Numerical simulation of water flow through a rectangular channel of a screen grid in Water treatment Process

HASSAN FADI MEHDI[1]

[1] Rhein-Waal University of applied sciences, 2009 Marie-Curie-Straße 1, 47533 Kleve, Germany
[2] Faculty of Technology and Bionics, 2009 Marie-Curie-Straße 1, 47533 Kleve, Germany
[3] Department of Mechanical Engineering, 2009 Marie-Curie-Straße 1, 47533 Kleve, Germany
[*] Hassan.Fadi-Mehdi@hsrw.org

Compiled February 28, 2023

Water treatment is crucial for the benefit of humans due to removing the waste particles from water using screens. Simulating fluid through a treatment proces and especially in 3D model is a very complex task, therefore for simplicity i simplified and limited the simulation through a rectangular 2D model. In this article water was simulated using The Navier-Stokes equations and discretized the equations using backward,forward, and central Finite Difference methods. Taken into consideration the boundary conditions using Neumann and Dirichlet conditions, as well as Laplace and Poisson equations were discretized using Finite difference methods. The simulation was done through several steps to achieve final results as defining the geometry and boundary conditions of the micro simplified model, discretizing the equations, and utilizing Matlab programming to put the numerical scheme into practice. The visualization of simulation results was done in Matlab to see the flow behaviour. The article highlights the principles and methods that can be applied to simulate fluid through waste water treatment on a simplified database of 2D screengrid.

1. INTRODUCTION

Wastewater is a very important process n removing pollutants and wastes from the water, whereas it benefits the whol community since it is being discharged back into the enviroment for the purpose of using it in homes,industries, and everywhere. The first treatment process is filtering where water flows through screen grids, and simulating the flow in between the screen grids to know what's going on there can lead to the improvement of the treatment process.

In order to have an idea of what's going on through the screen, it is important to understand the way water flows inside of the grids. Computational Fluid Dynamics which is abbreviated as CFD simulations are the key to analyze and model the flow behaviour of the fluid, where it provides graph plots, flow patterns, and alot of key parameters that let us understand what's going on.

In this article there will be a description of the simulation of the water flow through the defined simplified model using finit difference methods, defining the geometry and boundary conditions of Dirichlet and Neumann, discetizing the equations using finite difference methods ,discussion of the steps involved to achieve the simulation, derivation of the governing equations, implementing and visualizing the numerical scheme using Matlab programming where i will include some of the codes and algorithms that i have used in an attachment file.

2. METHODS USED

A. Taylor Expansion Series

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

where the function f(x) is around the point x=a, and $f^{(n)}(a)$ represents the nth derivative of f(x) evaluated at x=a, and n! represents the factorial of n.

B. Finite Difference methods(Euler's methods)

Where it starts from the Taylor's theorem so that it gives

$$u(x) = u(x_i) + (x-x_i)\frac{\partial u}{\partial x}|_i + \frac{(x-x_i)^2}{2!}\frac{\partial^2 u}{\partial x^2}|_i + \frac{(x-x_i)^n}{n!}\frac{\partial^n u}{\partial x^n} + \cdots$$

replacing x by either (x_i+1) or (x_i-1) we get the expression for the variable values at these points in terms the variable at xi and its derivatives.
Replace (x_i+1) wherever x appears.

$$\frac{\partial u}{\partial x}|_i = \frac{\partial u_{i+1} - u_i}{x_{i+1} - x_i} - \frac{(x_{i+1} - x_i)}{2}\frac{\partial^2 u}{\partial x^2}|_i - \frac{(x_{i+1} - x_i)^2}{3!}\frac{\partial^3 u}{\partial x^3}|_i - h.o.t$$

where h.o.t stands for higher order terms.
If $(x_{i+1} - x_i)$ is small then we neglected terms of order O(Δx)$\rightarrow$Truncation Error, as $\Delta x \rightarrow 0$,error$\rightarrow 0$, and therefore Forward difference converges and results to Forward difference approximation . Similarly can be done for the Backward and central difference, but for the central difference the error$\rightarrow$O$(\Delta x)^2$

B.1. Forward Difference

where the actual and next points are used

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1} - u_i}{\Delta x}$$

B.2. Backward Difference

where the actual and previous points are used

$$\frac{\partial u}{\partial x} \approx \frac{u_i - u_{i-1}}{\Delta x}$$

B.3. Central Difference

where the previous and next points are used

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1} - u_{i-1}}{\Delta x}$$

C. Second order derivatives

Obtaing central difference approximation of $2_n d$ order by combining the Forward and Backward difference for 1_{st} derivative. Considering taylor expansion

$$u_{i+1} = u_i + \Delta x \frac{\partial u}{\partial x}\Big|_i + \frac{\Delta x^2}{2}\frac{\partial^2 u}{\partial x^2}\Big|_i + \frac{\Delta x^3}{6}\frac{\partial^3 u}{\partial x^3}\Big|_i + h.o.t...$$
$$u_{i-1} = u_i - \Delta x \frac{\partial u}{\partial x}\Big|_i + \frac{\Delta x^2}{2}\frac{\partial^2 u}{\partial x^2}\Big|_i + \frac{\Delta x^3}{6}\frac{\partial^3 u}{\partial x^3}\Big|_i + h.o.t...$$

