Lab Report - 3

2D Steady Transport problems

- Sanath Keshav

1 GLS Implementation

```
elseif method == 3
    % GLS
    Ke = Ke + (nu*(Nx'*Nx+Ny'*Ny) + N_ig'*(ax*Nx+ay*Ny) + sig*(N_ig'*N_ig) + ...
        tau*(ax*Nx+ay*Ny - nu*(Nxx+Nyy) + sig*(N_ig))'*(ax*Nx+ay*Ny - nu*(Nxx+Nyy) + sig*(N_ig)))*dvolu;
    aux = N_ig*Xe;
    f_ig = SourceTerm(aux);
    fe = fe + (N_ig+tau*(ax*Nx+ay*Ny + nu*(Nxx+Nyy)))'*(f_ig*dvolu);

end
```

The GLS stabilization was implemented and verified for various test cases.

2 Quadratic Triangles and Quadratic Quadrilaterals

```
if elem == 0
    if p == 1
        ngaus = 4;
        Xe_ref = [-1,-1; 1,-1; 1,1; -1,1];
    elseif p == 2
        ngaus = 9;
        Xe_ref =  [-1,-1; 1,-1; 1,1; -1,1; ...
            0,-1; 1,0; 0,1; -1,0; 0,0];
    else
        error('not avilable interpolation degree');
    end
elseif elem == 1
    if p == 1
        ngaus = 3;
        Xe_ref = [0,0; 1,0; 0,1];
    elseif p == 2
        ngaus = 6;
        Xe_ref = [0,0; 1,0; 0,1; 0.5,0; 0.5,0.5; 0,0.5];
    else
        error('not avilable interpolation degree');
    end
else
    error('unavailable element')
end
```

First the iso-parametric elements were implemented into the reference elements.

The corresponding linear and quadratic shape functions and their first and second derivatives in each direction is coded into the function for triangular and quadrilateral elements.

```
if elem == 0
    if p == 1
        N    = [(1-xi).*(1-eta)/4, (1+xi).*(1-eta)/4, ...
            (1+xi).*(1+eta)/4, (1-xi).*(1+eta)/4];
        Nxi  = [(eta-1)/4, (1-eta)/4, (1+eta)/4, -(1+eta)/4];
        Neta = [(xi-1)/4, -(1+xi)/4,   (1+xi)/4,  (1-xi)/4 ];
        N2xi = [0.*eta, 0.*eta, 0.*eta, 0.*eta];
        N2eta = [0.*xi, 0.*xi, 0.*xi, 0.*xi];
    elseif p == 2
        N    = [xi.*(xi-1).*eta.*(eta-1)/4, xi.*(xi+1).*eta.*(eta-1)/4, ...
            xi.*(xi+1).*eta.*(eta+1)/4, xi.*(xi-1).*eta.*(eta+1)/4, ...
            (1-xi.^2).*eta.*(eta-1)/2,  xi.*(xi+1).*(1-eta.^2)/2,   ...
            (1-xi.^2).*eta.*(eta+1)/2,  xi.*(xi-1).*(1-eta.^2)/2,   ...
            (1-xi.^2).*(1-eta.^2)];
        Nxi  = [(xi-1/2).*eta.*(eta-1)/2,   (xi+1/2).*eta.*(eta-1)/2, ...
            (xi+1/2).*eta.*(eta+1)/2,   (xi-1/2).*eta.*(eta+1)/2, ...
            -xi.*eta.*(eta-1),          (xi+1/2).*(1-eta.^2),   ...
            -xi.*eta.*(eta+1),          (xi-1/2).*(1-eta.^2),   ...
            -2*xi.*(1-eta.^2)];
        Neta = [xi.*(xi-1).*(eta-1/2)/2,    xi.*(xi+1).*(eta-1/2)/2, ...
            xi.*(xi+1).*(eta+1/2)/2,    xi.*(xi-1).*(eta+1/2)/2, ...
            (1-xi.^2).*(eta-1/2),       xi.*(xi+1).*(-eta),    ...
            (1-xi.^2).*(eta+1/2),       xi.*(xi-1).*(-eta),    ...
            (1-xi.^2).*(-2*eta)];
        N2xi = [ (eta.*(eta - 1))/2, (eta.*(eta - 1))/2, ...
            (eta.*(eta + 1))/2, (eta.*(eta + 1))/2, ...
            -eta.*(eta - 1), 1 - eta.^2, ...
            -eta.*(eta + 1), 1 - eta.^2, ...
            2*eta.^2 - 2];
        N2eta = [ (xi.*(xi - 1))/2, (xi.*(xi + 1))/2, ...
            (xi.*(xi + 1))/2, (xi.*(xi - 1))/2, ...
            1 - xi.^2, -xi.*(xi + 1), ...
            1 - xi.^2, -xi.*(xi - 1), ...
            2*xi.^2 - 2];
```

