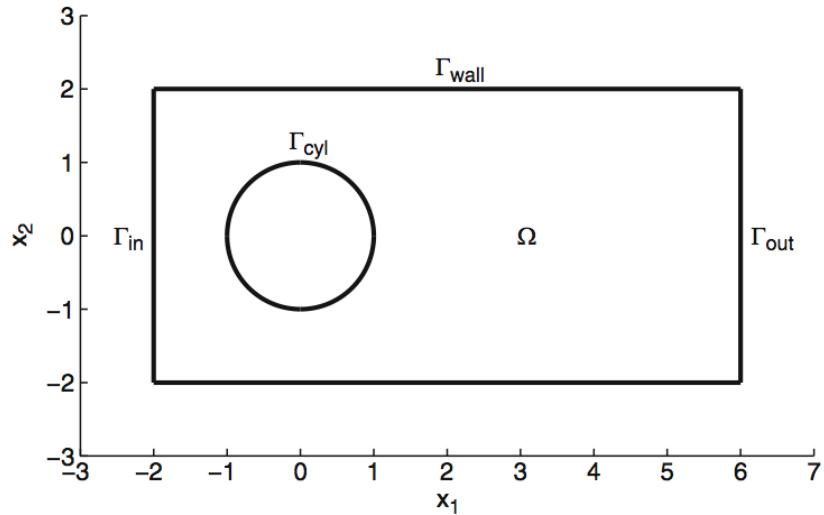


## Exercise 5 – Solutions

**5.1.** We want to solve the **heat transfer in a fluid flow** by the Galerkin least squares (GLS) finite element approximation. The strong form of the (scalar) equation is

$$-\nu \Delta u(x, y) + \mathbf{w}(x, y) \cdot \mathbf{grad} u(x, y) = f(x, y), \quad (x, y) \in \Omega, \quad (1)$$

where  $\Omega = \{(-2, 6) \times (2, 2) \setminus \{x^2 + y^2 \leq 1\}\}$ , see the graphics below.



We set  $f(x, y) = 0$ . The boundary conditions are

$$\begin{aligned} u &= 0, && \text{on } \partial\Omega_{\text{in}}, \\ u &= 1, && \text{on } \partial\Omega_{\text{cyl}}, \\ -\nu \mathbf{n} \cdot \mathbf{grad} u &= 0, && \text{on } \partial\Omega_{\text{out}}, \\ \mathbf{n} \cdot (-\nu \mathbf{grad} u + \mathbf{w} u) &= 0, && \text{on } \partial\Omega_{\text{wall}}. \end{aligned} \quad (2)$$

We set the diffusion coefficient  $\nu = 0.01$ . The wind is given by

$$\mathbf{w} = U_\infty \begin{pmatrix} 1 - \frac{x_1^2 - x_2^2}{(x_1^2 + x_2^2)^2} \\ \frac{-2x_1x_2}{(x_1^2 + x_2^2)^2} \end{pmatrix}, \quad U_\infty = 1.$$

We define

$$\mathcal{H}^1(\Omega) = \left\{ u \in L_2(\Omega) \mid \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \in L_2(\Omega) \right\}.$$

$$\mathcal{H}_E^1(\Omega) = \left\{ u \in \mathcal{H}^1(\Omega) \mid u = 0 \text{ on } \partial\Omega_{\text{in}}, u = 1 \text{ on } \partial\Omega_{\text{cyl}} \right\}$$

$$\mathcal{H}_{E_0}^1(\Omega) = \left\{ u \in \mathcal{H}^1(\Omega) \mid u = 0 \text{ on } \partial\Omega_{\text{in}}, u = 0 \text{ on } \partial\Omega_{\text{cyl}} \right\}$$

Multiplying the left of equation (1) by a test function  $v \in \mathcal{H}_{E_0}^1(\Omega)$  and integrating by parts both the diffusive and convective terms gives

$$\begin{aligned}
& \int_{\Omega} (-\nu \Delta u + \mathbf{w} \cdot \mathbf{grad} u) v \, dx dy \\
&= \int_{\Omega} \nu \mathbf{grad} u \cdot \mathbf{grad} v \, dx dy - \int_{\partial\Omega} \nu \mathbf{n} \cdot \mathbf{grad} u v \, ds - \int_{\Omega} u \mathbf{w} \cdot \mathbf{grad} v \, dx dy + \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{w} u v \, ds \\
&= \int_{\Omega} \nu \mathbf{grad} u \cdot \mathbf{grad} v \, dx dy - \int_{\Omega} u \mathbf{w} \cdot \mathbf{grad} v \, dx dy + \int_{\partial\Omega} \mathbf{n} \cdot (-\nu \mathbf{grad} u + \mathbf{w} u) v \, ds \\
&= \int_{\Omega} \nu \mathbf{grad} u \cdot \mathbf{grad} v \, dx dy - \int_{\Omega} u \mathbf{w} \cdot \mathbf{grad} v \, dx dy + \int_{\partial\Omega_{\text{out}}} \mathbf{n} \cdot \mathbf{w} u v \, ds.
\end{aligned}$$