By adding them together, it gives the forward approximate for 2_{nd} derivative :

$$\frac{\partial^2 u}{\partial x^2}\Big|_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} - O(\Delta x^2)$$

where $O(\Delta x^2)$ is the order of Δx^2

D. Explicit/Implicit Finite Difference methods

D.1. Explicit method

Can be computed forward in time using quantities from previous time steps. For example:

$$\frac{\partial T}{\partial t} = k\frac{\partial^2 T}{\partial x^2}$$
$$\rightarrow \frac{T_i^{n+1} - T_i^n}{\Delta t} = k\frac{T_{i+1}^n - 2T_i^n + T_{i-1}}{\Delta x^2}$$
$$\rightarrow T_i^{n+1} = T_i^n + k\Delta t\frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2}$$

D.2. Implicit method

Are evaluated at new timestep, the time update depends on its own, a casual recursive computation is not specified. For example:

$$\frac{\partial T}{\partial t} = k\frac{\partial^2 T}{\partial x^2}$$
$$\rightarrow \frac{T_i^{n+1} - T_i^n}{\Delta t} = k\frac{T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}}{\Delta x^2}$$

3. GOVERNING EQUATIONS AND THEIR DERIVATIONS

The Navier-Stokes equations are based on several laws that are

1. The law of conservation of mass
2. The law of conservation of momentum
3. The law of conservation of energy

Navier-Stokes equations are derived based on small element of the fluid taking into consideration that the size limit approaches to zero. The derived equations come in a set of partial differential equations that will be used to simulate fluid flow.
Starting by the first equation the derivation of the law of conservation of mass where it is known that for a system the $\frac{dM}{dt} = 0$ that mass cannot be created nor destroyed.

1. Conservation of Mass

$$\frac{\partial}{\partial t}\int \rho dV + \int \rho \cdot \tilde{\mathbf{v}} \cdot \hat{n} \cdot dA = 0$$

Where ρ is the density of the fluid, dV is the infinte decimal volume element,$\vec{v}$ isthe velocity, $\hat{n}$ is the outward normal, and dA is the differential of area.
$\frac{\partial}{\partial t}\int \rho dV$ is the rate of mass change in control volume.
$\int \rho \cdot \tilde{\mathbf{V}} \cdot \hat{n} \cdot dA$ is the net rate at which the mass is flowing across the control system.

The differential form of the conservation of mass that will be used in CFD calculations considering the fluid element of $\partial x \partial y \partial z$ is:

1.1 Rate of change of mass:

Assuming ρ is constant since it is considered small relative to the difference of volume.

$$\frac{\partial}{\partial t}\int \rho dV = \frac{\partial \rho}{\partial t}\partial x \partial y \partial z$$

1.2 Rate of mass flow:

Assuming a small cube and starting with x-direction of the quantity ρu where ρu = x-component of mass flow rate, the flow of mass can be obtained using the well known taylor series expansion resulting into these equations of :

$$massout = [\rho u + \frac{\partial}{\partial x}\rho u\frac{\partial x}{2}]\partial y \partial z$$
$$massin = [\rho u - \frac{\partial}{\partial x}\rho u\frac{\partial x}{2}]\partial y \partial z$$

Net rate mass flow in X-direction

$$\frac{\partial}{\partial x}(\rho u)\partial x \partial y \partial z$$

Similarly for y and z directions resulting in :

$$\frac{\partial}{\partial y}(\rho v)\partial x \partial y \partial z$$
$$\frac{\partial}{\partial z}(\rho w)\partial x \partial y \partial z$$

and the sum of these 3 equations will be the net rate of mass outflow which represents the second integral of The conservation of mass's equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) + \frac{\partial}{\partial z}(\rho w) = 0$$

2 Conservation of Momentum:

2.1 Conservation of Momentum for a system

Applying Newton's law to obtain the following equation:

$$\vec{F} = \frac{d}{dt}\int_{sys} \vec{V} dm$$

2.2 Conservation of Momentum for a control volume

$$\sum \vec{F_{cv}} = \frac{\partial}{\partial t}\int_{cv} \vec{V}\rho dv + \int_{cs} \vec{V}\rho\vec{V}\cdot\hat{n} dA$$

Consider an infinitesimal fluid mass ∂m:

$$\partial\vec{F} = \frac{d}{dt}\vec{V}dm = \partial m\frac{d}{dt}\vec{V}$$

$\frac{d\vec{V}}{dt}$ represents the acceleration then :

$$\partial\vec{F} = \partial m\cdot\vec{a}$$

There are two types of forces considered in fluids:

1. Body forces: As in weight of the element which is the differential of body force equals mass multiplied by the acceleration which is gravity

$$\partial\vec{F_b} = \partial m\cdot\vec{g}$$

2. Surface forces: which are also two types normal and tangential to the element of area.

Consider an element of area ∂A
∂F_n: normal to ∂A
$\partial F_1, \partial F_2$: tangent to ∂A
Normal Stress: $\sigma_n = \lim_{\sigma_n\to 0}\frac{\partial F_n}{\partial A}$
Shearing Stresses: $\tau_1 = \lim_{\partial A\to 0}\frac{\partial F_1}{\partial A}$, $\tau_2 = \lim_{\partial A\to 0}\frac{\partial F_2}{\partial A}$

