REPORT OF COUETTE FLOW PROBLEM

for

COMPUTATIONAL FLUID DYNAMICS

submitted by

SAURABH KUMAR
234103332

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

(October 2023)

Question:

Solve the following non-dimensional partial differential equation for **Couette Flow** using finite difference method with specified boundary conditions for j=1 to j=101(M) grid size as shown in Fig. 1. **Use pseudo-transient** solution approach **with initial condition of u-velocity as zero**.

$$\frac{\partial u}{\partial t} = \frac{1}{Re_H}\frac{\partial^2 u}{\partial y^2} \qquad Re_H = \frac{UH}{\nu} \qquad \epsilon = \sqrt{\frac{\sum_{j=1,M}(u^{n+1} - u^n)^2}{M}}$$

1. Explicit method : **FTCS**
2. Implicit method
   a) **BTCS**: Point Gauss-Seidel iterative method
   b) **BTCS**: Line Gauss-Seidel iterative method (TriDiagonal Matrix Algorithm)
   c) **Crank-Nicolson**: Line Gauss-Seidel iterative method (TriDiagonal Matrix Algorithm)

Discretize the above PDE using the discretization schemes mentioned above with uniform grid of M. Take Re_H=100 and $\Delta t = 5 \times 10^{-3}$ for Explicit method and $\Delta t = 10^{-2}$ for Implicit method.

Submit results in terms of velocity profiles at different time* including steady state results as shown in Fig. 2, **convergence history** (**ε vs *t***) for all schemes in a single plot and report on discretized algebraic equation of each discretization scheme, comparison study of number of time iterations and physical time taken to converge up to $\epsilon < 10^{-6}$.

