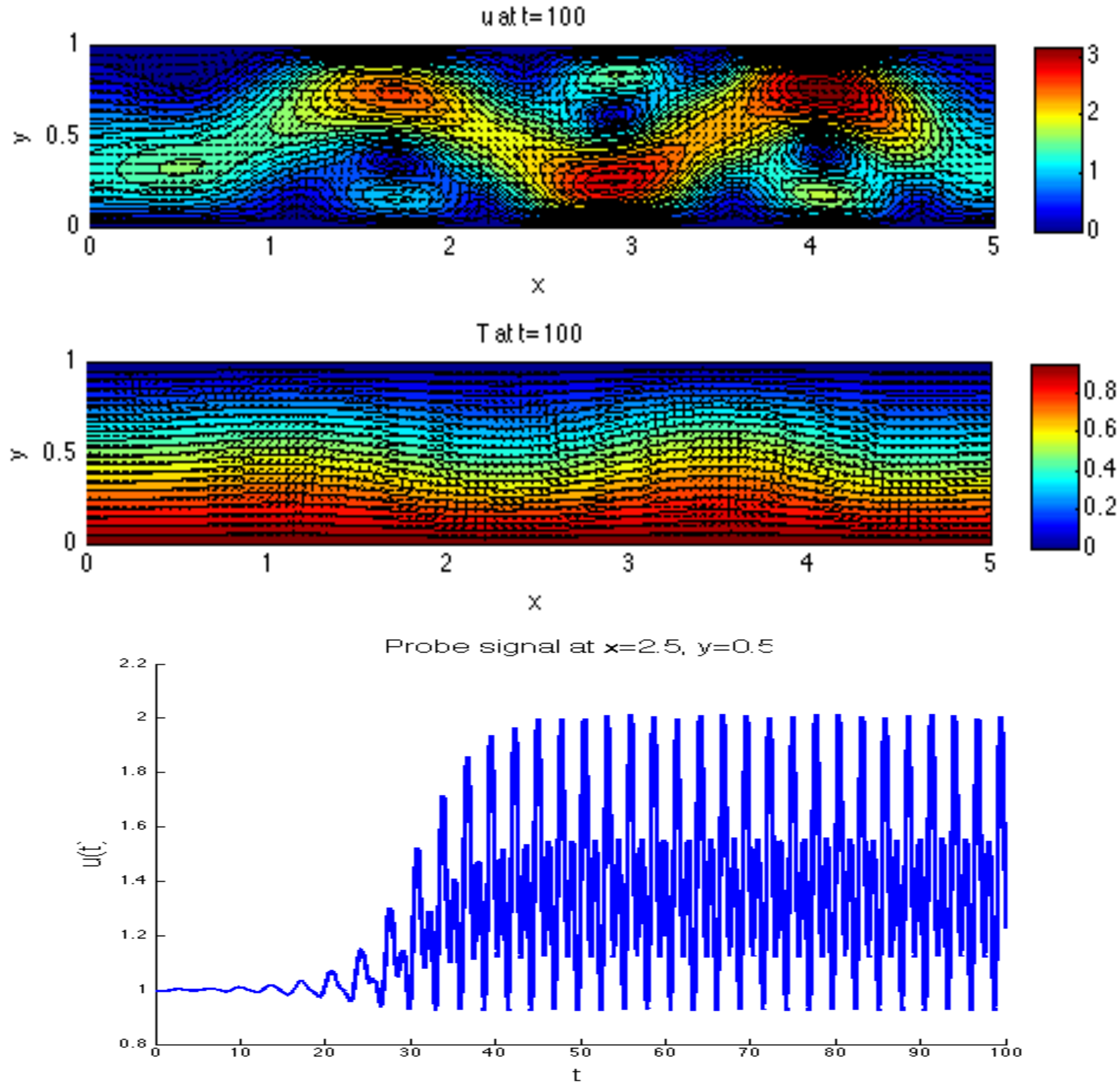
Poiseuille flow and active scalar equation – Re=10, Ra=2000, E =0