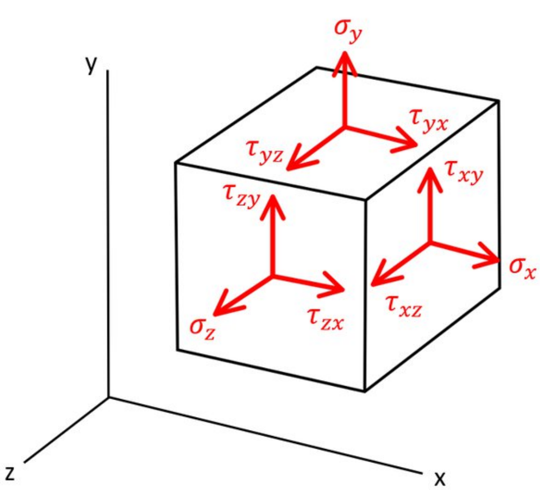

Fig. 1. showing the normal and shear stresses with respect to coordinate system. [1]

Using Taylor series expansion same as used in mass in and out equations to get the normal and shear stresses in x,y, and z directions, and combining them to get sum of forces with respect to the coordinates :

$$\partial F_{sx} = (\frac{\partial}{\partial x}\sigma_{xx} + \frac{\partial}{\partial y}\tau_{yx} + \frac{\partial}{\partial z}\tau_{zx})\partial x\partial y\partial z$$

$$\partial F_{sy} = (\frac{\partial}{\partial y}\sigma_{yy} + \frac{\partial}{\partial x}\tau_{xy} + \frac{\partial}{\partial z}\tau_{zy})\partial x\partial y\partial z$$

$$\partial F_{sz} = (\frac{\partial}{\partial z}\sigma_{zz} + \frac{\partial}{\partial x}\tau_{xz} + \frac{\partial}{\partial y}\tau_{yz})\partial x\partial y\partial z$$

where F_{sz}, F_{sy}, and F_{sx} are the surface forces in x,y, and z directions.

Equations of motion which is a physical argument based on Newton's law:

$$\partial\vec{F} = \partial m\cdot\vec{a}$$
$$\partial m = \rho\partial x\partial y\partial z$$

X-Direction:

$$\rho g_x + \frac{\partial}{\partial x}\sigma_{xx} + \frac{\partial}{\partial y}\tau_{yx} + \frac{\partial}{\partial z}\tau_{zx} = \rho(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z})$$

Y-Direction:

$$\rho g_y + \frac{\partial}{\partial y}\sigma_{yy} + \frac{\partial}{\partial x}\tau_{xy} + \frac{\partial}{\partial z}\tau_{zy} = \rho(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z})$$

Z-Direction:

$$\rho g_z + \frac{\partial}{\partial z}\sigma_{zz} + \frac{\partial}{\partial x}\tau_{xz} + \frac{\partial}{\partial y}\tau_{yz} = \rho(\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z})$$

For an Inviscid Flow which means that there are no shearing stresses since viscosity is neglected then:

$$\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = -P$$

Which gives the Euler's equation for example in X-direction:

$$\rho g_x - \frac{\partial P}{\partial x} = \rho(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z})$$

Differential Form of the Fluid Equations, and reducing the unknowns by limiting everything to Newtonian fluid where it has linear relationship between stresses and the rate of deformation.[2] Notice that neglecting the temperature assuming that there is no heat exchange doesn't need to have the energy equation (Isothermal).

Normal stresses:

$$\sigma_{xx} = -P + 2\mu\frac{\partial u}{\partial x}$$
$$\sigma_{yy} = -P + 2\mu\frac{\partial v}{\partial y}$$
$$\sigma_{zz} = -P + 2\mu\frac{\partial w}{\partial z}$$

where P is the thermodynamic pressure and μ is the coefficient of viscosity of newtonian fluid.

Assuming Incompressible Fluid which means that $\nabla \cdot \vec{V} = 0$ and adding the 3 normal stresses then

$$\sigma_{xx} + \sigma_{yy} + \sigma_{zz} = -P$$

For shearing stresses:

$$\tau_{xy} = \tau_{yx} = \mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})$$

$$\tau_{yz} = \tau_{zy} = \mu(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y})$$

$$\tau_{zx} = \tau_{xz} = \mu(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z})$$

Substituting these into equations equations of momentum respectively.It will give:

$$-\frac{\partial p}{\partial x} + 2\mu\frac{\partial^2 u}{\partial x^2} + \mu(\frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 v}{\partial y \partial x}) + \mu(\frac{\partial^2 w}{\partial z \partial x} + \frac{\partial^2 u}{\partial z^2})$$

Rewriting it as :

$$-\frac{\partial p}{\partial x} + \mu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}) + \mu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial y \partial x} + \frac{\partial^2 w}{\partial z \partial x})$$

The right side can be written as $\mu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial y \partial x} + \frac{\partial^2 w}{\partial z \partial x}) = 0$ because of incompressibility.
The X-direction momentum equation can then be written as

$$\rho(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}) = \rho g_x - \frac{\partial p}{\partial x} + \mu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2})$$

Using vector notation to obtain the Navier Stokes Equations butin Y and Z directions too:

$$\rho\left(\frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \nabla)\vec{V}\right) = \rho\vec{g} - \nabla p + \mu\nabla^2\vec{V}$$

Where $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$

Due to the complexity of computing and solving in 3 Dimensions, the simplified equations into 2D will be used in the further steps. 2D Navier-Stokes equations are explained step by step in the next section in 12 steps to compute Navier-Stokes equations.[2]

4. STEPS TO COMPUTE NAVIER-STOKES EQUATIONS

A. Step 1: 1D linear convection:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{1}$$

where u is a quantity that is transported with a flow that has a constant velocity c, where nonlinearity was neglected.
Assume arbitrary initial profile of $u(x, t = 0) = u_o(x)$ then after time t the quantity u(x) is just the initial profile displaced with distance $x = ct$.
Solution is given simply as $u(x, t) = u_o(x - ct)$