***FTCS:** t =0.5, 2.5, 10, 25, converged result and **Crank-Nicolson:** t=1, 5, 10, converged result

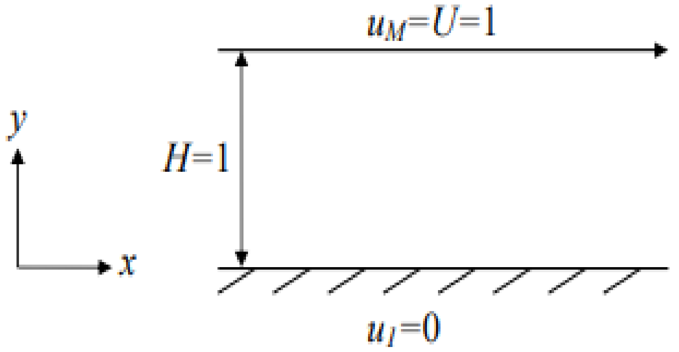


Fig. 1

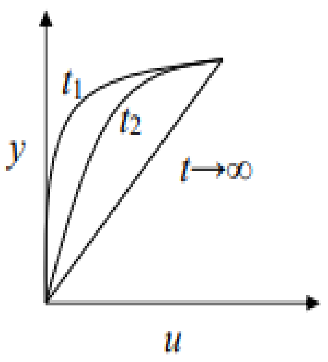


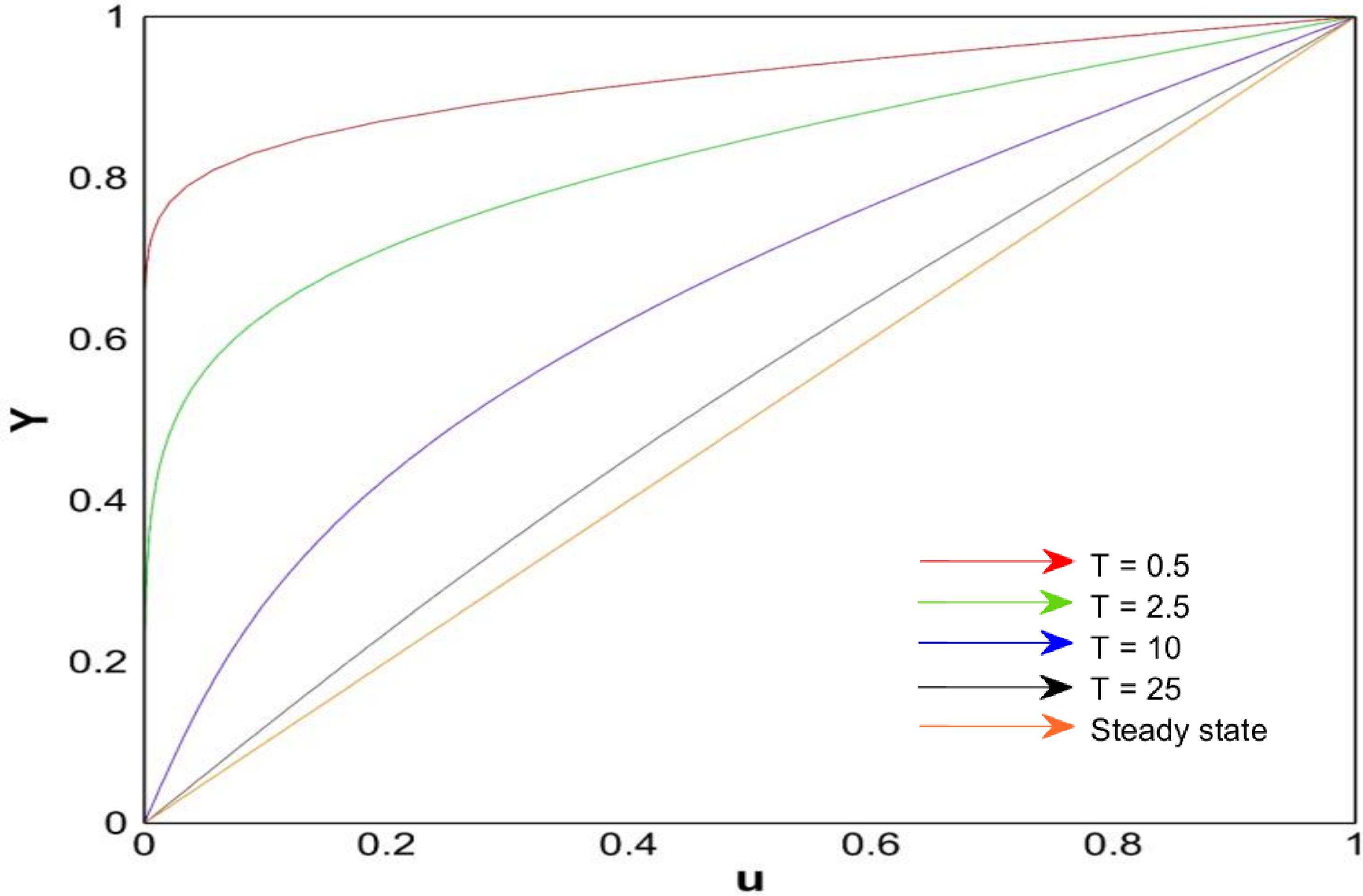
Fig. 2

<u>Solution</u>:

1. <u>Explicit Method</u>: <u>FTCS</u>:

<u>Result</u>:

Velocity profile at different time including steady state

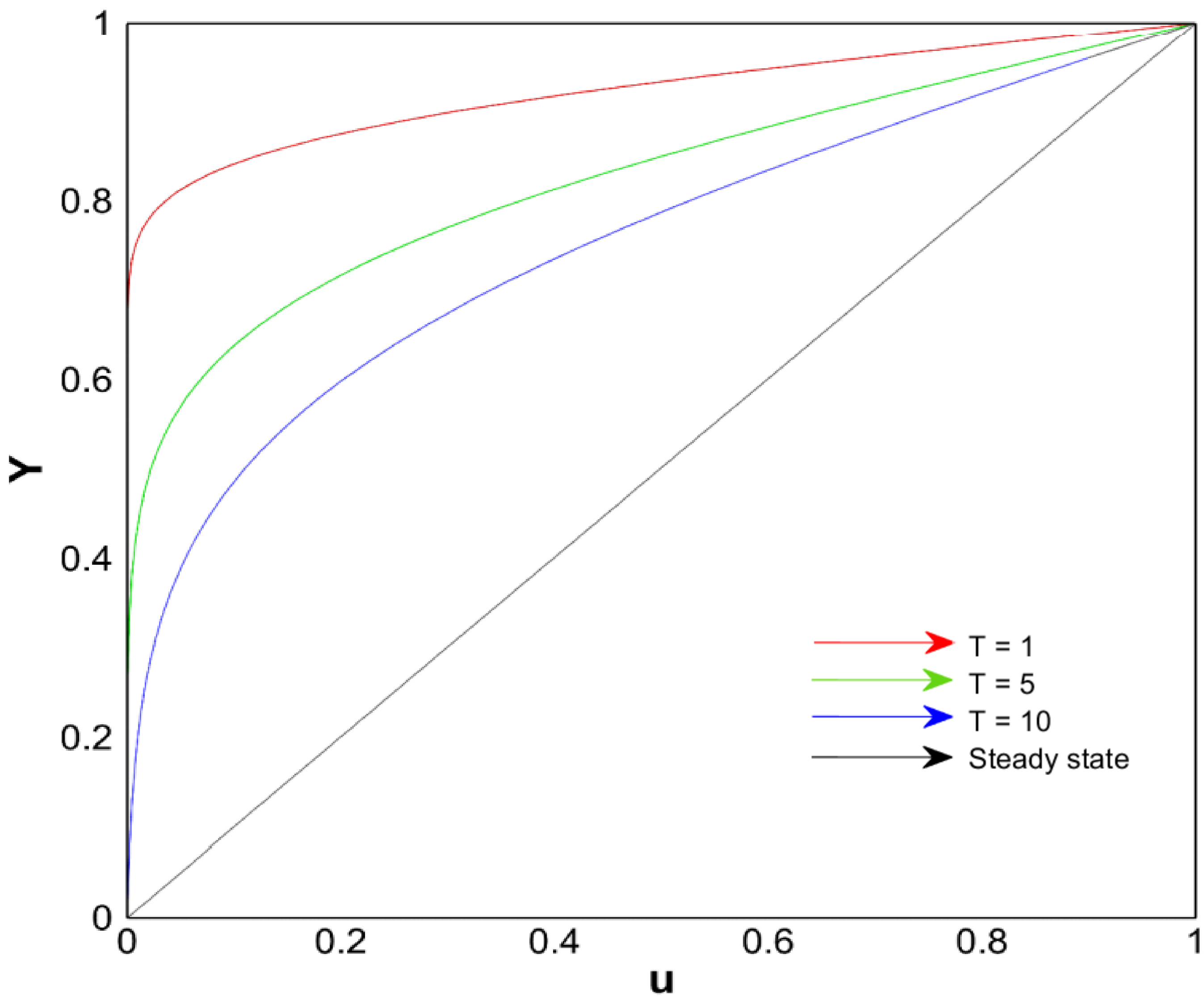


2. **Implicit Method** :

(A) **BTCS Point Gauss Seidel Iterative Method:**

Result:

