FOSSEE CFD-OpenFOAM Semester Long Internship Spring 2026

on

CFD Simulation of Hydraulic Jump using OpenFOAM v2412

Karri Vijay Reddy[1], Ranjit Desai[2], Himani Garg[3],
Sameer Jadhav[4], Parees Palkar[5]

[1]Institute of Aeronautical Engineering (IARE), Hyderabad
[2]WaterUnit Infrastructure Consultancy (Guide)
[3]Lund University, Sweden (Co-guide)
[4]IIT Bombay (Co-guide)
[5]FOSSEE, IIT Bombay (Mentor)

Abstract

A two-dimensional CFD simulation of a hydraulic jump in a rectangular open channel was carried out using OpenFOAM v2412. The VOF solver `interFoam` was used with the `kOmegaSST` RANS turbulence model. The inflow Froude number was set to $Fr_1 = 6.0$, giving a theoretically predicted conjugate depth of $y_2 = 0.80$ m. A sharp-crested weir placed at $x = 15.0$ m downstream served as the backwater control structure. The weir height of $h = 0.386$ m was determined by an iterative Excel calculator using the Rehbock weir discharge formula, then cross-checked with a standalone weir-only CFD run. The hydraulic jump simulation reproduced the expected free-surface transition, a recirculation roller zone, and submerged jump conditions. Analytical energy dissipation efficiency came out at 56.4%.

Keywords: hydraulic jump, OpenFOAM, interFoam, Volume-of-Fluid (VOF), kOmegaSST, free surface flow, open-channel hydraulics, sharp-crested weir, Rehbock formula, Froude number, energy dissipation, computational fluid dynamics, FOSSEE.

Contents

List of Tables

List of Figures

List of Equations

1 Introduction

1.1 Background of the Topic

A hydraulic jump occurs when a high-velocity supercritical flow ($Fr > 1$) abruptly transitions to a slower subcritical flow ($Fr < 1$). A large fraction of the kinetic energy is lost in the process through turbulent mixing, roller formation, and surface undulations. Bélanger (1838) first treated this analytically using the integral momentum equation.

In practice, hydraulic jumps appear below spillways, at the base of steep channels, and downstream of sluice gates. Stilling basins are deliberately designed to force a jump and protect downstream structures from scour. The turbulent roller region, strong recirculation, air entrainment, and pressure fluctuations make the jump one of the harder open-channel phenomena to simulate well.

1.2 Motivation of the Study

Experimental and analytical tools for studying hydraulic jumps are well established, but CFD offers things they cannot easily provide:

- Detailed spatial distributions of velocity, pressure, turbulent kinetic energy, and void fraction that are difficult or expensive to measure in a laboratory.
- The ability to study extreme Froude number conditions, large-scale channels, or complex geometries without building a physical model.
- Low-cost parametric studies varying inlet Froude number, tailwater depth, and weir geometry.

This project is part of the FOSSEE Semester Long Internship (IIT Bombay), which promotes open-source scientific computing tools (OpenFOAM, Python, Scilab) in Indian engineering education.

1.3 Problem Statement

The goal is to simulate a steady hydraulic jump in a two-dimensional rectangular open channel under the following conditions:

- Inlet Froude number: $Fr_1 = 6.0$ — steady jump regime
- Inlet flow depth: $y_1 = 0.10$ m
- Channel width: $b = 0.10$ m (unit-width equivalent for 2-D)
- Downstream backwater control: sharp-crested weir of height $h = 0.386$ m at $x = 15.0$ m, sized with the iterative Rehbock weir calculator

Target outcomes: conjugate depth $y_2 \approx 0.80$ m, jump length $L_j \approx 4.88$ m (Peterka), and energy dissipation efficiency $\eta \approx 56.4\%$.

1.4 Literature Review

Bélanger (1838) derived the sequent-depth equation from the integral momentum balance, neglecting friction. It remains the standard analytical reference for hydraulic jump depths.