A.1. Discretizing the equation

using Finite difference methods:
Space-time discretization:
where,
$i \rightarrow$ index of grid in x
$n \rightarrow$ index of grid in t
In this there will be a use of both forward and backward Finite difference methods in time and in space respectively within the numerical scheme.
Discrete equation of equation (1) using Forward and backward Finite difference method:

$$\frac{{u_i}^{n+1} - u_i^n}{\Delta t} + c\frac{u_i^n - u_{i-1}^n}{\Delta x} = 0 \tag{2}$$

A.2. Tranpose the equation

to get the unkown value at t_{n+1} assuming that the inital are know at time t_n

$${u_i}^{n+1} = u_i^n - c\frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n)$$

B. Step 2: 1D convection also known as Inviscid Burgers equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{3}$$

The only difference with step 1 is that here u is instead of c where non-linearity was added.
Discrete equation of equation (3) using Forward and backward finite difference method:

B.1. Discretize the equation

$$\frac{{u_i}^{n+1} - u_i^n}{\Delta t} + u_i^n\frac{u_i^n - u_{i-1}^n}{\Delta x} = 0 \tag{4}$$

B.2. Tranpose the equation

$${u_i}^{n+1} = u_i^n - u_i^n\frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n)$$

C. Step 3: 1D Diffusion known as heat equation:

$$\frac{\partial u}{\partial t} = \nu\frac{\partial^2 u}{\partial x^2} \tag{5}$$

Where,
$u \rightarrow$ temperature.
$\nu \rightarrow$ diffusion coefficient.

C.1. Discretizing the equation:

But here instead of using the BD in space we are going to use Central difference method :

$$\frac{{u_i}^{n+1} - u_i^n}{\Delta t} = \nu\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} \tag{6}$$

C.2. Tranpose the equation

$${u_i}^{n+1} = u_i^n + \nu\frac{\Delta t}{\Delta x^2}(u_{i+1}^n - 2u_1^n + u_{i-1}^n)$$

D. Step 4: 1D Burgers equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{7}$$

Where it is a combination of equations (3)(5)

D.1. Discretize the equation:

As before using FD and BD for time and space respectively, but 2_{nd} order method for the second derivatives :

$$\frac{u_i{}^{n+1}-u_i^n}{\Delta t}+u_i^n\frac{u_i^n-u_{i-1}^n}{\Delta x}=\nu\frac{u_{i+1}-2u_i+u_{i-1}}{\Delta x^2} \tag{8}$$

D.2. Tranpose the equation

$$u_i{}^{n+1}=u_i^n-u_i^n\frac{\Delta t}{\Delta x}(u_i^n-u_{i-1}^n)+\nu\frac{\Delta t}{\Delta x^2}(u_{i+1}^n-2u_1^n+u_{i-1}^n)$$

E. Step 5: 2D Burgers equation:

$$\frac{\partial u}{\partial t}+c\frac{\partial u}{\partial x}+c\frac{\partial u}{\partial y}=0 \tag{9}$$

Where it is a 2D version of (1)

E.1. Discretize the equation:

$$\frac{u_{ij}{}^{n+1}-u_{ij}^n}{\Delta t}+c\frac{u_{ij}^n-u_{i-1,j}^n}{\Delta x}+c\frac{u_{ij}^n-u_{i,j-1}^n}{\Delta y}=0 \tag{10}$$

E.2. Tranpose the equation

$$u_{ij}^{n+1}-u_{ij}^n-c\frac{\Delta t}{\Delta x}(u_{ij}^n-u_{i-1,j}^n)-c\frac{\Delta t}{\Delta x}(u_{ij}^n-u_{i,j-1}^n)$$

F. Step 6: 2D Convection:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=0 \tag{11}$$

Where it is a 2D version of (3)

F.1. Discretize the equation:

$$\frac{u_{ij}{}^{n+1}-u_{ij}^n}{\Delta t}+u_{ij}^n\frac{u_{ij}^n-u_{i-1,j}^n}{\Delta x}+v_{ij}^n\frac{u_{ij}^n-u_{i,j-1}^n}{\Delta y}=0 \tag{12}$$

$$\frac{v_{ij}{}^{n+1}-v_{ij}^n}{\Delta t}+u_{ij}^n\frac{v_{ij}^n-v_{i-1,j}^n}{\Delta x}+v_{ij}^n\frac{v_{ij}^n-v_{i,j-1}^n}{\Delta y}=0 \tag{13}$$

G. Step 7: 2D Diffusion:

$$\frac{\partial u}{\partial t}=\nu(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2} \tag{14}$$

G.1. Discretize the equation:

$$\frac{u_{ij}{}^{n+1}-u_{ij}^n}{\Delta t}=\nu\frac{u_{i-1,j}^n-2u_{i,j}^n+u_{i+1,j}^n}{\Delta x^2}+\nu\frac{u_{i,j-1}^n-2u_{i,j}^n+u_{i,j+1}^n}{\Delta y^2} \tag{15}$$

H. Step 8: 2D Burgers equation:

$$\begin{aligned}\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+u\frac{\partial u}{\partial y}&=\nu(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2})\\ \frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+u\frac{\partial v}{\partial y}&=\nu(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2})\end{aligned} \tag{16}$$