Poisuille flow and active scalar equation – Re=10, Ra=2000, E =1e-3



SG2225

- Oral exam by appointment with Luca
  - Project
  - Few theory questions

```
% Parameters for test case : Poiseille flow + Energy equation

  Pr = 0.1;      % Prandtl number
  Ra = 2000      % Rayleigh number
  Re = 1./Pr;    % Reynolds number
  Ri = Ra*Pr;    % Richardson number
  E  = 1e-4;     % Eckert number

  dt = 0.0005;   % time step
  Tf = 100;      % final time
  Lx = 5;        % width of box
  Ly = 1;        % height of box
  Nx = 100;      % number of cells in x
  Ny = 20;       % number of cells in y
  ig = 200;      % number of iterations between output

  % Boundary and initial conditions:
  Utop = 0.;
  Tbottom = 1.;
  Ttop = 0.;
  namp = 0.1; % noise amplitude

%-----------------------------------------
```

```
% Number of iterations
Nit = floor(Tf/dt);

% Spatial grid: Location of corners
x = linspace(0,Lx,Nx+1);  y = linspace(0,Ly,Ny+1);

% Grid spacing
dx = Lx/Nx;   dy = Ly/Ny;

% Boundary conditions:
uN = x*0+Utop;    vN = avg(x,2)*0;            uS = x*0;        vS = avg(x,2)*0;
uW = 4*avg(y,2).*(1-avg(y,2));  vW = y*0;
uE = 4*avg(y,2).*(1-avg(y,2));  vE = y*0;
tN = ones(1,Nx+2)*Ttop;  tS = ones(1,Nx+2)*Tbottom;

% Initial conditions
U = kron(4*ones(Nx-1,1),avg(y,2).*(1-avg(y,2)));  V = zeros(Nx,Ny-1);

% linear profile for T with random noise
T = ones(Nx,Ny)*diag(avg(Ly-y'))*(Tbottom-Ttop)+namp*rand(Nx,Ny);

% Time series
tser = [];   Tser = [];
%----------------------------------------
```

```
%---------------------------------------
% Main loop over iterations

for k = 1:Nit

   % Periodic B.C for side walls (velocities)
   uW = U(end,:);      uE = U(1,:);

   % include all boundary points for u and v (linear extrapolation
   % for ghost cells) into extended array (Ue,Ve)
   Ue = [uW; U; uE ]; Ue = [2*uS'-Ue(:,1) Ue  2*uN'-Ue(:,end)];
   Ve = [vS' V  vN']; Ve = [2*vW-Ve(1,:);Ve;2*vE-Ve(end,:)];

   % averaged (Ua,Va) and differentiated (Ud,Vd) of u and v on corners
   Ua = avg(Ue,2); Ud = diff(Ue,1,2)/2;
   Va = avg(Ve  ); Vd = diff(Ve    )/2;

   % construct individual parts of nonlinear terms
   dUVdx = diff( Ua.*Va    )/dx;
   dUVdy = diff( Ua.*Va,1,2)/dy;
   Ub    = avg( Ue(:,2:end-1)  );      Vb    = avg( Ve(2:end-1,:),2);
   dU2dx = diff( Ub.^2    )/dx;
   dV2dy = diff( Vb.^2,1,2)/dy;
```

```
% treat viscosity explicitly
  viscu = diff( Ue(:,2:end-1),2   )/dx^2 + ...
          diff( Ue(2:end-1,:),2,2 )/dy^2;
  viscv = diff( Ve(2:end-1,:),2,2 )/dy^2 + ...
          diff( Ve(:,2:end-1),2   )/dx^2;

  % Compute force terms
  fx = 8*ones(Nx-1,Ny)/Re;
  fy = Ri*avg(T-Ttop,2);

  % Add force terms
  U = U + dt/Re*viscu - dt*(dUVdy(2:end-1,:)+dU2dx) + dt*fx;
  V = V + dt/Re*viscv - dt*(dUVdx(:,2:end-1)+dV2dy) + dt*fy;

  % Update periodic B.C for side walls (velocities)
  uW = U(end,:);
  uE = U(1,:);

  % pressure correction, Dirichlet P=0 at (1,1)
  rhs = diff( [uW;U;uE] )/dx + diff( [vS' V vN'],1,2 )/dy;
  rhs = reshape(rhs,Nx*Ny,1);
  rhs(1) = 0;
  P = Lp\rhs;
  P = reshape(P,Nx,Ny);
```

```
 % apply pressure correction
  U = U - diff(P    )/dx;
  V = V - diff(P,1,2)/dy;

  % Calculate Dissipation
  dUdx = diff(Ue(:,2:end-1))/dx;
  dUdy = diff(avg(Ua,1),1,2)/dy;
  dVdx = diff(avg(Va,2))/dx;
  dVdy = diff(Ve(2:end-1,:),1,2)/dy;
  Phi = (E/Re)*[ 2*dUdx.^2 + (dUdy+dVdx).^2 + 2*dVdy.^2];

  % Temperature equation
  Te = [T(1,:); T ;T(end,:)]; % adiabatic side-walls
  Te = [2*tS'-Te(:,1) Te 2*tN'-Te(:,end)];
  Tu = 0.5*(Te(1:end-1,:) + Te(2:end, :)).*Ue;
  Tv = 0.5*(Te(:,1:end-1) + Te(:,2:end) ).*Ve;
  H = -(diff(Tu(:,2:end-1))/dx + diff(Tv(2:end-1,:),1,2)/dy);
  H = H + (1/(Pr*Re))*( diff(Te(:,2:end-1),2  )/dx^2 + ...
                    diff(Te(2:end-1,:),2,2)/dy^2 );
  H = H + Phi;
  T = T + H*dt;
end
  %--------------------------------------
```