Peterka (1958) gave empirical correlations for jump length ($L_j = 6.1\,y_2$) and stilling basin design. Here, Peterka's formula was used only to set the channel length.

Rehbock (1929) developed the coefficient formula $C = 3.27 + 0.40\,(H/h)$ for sharp-crested weir discharge, which underpins the weir height calculator used in this project.

Chaudhry (2008) covers open-channel hydraulics comprehensively, including weir discharge theory, hydraulic jump classification, and energy dissipation. It was the primary reference for the analytical calculations here.

Bayon et al. (2016) tested several RANS turbulence models (`kEpsilon`, `kOmegaSST`, `v2f`) for hydraulic jump simulation and found `kOmegaSST` gave satisfactory free-surface predictions.

Witt et al. (2018) demonstrated VOF-based CFD simulation of hydraulic jumps using OpenFOAM, validating `interFoam` against experimental data.

2 Governing Equations

The flow is governed by the incompressible, immiscible two-phase Navier–Stokes equations solved under the VOF framework.

Continuity Equation

$$\nabla \cdot \mathbf{U} = 0 \tag{1}$$

Momentum Equation

$$\frac{\partial(\rho \mathbf{U})}{\partial t} + \nabla \cdot (\rho \mathbf{U}\mathbf{U}) = -\nabla p^* + \nabla \cdot \left[\mu_{\text{eff}} \left(\nabla \mathbf{U} + (\nabla \mathbf{U})^{\mathsf{T}}\right)\right] + \rho \mathbf{g} + \mathbf{F}_\sigma \tag{2}$$

where $p^* = p - \rho \mathbf{g} \cdot \mathbf{x}$ is the modified pressure, $\mu_{\text{eff}} = \mu_{\text{lam}} + \mu_t$ is the effective dynamic viscosity, and $\mathbf{F}_\sigma$ is the surface-tension force (negligible at hydraulic scale).

Phase-Fraction Transport (VOF)

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{U}) + \nabla \cdot [\alpha(1-\alpha)\mathbf{U}_r] = 0 \tag{3}$$

The phase fraction $\alpha \in [0,1]$ gives the volume fraction of water ($\alpha = 1$: pure water; $\alpha = 0$: pure air). The third term uses a relative velocity $\mathbf{U}_r$ to compress the interface and limit artificial smearing.

Mixture Properties

$$\rho = \alpha \rho_w + (1-\alpha)\rho_a \tag{4}$$

$$\mu_{\text{lam}} = \alpha \mu_w + (1-\alpha)\mu_a \tag{5}$$

Bélanger Sequent-Depth Equation

$$\frac{y_2}{y_1} = \frac{1}{2}\left(\sqrt{1 + 8Fr_1^2} - 1\right) \tag{6}$$

Froude Number

$$Fr_1 = \frac{U_1}{\sqrt{g\, y_1}} \tag{7}$$

Energy Dissipation

$$\Delta E = \frac{(y_2 - y_1)^3}{4 y_1 y_2} \tag{8}$$

$$\eta = \frac{\Delta E}{E_1}, \qquad E_1 = y_1 + \frac{U_1^2}{2g} \tag{9}$$

Jump Length — Peterka (1958)

$$L_j = 6.10\, y_2 \tag{10}$$

Peterka's formula was used only to size the channel ($L_j = 6.10 \times 0.80 = 4.88$ m), providing a conservative upper bound so the full roller fits within the domain upstream of the weir.

Sharp-Crested Weir Discharge — Rehbock / Chaudhry

The general weir discharge formula (Chaudhry 2008, Eq. 14-9) is:

$$Q = C \cdot L \cdot H^{1.5} \tag{11}$$

where C is the discharge coefficient, L is the effective weir crest length, and H is the head above the weir crest (excluding velocity head). Rehbock (1929) gives:

$$C = 3.27 + 0.40\,\frac{H}{h} \tag{12}$$

where h is the weir height above the channel floor. This expression is valid for $H/h \leq 10$ (Rouse, as cited in Chaudhry 2008).