This is true because  $v = 0$  on  $\partial\Omega_{\text{in}} \cup \partial\Omega_{\text{cyl}}$  and  $\mathbf{n} \cdot (-\nu \mathbf{grad} u + \mathbf{w} u) = 0$  on  $\partial\Omega_{\text{wall}}$ .

So, the weak form of (1)–(2) is: Find  $u \in \mathcal{H}_E^1(\Omega)$  such that

$$\int_{\Omega} \nu \mathbf{grad} u \cdot \mathbf{grad} v \, dx dy + \int_{\Omega} u \mathbf{w} \cdot \mathbf{grad} v \, dx dy + \int_{\partial\Omega_{\text{out}}} \mathbf{n} \cdot \mathbf{w} u v \, ds = \int_{\Omega} f v \, dx dy \quad (3)$$

for all  $v \in \mathcal{H}_{E_0}^1(\Omega)$ .

We use the functionality in `HeatFlowSolver2D.m` that was provided on the webpage and is described in the book by Larson & Bengzon. Remember that  $f = 0$ .

The domain integrals are handled by `assema` and `ConvectionAssembler2D`. For the line integral on  $\partial\Omega_{\text{out}}$  we notice that  $\mathbf{n} = [1, 0]^T$ .

So, in `HeatFlowSolver2D` we replace

```
[R, r] = RobinAssembler2D(p, e, @Kappa2, @gD2, @gN2); % Robin BC
```

by

```
[R] = OutFlowBoundary2D(p, e); % Robin BC at outflow
```

where

```

function [R] = OutFlowBoundary2D(p, e)
    np = size(p, 2); % number of nodes
    ne = size(e, 2); % number of boundary edges
    R = sparse(np, np); % allocate boundary matrix
    for E = 1:ne
        loc2g1b = e(1:2, E); % boundary nodes
        x = p(1, loc2g1b); % node x-coordinates
        y = p(2, loc2g1b); % node y-
        
        if x(1) > 5.999 & x(2) > 5.999, % outflow boundary edge
            len = sqrt((x(1)-x(2))^2+(y(1)-y(2))^2); % edge length
            xc = mean(x); yc = mean(y); % edge mid-point
            [bx, ~] = FlowField(xc, yc);
            RE = bx/6*[2 1; 1 2]*len; % edge boundary matrix
            R(loc2g1b, loc2g1b) = R(loc2g1b, loc2g1b) + RE;
        end
    end
end

```

The main function is modified as follows,

```
function [A,r] = myHeatFlowSolver2D()
    channel=RectCircg(); % channel geometry
    epsilon=0.01; % diffusion parameter
    h=0.1; % mesh size
    [p,e,t]=initmesh(channel,'hmax',h); % create mesh
    pdemesh(p,e,t)
    pause

    A=assema(p,t,1,0,0); % stiffness matrix
    x=p(1,:); y=p(2,:);
    [bx,by] = FlowField(x,y); % evaluate vector field b
    C = ConvectionAssembler2D(p,t,bx,by); % convection matrix
    Sd = SDAssembler2D(p,t,bx,by); % GLS stabilization matrix
    [R] = OutFlowBoundary2D(p,e); % Robin BC
    delta = h; % stabilization parameter
    A = epsilon*A - C' + R; %% + delta*Sd;
    np = size(p,2); % enforce zero Dirichlet BC
    inflow = find(p(1,:)<-1.999);
    cylinder = find((p(1,:).^2 + p(2,:).^2) < 1.001);
    free = setdiff([1:np],[inflow, cylinder]);
    r = -A(free,cylinder)*ones(length(cylinder),1);
    A = A(free,free);
    u = zeros(np,1); u(cylinder) = 1;
    u(free) = A\r; % solve linear system
    pdeplot(p,[],t,'xydata',u,'xystyle','off','contour',...
    'on','levels',15,'colorbar','on','colormap','hot')
    axis([-2 6 -2 2])
    axis equal
end
```

**5.2** Uncomment `+ delta*Sd;` on line 16 of `myHeatFlowSolver2D`.