Q1 and Q2 Shape functions

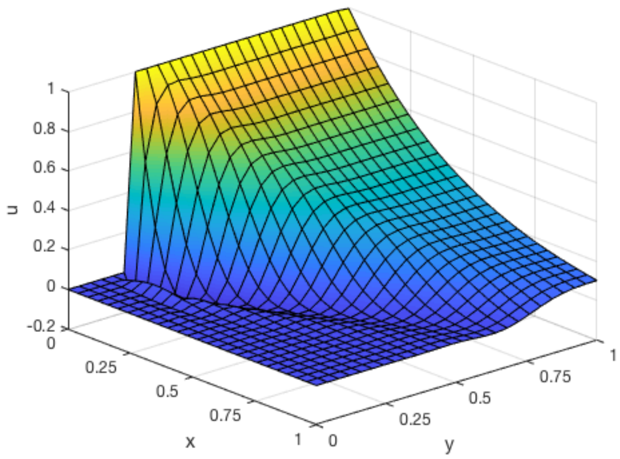
```
elseif elem == 1
    if p == 1
        N = [1-xi-eta, xi, eta];
        Nxi = [-ones(size(xi)), ones(size(xi)), zeros(size(xi))];
        Neta = [-ones(size(xi)), zeros(size(xi)), ones(size(xi))];
        N2xi = [0.*eta, 0.*eta, 0.*eta];
        N2eta = [0.*xi, 0.*xi, 0.*xi];
    elseif p == 2
        N = [xi.*(2*xi - 1), eta.*(2*eta - 1), ...
            (1-xi-eta).*(2*(1-xi-eta) - 1), 4.*xi.*eta ...
            4*eta.*(1-xi-eta) , 4*xi.*(1-xi-eta)];
        Nxi = [ 4.*xi - 1, 0*xi, 4.*eta + 4.*xi - 3, 4.*eta, -4.*eta, 4 - 8.*xi - 4.*eta];
        Neta = [ 0*xi, 4*eta - 1, 4*eta + 4*xi - 3, 4*xi, 4 - 4*xi - 8*eta, -4*xi];
        N2xi = [ 4*ones(size(xi)), 0*ones(size(xi)), 4*ones(size(xi)),...
            0*ones(size(xi)), 0*ones(size(xi)), -8*ones(size(xi))];
        N2eta = [ 0*ones(size(xi)), 4*ones(size(xi)), 4*ones(size(xi)),...
            0*ones(size(xi)), -8*ones(size(xi)), 0*ones(size(xi))];
```

P1 and P2 Shape functions

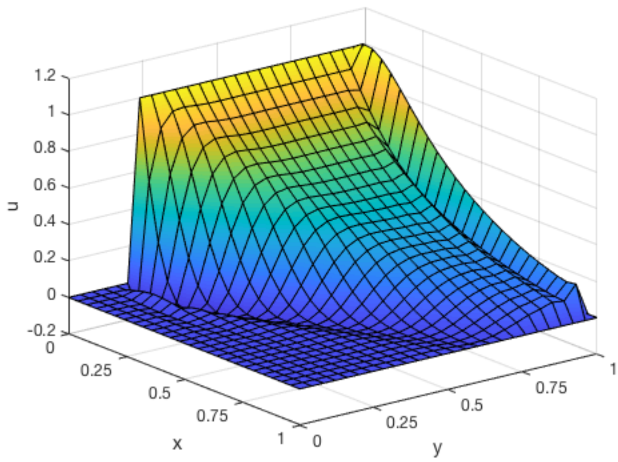
3 Numerical results

Case 1: $\quad ||a|| = \dfrac{1}{2}\ ,\ \nu = 10^{-4}\ ,\ \sigma = 1$

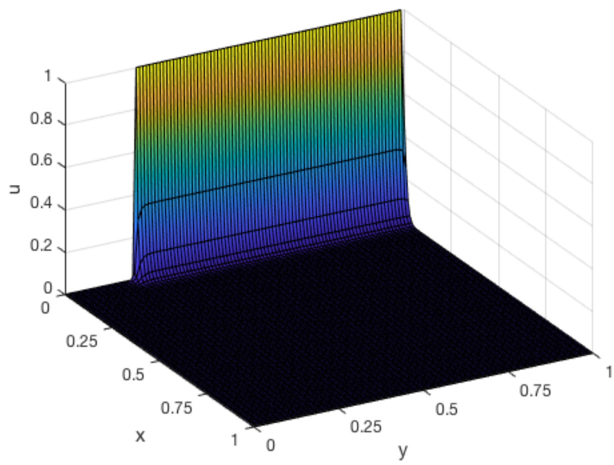
Case 2: $\quad ||a|| = 10^{-3}\ ,\ \nu = 10^{-4}\ ,\ \sigma = 1$



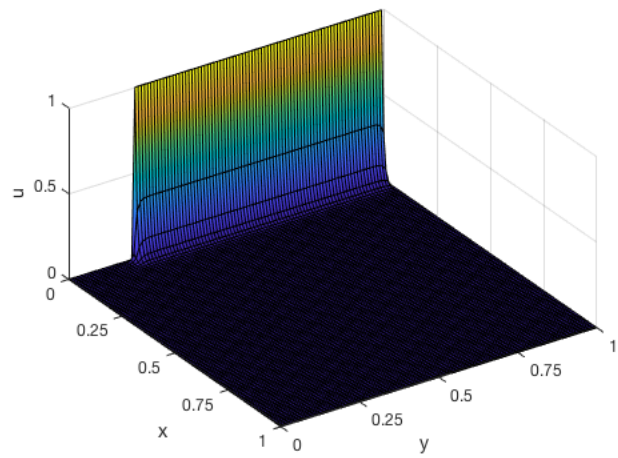
Case 1: Neumann BCs



Case 1: Dirichlet BCs



Case 2: Neumann BCs



Case 2: Dirichlet BCs

The difference between the application of neumann bcs and dirichlet bcs can be observed better in case 1 where the situation is convection reaction dominated case rather than the reaction dominated case 2. With the application of neumann bcs on the outflow boundary, smooth solutions are obtained where as with the dirichlet case and abrupt discontinuous solution is observed somewhat similar to the behavior of a boundary layer. In case 2, since the convection velocity and the diffusion coefficient if very low the front does not move much.