3 Turbulence Modelling

The `kOmegaSST` (Shear-Stress Transport) model by Menter (1994) was used. It blends the k–ω model near walls, which handles adverse pressure gradients well, with the k–ε model in the free stream, which avoids the free-stream sensitivity of k–ω.

Transport Equations

Turbulent kinetic energy k:

$$\frac{\partial(\rho k)}{\partial t} + \nabla \cdot (\rho \mathbf{U} k) = \tilde{P}_k - \beta^* \rho \omega k + \nabla \cdot [(\mu + \sigma_k \mu_t) \nabla k] \tag{13}$$

Specific dissipation rate ω:

$$\frac{\partial(\rho \omega)}{\partial t} + \nabla \cdot (\rho \mathbf{U} \omega) = \frac{\gamma}{\nu_t} \tilde{P}_k - \beta \rho \omega^2 + \nabla \cdot [(\mu + \sigma_\omega \mu_t) \nabla \omega] + 2(1 - F_1) \frac{\rho \sigma_{\omega 2}}{\omega} \nabla k \cdot \nabla \omega \tag{14}$$

Turbulent eddy viscosity:

$$\mu_t = \frac{\rho a_1 k}{\max(a_1 \omega,\, \|\mathbf{S}\| F_2)} \tag{15}$$

where F_1 and F_2 are blending functions depending on wall distance, and $\|\mathbf{S}\|$ is the magnitude of the strain-rate tensor.

Wall Treatment

`nutkWallFunction` and `omegaWallFunction` are applied on all no-slip walls. The mesh targets y^+ in the log-law region ($30 < y^+ < 300$), consistent with high-Reynolds wall functions. Note that `nutkSSTWallFunction` is not available in OpenFOAM v2412; `nutkWallFunction` is the correct choice.

Inlet Turbulence Values

Table 1: Inlet turbulence boundary values ($I = 5\%$)

Parameter	**Formula**	**Value**	**Unit**
Turbulence intensity I	given	0.05	–
Inlet velocity U_1	$Q/(b\,y_1)$	5.94	m/s
Turb. kinetic energy k	$\frac{3}{2}(U_1 I)^2$	4.14×10^{-3}	$\mathrm{m^2/s^2}$
Spec. dissipation ω	$\varepsilon/(C_\mu k)$	0.1178	$\mathrm{s^{-1}}$
Eddy viscosity ν_t	k/ω	3.51×10^{-5}	$\mathrm{m^2/s}$
ν_t/ν ratio	–	≈ 35	–

4 Computational Domain

4.1 Geometry

The channel is two-dimensional (one cell in the spanwise z-direction). It has a 15.0 m upstream reach, a sharp-crested weir at $x = 15.0$ m, and a 4.0 m downstream pool, for a total length of 19.1 m. All geometry constants are defined in `system/include/hydraulicParameters`.

Table 2: Domain geometry parameters

Parameter	**Value**	**Unit**
Total channel length (`channelLength`)	19.1	m
Channel height	1.5	m
Channel depth (spanwise)	0.1	m
Upstream length (`weirPosition`)	15.0	m
Downstream length (`afterWeirLength`)	4.0	m
Weir start (`weirStart`)	$x = 15.0$	m
Weir end (`weirEnd`)	$x = 15.1$	m
Weir height h	0.386	m
Weir thickness	0.1	m

The upstream reach of 15.0 m is over three times the Peterka jump length ($L_j = 4.88$ m), so the jump toe is expected near $x \approx 10.1$–11.5 m from the inlet.