H.1. Discretize the equation:

$$\begin{aligned}&(1)\frac{u_{ij}{}^{n+1}-u_{ij}^n}{\Delta t}+u_{i,j}^n\frac{u_{ij}^n-u_{i-1,j}^n}{\Delta x}+v_{i,j}^n\frac{u_{ij}^n-u_{i,j-1}^n}{\Delta y}\\ &=\nu(\frac{u_{i-1,j}^n-2u_{i,j}^n+u_{i+1,j}^n}{\Delta x^2}+\frac{u_{i,j-1}^n-2u_{i,j}^n+u_{i,j+1}^n}{\Delta y^2})\\ &(2)\frac{v_{ij}{}^{n+1}-v_{ij}^n}{\Delta t}+u_{i,j}^n\frac{v_{ij}^n-v_{i-1,j}^n}{\Delta x}+v_{i,j}^n\frac{v_{ij}^n-v_{i,j-1}^n}{\Delta y}\\ &=\nu(\frac{v_{i-1,j}^n-2v_{i,j}^n+v_{i+1,j}^n}{\Delta x^2}+\frac{v_{i,j-1}^n-2v_{i,j}^n+v_{i,j+1}^n}{\Delta y^2})\end{aligned} \tag{17}$$

I. Step 9 : Laplace equation

$$\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}=0 \tag{18}$$

I.1. Discretize the equation

Using 2_{nd} order Central difference method

$$\frac{p_{i+1,j}^n-2p_{i,j}^n+p_{i-1,j}^n}{\Delta x^2}+\frac{p_{i,j+1}^n-2p_{i,j}^n+p_{i,j-1}^n}{\Delta y^2}=0 \tag{19}$$

I.2. Transpose

$$p_{i,j}^n=\frac{\Delta y^2(p_{i+1,j}^n+p_{i-1,j}^n)+\Delta x^2(p_{i,j+1}^n+p_{i,j-1}^n)}{2(\Delta x^2+\Delta y^2)}$$

I.3. Analytical solution

This is to compare the results between analytical and numerical solutions

$$p(x,y)=\frac{x}{4}-4\sum_{n=1}^{\infty}\frac{1}{(n\pi)^2\sin h(2\pi n)}\sin h(n\pi x)\cos(n\pi y)[2] \tag{20}$$

I.4. Error between Analytical and Numerical solution

Error=p(analytical)-p(numerical)

J. Step 10: The Poisson equation

Same as equation (18) but with a right hand side variable b.

$$\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}=b \tag{21}$$

J.1. Discretize the equation

$$\frac{p_{i+1,j}^n-2p_{i,j}^n+p_{i-1,j}^n}{\Delta x^2}+\frac{p_{i,j+1}^n-2p_{i,j}^n+p_{i,j-1}^n}{\Delta y^2}=b_{i,j}^n \tag{22}$$

J.2. Transpose

$$p_{i,j}^n=\frac{\Delta y^2(p_{i+1,j}^n+p_{i-1,j}^n)+\Delta x^2(p_{i,j+1}^n+p_{i,j-1}^n)-b_{i,j}^n\Delta x^2\Delta y^2}{2(\Delta x^2+\Delta y^2)}$$

K. Final Step solving Navier-Stokes equations for a cavity flow

Discretize the momentum equation in 2D

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\nu(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}) \tag{23}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial y}+\nu(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}) \tag{24}$$

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = -\rho\left(\frac{\partial u}{\partial x}\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial x} + \frac{\partial v}{\partial y}\frac{\partial v}{\partial y}\right) + \rho\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \quad \textbf{(25)}$$

Discretize all three equations (23)(24)(25).

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} + u_{i,j}^{n}\frac{u_{i,j}^{n} - u_{i-1,j}^{n}}{\Delta x} + v_{i,j}^{n}\frac{u_{i,j}^{n} - u_{i,j-1}^{n}}{\Delta y} =$$
$$= -\frac{1}{\rho}\frac{p_{i+1,j}^{n} - p_{i-1,j}^{n}}{2\Delta x} + \nu\left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{\Delta x^2} + \right.$$
$$\left. + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\Delta y^2}\right) \quad \textbf{(26)}$$

first equation is discretized.

$$\frac{v_{i,j}^{n+1} - v_{i,j}^{n}}{\Delta t} + u_{i,j}^{n}\frac{v_{i,j}^{n} - v_{i-1,j}^{n}}{\Delta x} + v_{i,j}^{n}\frac{v_{i,j}^{n} - v_{i,j-1}^{n}}{\Delta y} =$$
$$= -\frac{1}{\rho}\frac{p_{i,j+1}^{n} - p_{i,j-1}^{n}}{2\Delta y} + \nu\left(\frac{v_{i+1,j}^{n} - 2v_{i,j}^{n} + v_{i-1,j}^{n}}{\Delta x^2} + \right.$$
$$\left. + \frac{v_{i,j+1}^{n} - 2v_{i,j}^{n} + v_{i,j-1}^{n}}{\Delta y^2}\right) \quad \textbf{(27)}$$

second is discretized.

$$\frac{p_{i+1,j}^{n} - 2p_{i,j}^{n} + p_{i-1,j}^{n}}{\Delta x^2} + \frac{p_{i,j+1}^{n} - 2p_{i,j}^{n} + p_{i,j-1}^{n}}{\Delta y^2} =$$
$$= -\rho\left(\left(\frac{u_{i+1,j}^{n} - u_{i-1,j}^{n}}{2\Delta x}\right)^2 + 2\left(\frac{u_{i,j+1}^{n} - u_{i,j-1}^{n}}{2\Delta y}\right)\left(\frac{v_{i+1,j}^{n} - v_{i-1,j}^{n}}{2\Delta x}\right) + \right.$$
$$\left. + \left(\frac{v_{i,j+1}^{n} - v_{i,j-1}^{n}}{2\Delta y}\right)^2\right) + \frac{\rho}{\Delta t}\left(\frac{u_{i+1,j}^{n} - u_{i-1,j}^{n}}{2\Delta x} + \frac{v_{i,j+1}^{n} - v_{i,j-1}^{n}}{2\Delta y}\right) \quad \textbf{(28)}$$

Equations (23)(24)(25) were discretized into equations (26)(27)(28)using 4 discretization methods that are foraward, backward, central, and 2nd order central difference methods.

5. NUMERICAL SIMULATIONS DISCUSSIONS AND RESULTS

Computing the Navier-Stoke equations is a really complex task, i simplified the model into a square channel of the screen rid, and simplified the Navier-Stokes quations into 2 Dimensional were i started from 1D and went up to 2D to reduce complexity. As included in section 4. about the steps to compute the Navier-Stokes equations, 12 steps were taken into consideration and each step has its own code and its own numerical simulation. It made the final step easier by going through step by step. Numerical methods were used to discretize most of the equations and to solve them. The simulations were implemented on Matlab were i tried my best to understand the whole concepts of Navier-Stokes equations and the simplified steps [2] and try to code it in my own way. The script will be found in the same folder of this article where it is attached.

First step was to solve the 1D linear convection equation using Forward and Backward Difference schemes for the time and space derivatives. Initial conditions were established were there is a quantity of variable u that is being transported by a constant velocity which is "c" within the domain and the length which was set to be from 0 to 10 [0 10], u=2 from interval 1 to 2 and everywhere else 1, and also setting the boundary conditions which is u=1 at locations of x at 0 and 10. the result of solving

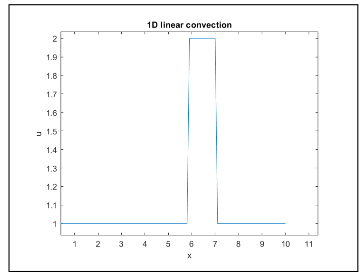

Fig. 2. showing the numerical result of the 1D linear convection

this equation is in figure 2. As you can see it is linear and kind of a square wave.

Step 2 was kind of the same concept of step 1 but the constant velocity c changed into u variable to be computed at a location of i with time. So basically it is 1D non-linear convection and has the same initial, boundary,discretized conditions as step 1. as you can see the the non-linearity in figure 3. As you can see the velocity non-linearly increase gradually and started decreasing after the initial conditions stated that u is 2 from interval 1 to 2.

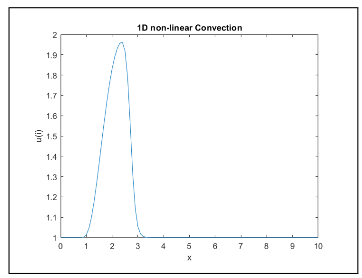

Fig. 3. showing the numerical result of the 1D no-linear convection

In step 3 which is the 1D diffusion the equation was solved using the Central Difference scheme for the second derivative which is a combination of the forward and backward difference. Initial conditions still the same except that number of time steps is reduced to make sure of stability, and two more values were introduced the viscosity and the courant number. The Courant number is a dimensionless value representing the time a particle stays in one cell of the mesh[3]. Overall, this code uses the Central Difference method to create a numerical solution to the 1D Diffusion equation and shows the result at each time step as shown in figure 4.

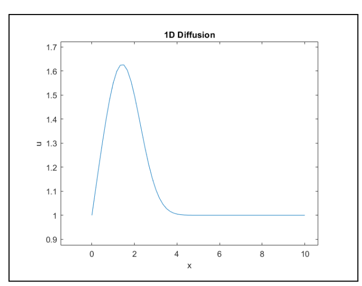

Fig. 4. showing the numerical result of the 1D no-linear convection

Step 4 solves the Burgers equation in 1D which is a nonlinear partial differential equation that explains fluid flow. The equation is quite difficult to be solved analatucally, therefore numerical methods are used oftenly to approximate the solutions. Initial and boundary conditions were defined using mathemtical symbols, the equation was descritezed in 3 methods FD for time, BD for space, and CD for second order. The numerical and analytical solutions were plotted in the same graph were we can compare the result between the computed and the exact. As you can see in figure 5 the exact solution doesn't differ that much than the numerical, therefore we can say that the error is minimal here.

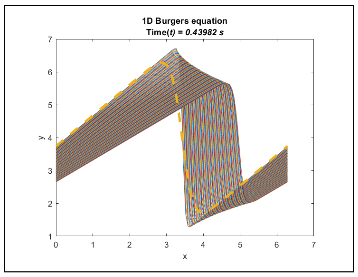

Fig. 5. showing the numerical result of the 1D Burgers equation

Step 5 implements a solution of a 2D linear convection where the difference between this step and step 1 is that we now have two spatial dimensions to account for as we step forward in time. Initial and boundary conditions are set, and the 2D mesh is created, and the time loop for each time step the velocity is being updated and approximated using finite differences. This is the last form of the rectangular 3d mesh plot shown in 6. Run the Matlab code to see the decomposition.

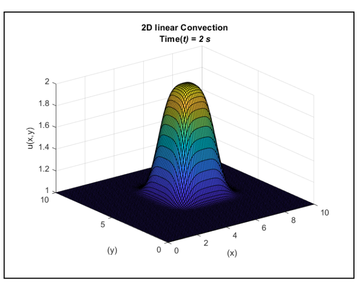

Fig. 6. showing the numerical result of the 2D linear convection

Step 6 basically the same concepts between step 1 and 2, there is an extra variable here that is in the y direction, we solve 2D Convection, represented by the pair of coupled partial differential equations to result in this 3D non-linear mesh plot in figure 7

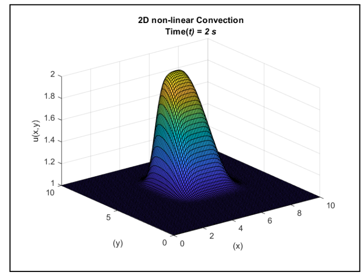

Fig. 7. showing the numerical result of the 1D no-linear convection

Step 7 solving the 2D diffusion equation, it shows the high concentration of u in the starting range diffuses outwards, causing a progressive reduction in the concentration in that area. This process is clearly seen in the graph 8.

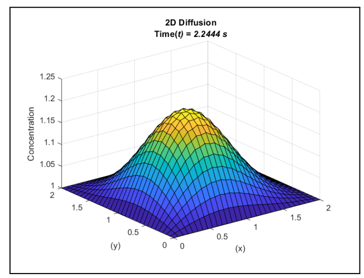

Fig. 8. showing the numerical result of the 2D Diffusion

Step 8 2D Burgers' equation is a combines the effects of the nonlinear advection associated with conservation laws and, the diffusion associated with the heat equation. These two result in non-linearity[4].The solution is then updated using time iterations and finite difference approximations of the convection and diffusion terms in a nested for loop.which displays a 3D surface plot of the solution as shown in figure 9.

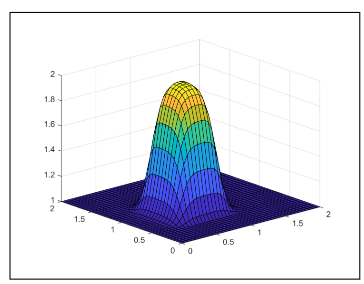

Fig. 9. showing the numerical result of the 2D Burgers equation

Step 9 Using central differences as the discretization, this algorithm numerically solves the Laplace equation as shown in figure 10. The steady-state distribution of heat in an area is described by a partial differential equation the Laplace equation. The equation is solved within a 2x2 square domain. Analytical solution was also included were the erro was computed as shown in figure**??**

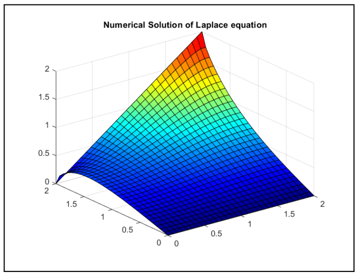

Fig. 10. showing the numerical result of the Laplace equation

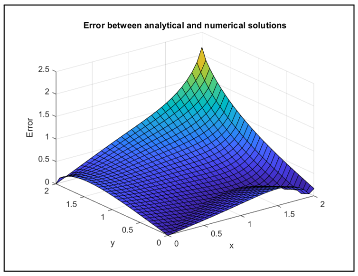

Fig. 11. showing the truncation error between the exact and numerical

Step 10 poisson's equation where it is the same as laplace's equation but it has an extra source. Unlike to the Laplace equation, the field contains a finite value that has an impact on the solution. The source term spikes with a positive value at one location and a negative at another location in the domain. We can see that the source spikes lead to the propagation of a wave-like pattern over the domain. After a number of time steps, the wave pattern gradually stabilizes and the colormap displays the solution's magnitude, with blue denoting low values and red denoting high values as shown in figure 12

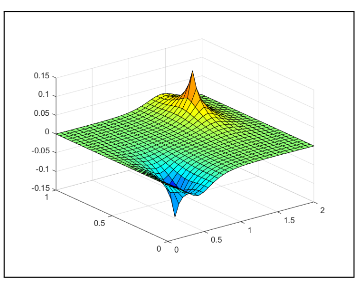

Fig. 12. showing the numerical result of the Poisson's equation

Step 11 which is almost the last step The Navier-Stokes equations were fully used on a 2D lid cavity example were the motion of the fluid inside is described. Initial and boundary conditions,the size of the grid, the time step, and the physical properties of the fluid where all set. Preallocation of the arrays to store the velocity, pressure, and other variables. The equations were discretized by the first-order forward difference method to approximate the time derivative of the velocity components, and the second-order central difference method was used to approximate the spatial derivatives of the velocity components and pressure. The Poisson equation, which connects the pressure's Laplacian to the velocity field's divergence, is then used by the algorithm to solve for the pressure field. The pressure field and the old velocity field are then used to update the new velocity field. Yielding to a plot of simulation of the flow of fluid in a 2D lid cavity which represents the pressure and velocity field that show the direction and magnitude of the fluid flow as shown in figure **??**.