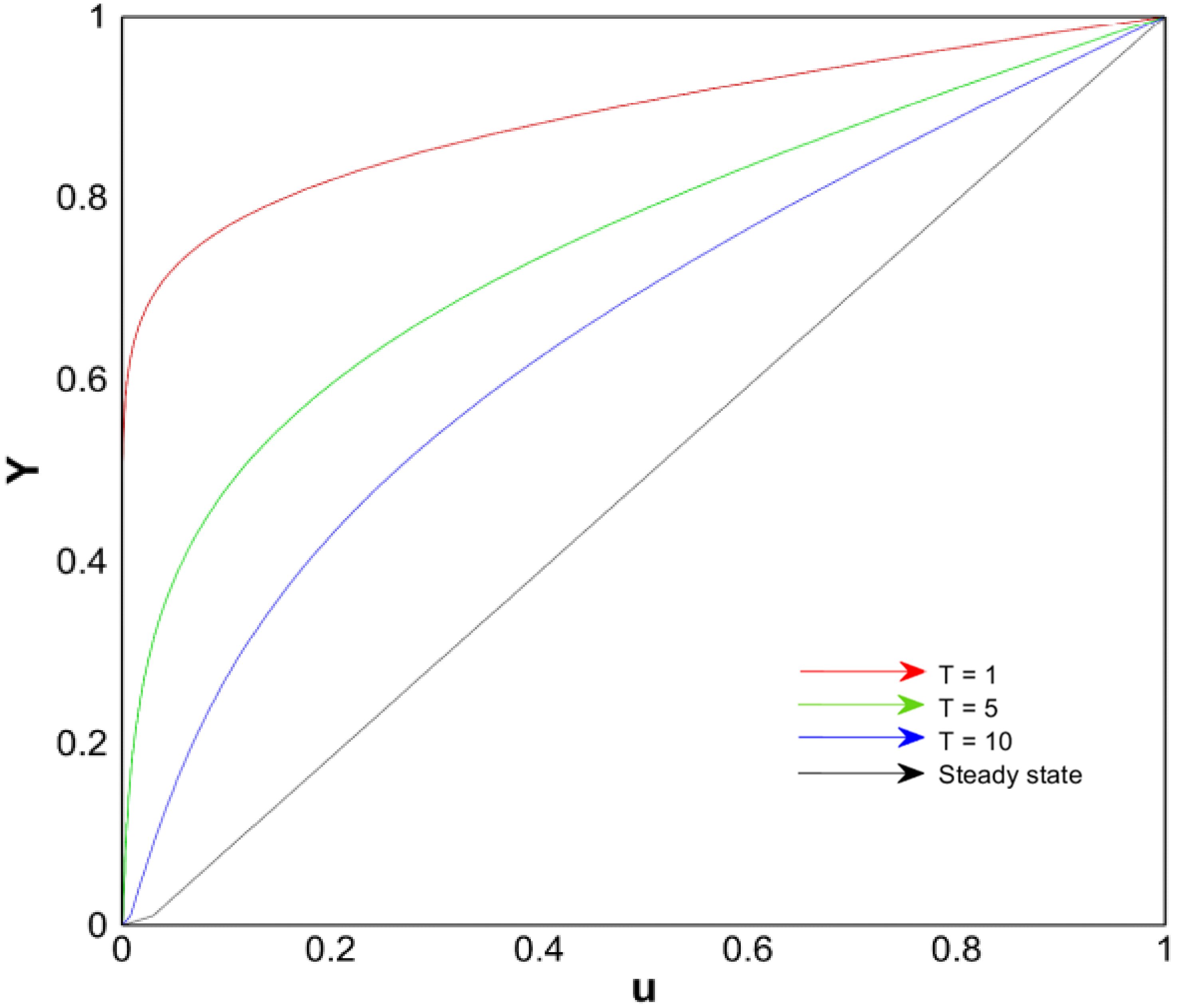
Velocity profile at different time including steady state



(B) BTCS Line Gauss Seidel Iterative Method:

Result:

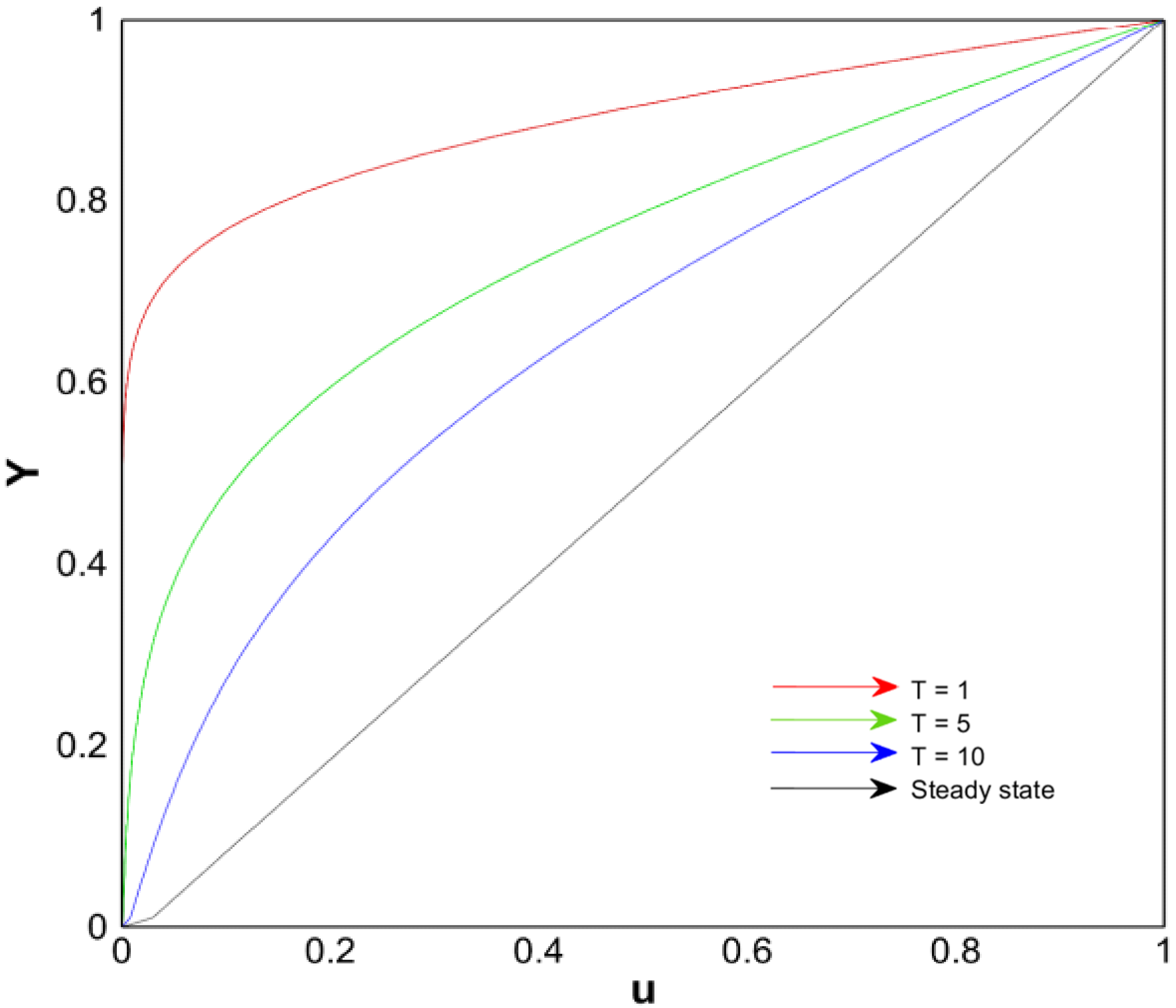
Velocity profile at different time including steady state