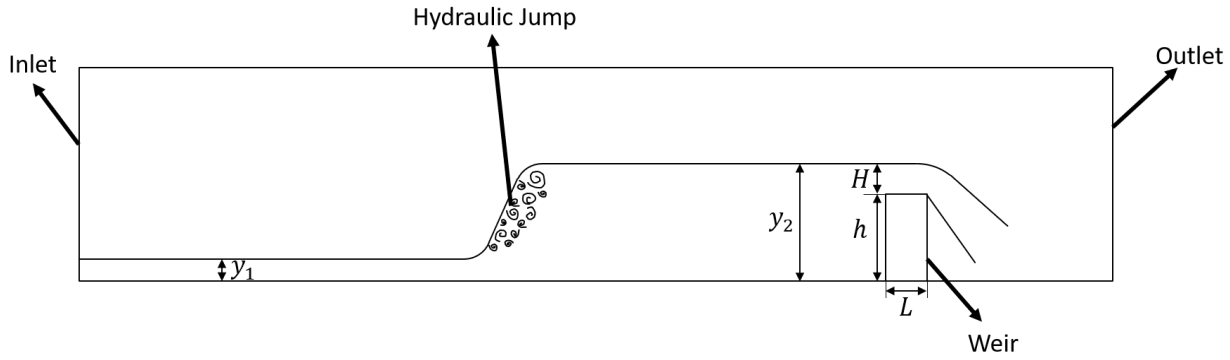

Figure 1: Schematic of the computational domain showing channel, weir, and block layout

4.2 Mesh

The mesh was generated with `blockMesh` from OpenFOAM v2412. The domain uses six hexahedral blocks, all with `simpleGrading (1 1 1)` and no stretching. The inlet patch spans only $0 \leq y \leq 0.1$ m, matching the supercritical inlet depth $y_1 = 0.1$ m.

Table 3: Block structure for `blockMesh`

B.	Description	x-range	y-range (m)	Cells (x×y×z)
0	Upstream bottom	0 – weirStart	0.000 – 0.100	$800 \times 5 \times 1$
1	Upstream middle	0 – weirStart	0.100 – 0.386	$800 \times 15 \times 1$
2	Upstream top	0 – weirStart	0.386 – 1.500	$800 \times 60 \times 1$
3	Weir top	weirStart – weirEnd	0.386 – 1.500	$5 \times 60 \times 1$
4	Downstream top	weirEnd – end	0.386 – 1.500	$180 \times 60 \times 1$
5	Downstream bottom	weirEnd – end	0.000 – 0.386	$180 \times 20 \times 1$

Total cell count: **78,700 hexahedral cells**. The mesh is fully orthogonal with zero non-orthogonality and negligible skewness, confirmed by `checkMesh`.

Figure 2: blockMesh grid showing the six-block structure

4.3 Boundary and Initial Conditions

Velocity and Pressure

Table 4: Boundary conditions for **U** and **p_rgh**

Patch	U type	U value	p_rgh type	value
inlet	`fixedValue`	(5.94, 0, 0) m/s	`fixedFluxPressure`	
outlet	`inletOutlet`	(0, 0, 0)	`totalPressure`	p_0=0
weir	`noSlip`	–	`fixedFluxPressure`	
walls	`noSlip`	–	`fixedFluxPressure`	
atmosphere	`pressureInlet-OutletVelocity`	–	`totalPressure`	p_0=0
front&Back	`empty`	–	`empty`	–

Phase Fraction α.water

Table 5: Boundary conditions for α.water

Patch	Type	Value / Remark
inlet	`fixedValue`	1 (pure water inflow)
outlet	`inletOutlet`	inletValue = 1 (backflow brings water)
atmosphere	`inletOutlet`	inletValue = 0 (backflow brings air)
weir / walls	`zeroGradient`	–
frontAndBack	`empty`	–

Turbulence Variables k, ω, ν_t

Table 6: Boundary conditions for k, ω, and ν_t

Patch	k	ω	ν_t
inlet	`fixedValue` 4.14×10^{-3}	`fixedValue` 0.1178	`calculated`
outlet	`inletOutlet`	`inletOutlet`	`calculated`
weir / walls	`kLowReWall-Function`	`omegaWall-Function`	`nutkWall-Function`
atmosphere	`inletOutlet`	`inletOutlet`	`calculated`
frontAndBack	`empty`	`empty`	`empty`

Initial Conditions (`setFieldsDict`)

The domain was pre-seeded with three water regions to speed up convergence:

1. **Inlet primer** — small box at the inlet ($0 \leq x \leq 0.1$ m, $0 \leq y \leq 0.1$ m), $\alpha = 1$.
2. **Upstream pool** — from $x = 10.12$ m to $x = 15.0$ m, $0 \leq y \leq 0.80$ m, $\alpha = 1$.
3. **Downstream pool** — from $x = 15.1$ m to $x = 19.1$ m at a shallow depth, $\alpha = 1$.

4.4 Solver Selection

`interFoam` was chosen because it solves the two-phase VOF equations natively, captures the air–water free surface, supports RAS turbulence models including `kOmegaSST`, and is the standard OpenFOAM solver for open-channel free-surface flows.

Table 7: Linear solver settings

Variable	Solver	Tolerance
p_{rgh}	GAMG + GaussSeidel smoother	5×10^{-9}
U	PBiCGStab + DILU	10^{-8}
k, ω	smoothSolver + symGaussSeidel	10^{-8}

PIMPLE settings: `nOuterCorrectors = 3`, `nCorrectors = 2`, `nNonOrthogonalCorrectors = 0`.

4.5 Discretisation Schemes

Table 8: Finite-volume discretisation schemes (`fvSchemes`)

Term	**Scheme**
Time derivative	Euler (1^{st}-order implicit)
Gradient – default	Gauss linear
Gradient – **U**	Cell-limited Gauss linear (limiter = 1)
Gradient – α	Gauss linear
Divergence – momentum	Gauss linearUpwind `grad(U)`
Divergence – α	Gauss vanLeer
Divergence – interface compression	Gauss interfaceCompression
Divergence – k, ω	Gauss limitedLinear 1
Laplacian	Gauss linear orthogonal
Surface-normal gradient	orthogonal
Wall distance	meshWave

4.6 Simulation Parameters

Table 9: Key simulation parameters

Parameter	**Value**	**Remarks**
Reynolds number $Re = U_1 y_1/\nu$	$\approx 5.94 \times 10^5$	Fully turbulent
Max Courant number (Co)	0.5	Adaptive Δt
Max α Courant number	0.5	Interface stability
Max time step $\Delta t_{\max}$	0.01 s	Upper bound
End time	20 s	Steady state
Write interval	0.1 s	200 snapshots

5 Implementation in OpenFOAM

5.1 Weir Height Design — Excel Calculator

Before running the full hydraulic jump simulation, the weir height h was determined iteratively using an Excel calculator, the Rehbock weir discharge formula (Eqs. (11)–(12)), and two validation CFD runs.

Design Procedure

The weir must pass the channel inflow discharge $Q = 0.0594$ m^3/s while raising the tailwater to at least y_2. Since C depends on H/h and $H = y_2 - h$, the system is implicit and must be solved iteratively.

Iteration 1 — Initial estimate:

1. Set the initial ratio $r = H/h = 0.6166$ as a starting guess.
2. Use Excel Goal Seek to drive the error $E = Q - C \cdot L \cdot H^{1.5}$ to zero by varying h, giving $h = 0.495$ m.
3. Run weir-only CFD with $h = 0.495$ m. Simulated tailwater: $y_2 = 0.90$ m.

Iteration 2 — Correction using CFD result:

1. Recalculate the head: $H = 0.90 - 0.495 = 0.405$ m.
2. Back-calculate C: $C = Q/(L \cdot H^{1.5}) = 0.0594/(0.1 \times 0.405^{1.5}) = 3.517$
3. Re-run Goal Seek with updated C, giving $h = 0.386$ m.
4. Run second weir-only CFD with $h = 0.386$ m. Simulated tailwater: $y_2 = 0.81$ m — submerged jump confirmed.