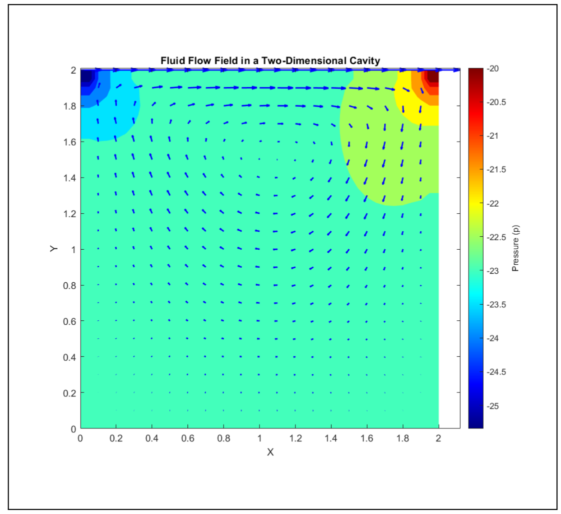

Fig. 13. showing the numerical result of Navier-Stokes equations in a 2D lid cavity

Step 12 which is the final step of the whole simulation, it is the same as step 11 but the difference is to add a source term F to the u-momentum equation, to mimic the effect of a pressure-driven channel flow. to the which is the simulation of a simplified square channel of a screen grid that is in a wastewater treatment process. As well as the whole steps Initial conditions and dirichlet boundary conditions were applied, where the velocity in x and y directions and the pressure are 0 everywhere and at the boundaries u v and p are periodic in x direction which means whatever happen at x=0 happen at x=2, and no slip in the y direction (u,v,dp/dy=0 @ y=0,2).by that the momentum equation and the pressure poisson equations are discretized using central difference methods. This simulation gives the velocity and pressure fields of the water flowing through the 2D square channel shown in figure 14 which shows how the velocity decreases towards the boundaries, and here we can assure that our boundary conditions dirichlet conditions are working perfectly, and at the same considering such flow is the typical flow through a pipe.

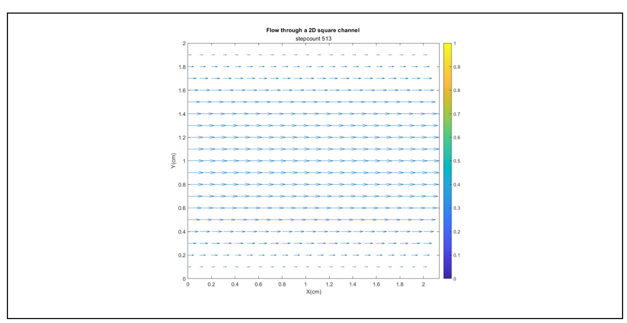

Fig. 14. showing the numerical result of Navier-Stokes equations through a square channel

6. CONCLUSION

Flow field through a square channel of a screen grid was numerically investigated in this study, were all the governing equations used were discretized with severl numerical methods and solved by them. As known some of the analytical equations are too complex to solve without computation, and numerical methods of simulation made it easier. In this article my main goal was to reach to a way to be able to simulate the fluid inside that screen, although i know that i simplified it and neglected several things due to complexity, but at the end i still got the results through a square channel. Knowing that the domain was 2x2 of the square channel and taking the suitable boundary and initial conditions that have completed each other in order to be able to solved and iterated throughout the loops. At each specific time a new value was being updated, everything was derived and explained till it reached till the solutions of the navierstokes from the beginning, please check the matlab script available in the folder. In order to optimize the treatment process and guarantee the effective removal of pollutants and wastes from water, fluid through a waste water treatment process simulation utilizing CFD simulations and finite difference methods is an important method.In conclusion, it is important to carefully evaluate the geometry, boundary conditions, governing equations, numerical methods, and implementation while modeling fluid flow through a 2D rectangular channel. These methods allowed us to correctly model the flow dynamics and assess the channel using Matlab and the finite difference methods. In order to examine how fluid flow behaves in practical applications, this simulation can be expanded to include more intricate geometries and boundary conditions.

REFERENCES

1. J. Stam, "Real-time fluid dynamics for games," in *Proceedings of the game developer conference,* vol. 18 (2003), p. 25.
2. L. Barba and G. Forsyth, "Cfd python: the 12 steps to navier-stokes equations," J. Open Source Educ. **2**, 21 (2019).
3. M. I. Yuce and D. A. Kareem, "A numerical analysis of fluid flow around circular and square cylinders," Journal-American Water Work. Assoc. **108**, E546–E554 (2016).
4. C. Shu and B. E. Richards, "Application of generalized differential quadrature to solve two-dimensional incompressible navier-stokes equations," Int. J. for Numer. Methods Fluids **15**, 791–798 (1992).
5. D. Holmes and S. Connell, "Solution of the 2d navier-stokes equations on unstructured adaptive grids," in *9th computational fluid dynamics conference,* (1989), p. 1932.
6. B. Dewals, S. Kantoush, S. Erpicum, M. Pirotton, and A. Schleiss, "Experimental and numerical analysis of flow instabilities in rectangular shallow basins," Environ. fluid mechanics **8**, 31–54 (2008).
7. D.-h. Yu and A. Kareem, "Numerical simulation of flow around rectangular prism," J. wind engineering industrial aerodynamics **67**, 195–208 (1997).
8. E. Hawassa, "Numerical simulation of incompressible navier stokes equations applied to model blood flow," Ph.D. thesis, Hawassa University (2017).
9. O. Herrera-Granados, "Numerical analysis of flow behavior in a rectangular channel with submerged weirs," Water **13**, 1396 (2021).
10. C. Zidani, B. Benyoucef, F. Didi, and N. Guendouz, "Simulation and numerical analysis of a rectangular pipe with transversal baffle–comparison between zigzag and plane baffles," Arch. Thermodyn. pp. 269–283 (2020).
11. E. G. Tsega, "A finite volume solution of unsteady incompressible navier–stokes equations using matlab," Num. Com. Meth. Sci. Eng **1**, 117 (2019).

1 2 3 4 5 6 7 8 9 10 11