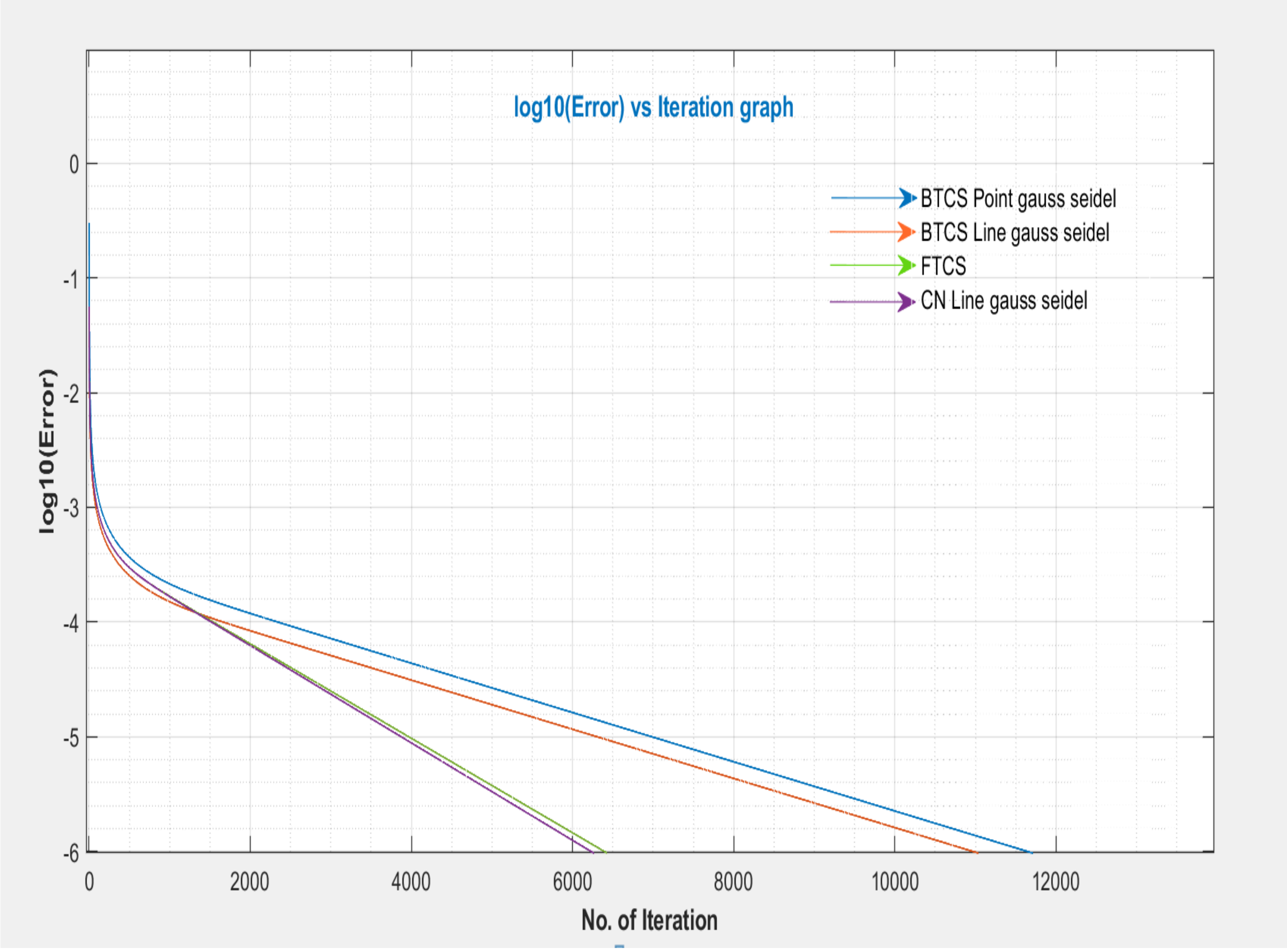
(C) BTCS Crank Nicolson Iterative Method (TDMA):

Result:

Velocity profile at different time including steady state



Comparison plot between Iteration v/s log10($Error$) for all the methods:



Convergence history { time *v/s* log10(E*rror*) } of all Methods in single plot:

Result: