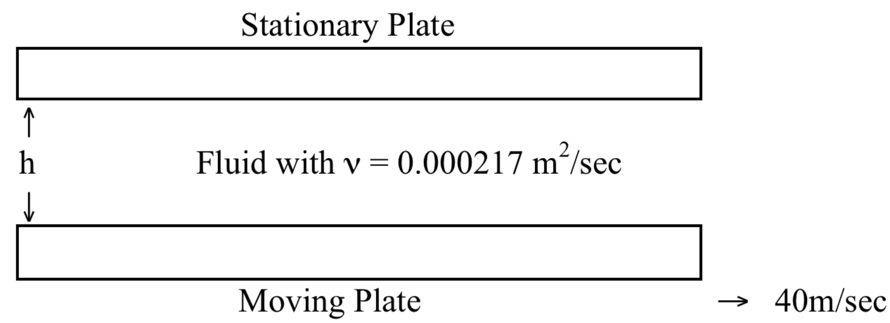
Computational Fluid Dynamics (24-718)

Homework 2, Assigned on: Sept 17, Due on: Sept 26

Submit math formulations, results, and discussion as hard copies, which will be returned after grading. Workable copies of computer codes for all problems should be submitted to the Canvas. Put all files in one zip file, and use following format to name the zip file: $lasename - firstname - hw\#$. For example "$singh - satbir - hw1$".

Problem 1. Consider fluid bounded by two parallel plates extended to infinity such that no end effects are encountered. The lower wall is suddenly accelerated in the x-direction at 40 m/s as shown in the Figure. The spacing between two plates is denoted by h, and it is set to 0.04 m. y-direction is taken perpendicular to the plates such that lower plate lies in xz plane. This problem can be represented by following one-dimensional



(1-D) parabolic equation:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2}$$

where ν is kinematic viscosity of the fluid. For this problem, take ν to be 2.17 $\times 10^{-4}$ m^2/s. Compute the velocity profile $u = u(t, y)$ starting with an initial velocity of, $u(0, 0) = 40 m/s, u(0, y) = 0$ for $0 < y \leq h$, using FTCS explicit method, Laasonen method (FTCS implicit), and Crank-Nicolson method. For FTCS implicit and Crank-Nicolson methods, use MATLAB or C++ built-in function to invert the matrix.

(1) Based on von-Neumann stability analysis, what is the maximum allowable timestep (Δt_{max}) for FTCS explicit method. **(5 points)**

(2) Prove that Crank-Nicolson method is second-order accurate in Δt. **(10 points)**

(3) Provide discretized form of equations on interior nodes of the domain for all methods. **(15 points)**

(4) Provide discretized form of equations for implementation of boundary conditions. **(5 points)**

(5) For each method plot u as a function of y at times of 0.0, 0.18, 0.36, 0.54, 0.72, 0.90, 1.08 for $\Delta y = 0.001, \Delta t = 0.002$. For each method, plot data at all times in the same figure but do not plot multiple methods in the same figure. **(30 points)**

(6) For all methods plot, in the same figure, u as a function of y at time 1.08 for $\Delta y = 0.001, \Delta t = 0.002$. **(5 points)**

(7) Repeat part (5) for $\Delta y = 0.001, \Delta t = 0.0025$. **(10 points)**

(8) For Crank-Nicolson method, invert the matrix using TDMA. Plot , in the same figure, u as a function of y at time 1.08 for $\Delta y = 0.001, \Delta t = 0.002$ for the MATLAB or C++ built-in matrix inversion algorithm and the TDMA algorithm. **(10 points)**

(9) Report, in a bar chart, CPU time taken by each method, including TDMA inversion, for $\Delta y = 0.001, \Delta t = 0.002$. **(10 points)**