

Problem description

This second part of the report is aimed at analyzing the methods used to solve a steady Navier-Stokes problem which describes general viscous steady Newtonian flows. A constraint to the Navier-Stokes problem is created through the assumption of incompressibility of the fluid. Solving such problem using finite elements provide some difficulties. This is due to the fact that pressure is acting as a Lagrange multiplier. Another difficulty in solving the Navier-Stokes problem is the nonlinear convective. This term causes instability in the solution. In addition, the nonlinearity of the convective term requires iterative methods in order to solve the problem. Such iterative methods are the Picard and Newton-Raphson methods. Both methods are applied and compared in this report.

The problem in hand is described as a one*one square domain. The boundary conditions (BC) are as follows: three sides are fixed, while the upper side has a prescribed velocity of magnitude one in the x-direction. The pressure is set to zero at the lower left corner. The described BCs induce a discontinuity at the upper two corners. This results in a singularity in the pressure solution.

The problem is governed by the following differential equation:

$$\begin{aligned} -\nu \nabla^2 \mathbf{v} + (\mathbf{v} \cdot \nabla) \mathbf{v} + \nabla p &= \mathbf{f} && \text{in } \Omega \\ \nabla \cdot \mathbf{v} &= 0 && \text{in } \Omega \\ \mathbf{v} &= \mathbf{v}_D && \text{on } \partial\Omega \end{aligned}$$

Equation 1 Governing differential equation

The Galerkin discretization leads to the following matrix form:

$$\begin{pmatrix} \mathbf{K} + \mathbf{C}(\mathbf{v}) & \mathbf{G}^T \\ \mathbf{G} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}$$

Equation 2 Matrix form of the system to be solved

Where:

$$c(\mathbf{w}, \mathbf{v}, \mathbf{a}) := \int_{\Omega} \mathbf{w} \cdot (\mathbf{a} \cdot \nabla) \mathbf{v} d\Omega$$

Equation 3 Convection matrix

It could be seen that the convection matrix is nonlinear in \mathbf{v} . In order to solve such a problem, iterative methods are required.

Picard method

The first method to be implemented is the Picard method. The system of equations to be solved can be written as:

$$\mathbf{A}(\mathbf{x})\mathbf{x} = \mathbf{b}(\mathbf{x})$$

Equation 4 Picard system of equations

An initial guess for the velocity is applied to the solution, then the updated value for the next iteration is calculated as follows:

$$\mathbf{A}(\mathbf{x}^k)\mathbf{x}^{k+1} = \mathbf{b}(\mathbf{x}^k)$$

Equation 5 Updated value of the unknown

Successive iterations are conducted until a satisfactory solution is achieved according to a tolerance specified by the user. The results are obtained as follows:

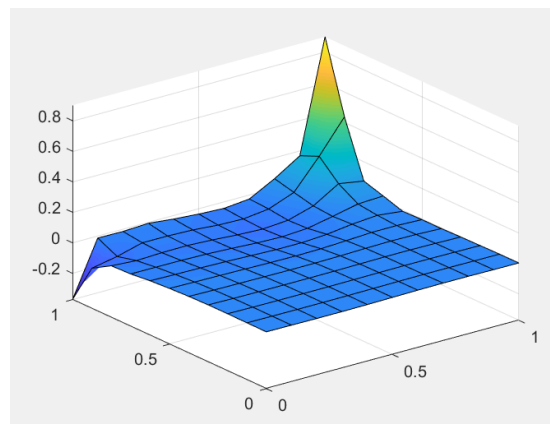


Figure 1 Pressure solution using the Picard method

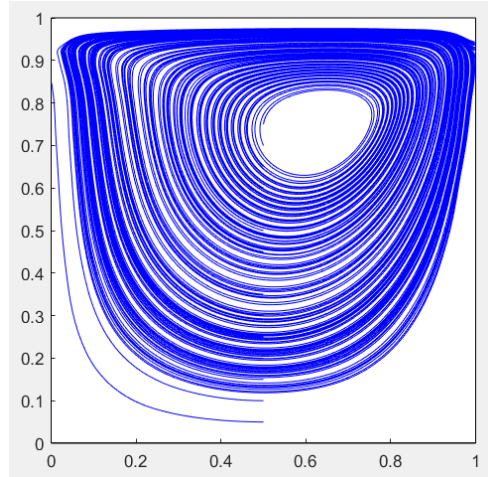


Figure 2 Stream lines solution using the Picard method

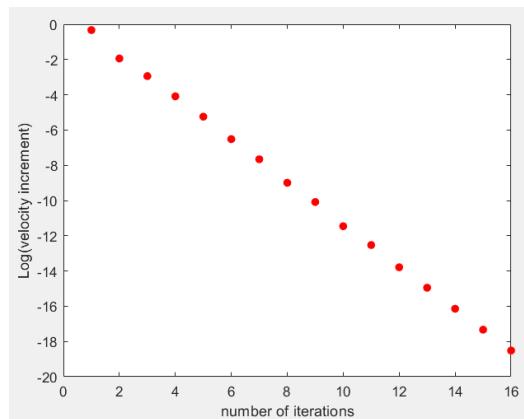


Figure 3 Convergence of the velocity solution using the Picard method

The implementation of the solution using Picard method converged using 16 iterations problem. As expected, this method's convergence follows a linear pattern as shown in Figure 3.

Newton Raphson method

The second method to be implemented is the Newton Raphson method. In order to solve the system matrix (Equation 2) the following iterative system of equations is to be solved:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta \mathbf{x}^{k+1}$$

Equation 6 Iterative scheme

In order to compute the incremental term, the Jacobean and the residual terms are required. The residual term is defined as follows:

$$\mathbf{r} = \begin{bmatrix} (\mathbf{K} + \mathbf{C}(\mathbf{v}))\mathbf{v} + \mathbf{G}^T \mathbf{p} - \mathbf{f} \\ \mathbf{G}\mathbf{v} \end{bmatrix}$$

Equation 7 residual term

The Jacobian is given as follows:

$$\mathbf{J} = \begin{pmatrix} \frac{d\mathbf{r}_1}{d\mathbf{v}} & \mathbf{G}^T \\ \mathbf{G} & \mathbf{0} \end{pmatrix}$$

Equation 8 Jacobian term

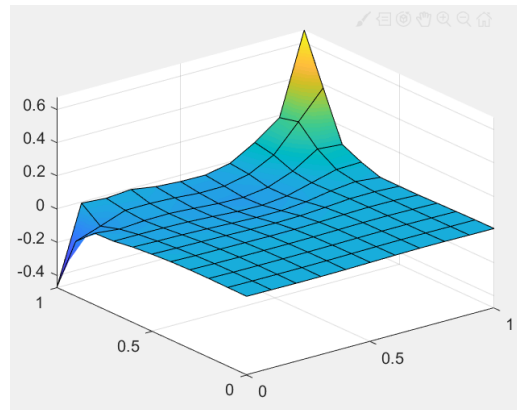


Figure 4 Pressure solution using the Newton Raphson method

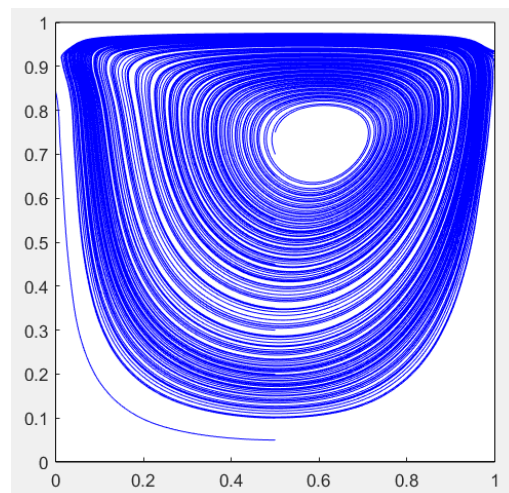


Figure 5 Stream lines solution using the Newton Raphson method

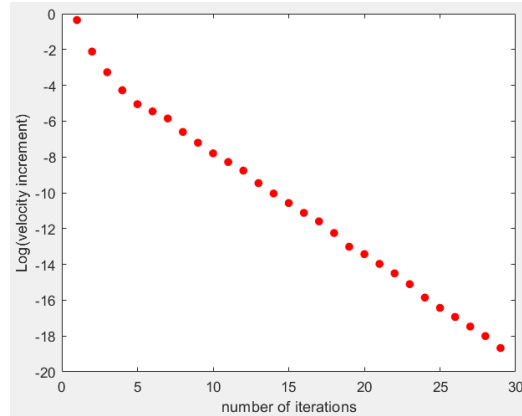


Figure 6 Convergence of the velocity solution using the Newton Raphson method

The implementation of the solution using Newton Raphson method converged using 29 iterations problem. However, the convergence plot (Figure 6) shows an almost linear behavior. The expected behavior is quadratic hence the solution does not agree with the theory. Thus, an error in the implementation of the method might have occurred.

Code

Picard

Initial conditions:

```

velo = zeros(ndofV/2,2);
y2 = dom(4);
nodesY2 = find(abs(X(:,2)-y2) < 1e-6);
velo(nodesY2,1) = 1;

save velocity.mat

```

Modifications to the EleMatStokes function in the StokesSystem function:

```

% Assemble the element matrices
C(Te_dof, Te_dof) = C(Te_dof, Te_dof) + Ce;

%Create the element convection matrix
Ce = Ce + Ngp'*(veloelem(ig,1)*Nx+veloelem(ig,2)*Ny)*dvolu;

```

Newton Raphson

Modifications to the iterative scheme in the mainNavierStokes:

```
C = ConvectionMatrix(X,T,referenceElement,velo);
Cred = C(dofUnk,dofUnk);
velored = veloVect(dofUnk);

%Calculate the residual
r = [(Kred+Cred)*velored+Gred'*pres-fred; Gred*velored];

%Calculate the jacobian
J = [Kred+2*Cred Gred' ; Gred zeros(nunkP)];

%Calculate the velocity increment
solInc = -J\r ;
```