Table 10: Excel weir calculator — final values at convergence

Parameter	**Value**	**Unit**
Inlet depth y_1	0.10	m
Inlet Froude number Fr_1	6.0	–
Channel width $L = b$	0.10	m
Flow discharge Q	0.05943	m^3/s
Sequent depth y_2	0.80	m
Ratio $r = H/h$	0.6166	–
Weir height h	0.386	m
Head H	0.305	m
Discharge coefficient C	3.517	–
Weir discharge Q_w	0.05928	m^3/s
Error $Q - Q_w$	-1.52×10^{-4}	m^3/s

	A	B
1	Inlet size	0.1
2	Inlet Froude number	6
3		
4	g	9.81
5		
6	Upstream depth y1	0.1
7	Channel width b	0.1
8		
9	Velocity V1	5.942726647
10	Flow discharge Q	0.059427266
11		
12	Sequent depth y2	0.8
13		
14	Ratio r = H/h	0.617707425
15		
16	Weir height h	0.386
17	Head H	0.414
18		
19	C	2.22
20	Weir discharge Qw	0.05913621
21		
22	Error	-0.00029106
23		

Figure 3: Excel weir calculator showing inputs, Goal Seek parameters, and converged outputs

5.2 Phase 1 — Weir-Only CFD Validation Case (`weirOverflow_final`)

A standalone weir-only case was run to verify that the weir geometry passes the correct discharge Q, check the tailwater depth y_2, and confirm the setup before introducing full jump dynamics.

Case Directory Structure

```
weirOverflow_final/
|-- 0.orig/
|   |-- U, p_rgh, alpha.water, k, omega, nut
|   `-- include/initialConditions
|-- constant/
|   |-- transportProperties, turbulenceProperties, g
|-- system/
|   |-- blockMeshDict, setFieldsDict, controlDict
|   |-- fvSchemes, fvSolution, decomposeParDict
|-- Allrun
`-- Allclean
```

Listing 1: Weir-only case directory structure

Geometry and Mesh

Table 11: Weir-only case: domain geometry

Parameter	Value	Unit
Total channel length	5.1	m
Channel height	1.0	m
Channel depth	0.1	m
Weir position (start)	$x = 2.5$	m
Weir position (end)	$x = 2.6$	m
Weir height h	0.386	m
Weir thickness	0.1	m

```
weirHeight   0.386;

vertices
(
    (0    0           0)   //  0
    (0    $weirHeight 0)   //  1
    (0    1           0)   //  2
    (2.5  1           0)   //  3
    (2.6  1           0)   //  4
    (5.1  1           0)   //  5
```

```
    (5.1  $weirHeight 0)   //  6
    (5.1  0           0)   //  7
    (2.5  0           0)   //  8
    (2.5  $weirHeight 0)   //  9
    (2.6  $weirHeight 0)   // 10
    (2.6  0           0)   // 11
    // z = 0.1 copies: vertices 12-23
);

blocks
(
    hex (0 8 9 1 12 20 21 13)   (120 20 1) simpleGrading (1 1 1)
    hex (1 9 3 2 13 21 15 14)   (120 40 1) simpleGrading (1 1 1)
    hex (9 10 4 3 21 22 16 15)  (5   40 1) simpleGrading (1 1 1)
    hex (10 6 5 4 22 18 17 16)  (120 40 1) simpleGrading (1 1 1)
    hex (11 7 6 10 23 19 18 22) (120 20 1) simpleGrading (1 1 1)
);
```

Listing 2: Weir-only `blockMeshDict`

Fluid Properties

```
phases   (water air);
water { transportModel Newtonian; nu 1e-06;    rho 1000; }
air   { transportModel Newtonian; nu 1.48e-05; rho 1;    }
sigma   0.07;
```

Listing 3: `constant/transportProperties`

Initial Conditions

```
defaultFieldValues ( volScalarFieldValue alpha.water 0 );
regions
(
    boxToCell
    {
        box (0 0 0) (2.5 0.2 0.1);
        fieldValues ( volScalarFieldValue alpha.water 1 );
    }
);
```

Listing 4: `system/setFieldsDict`

Boundary Conditions

```
inletFlowRate    0.0594;
pressure         0;
```

```
turbulentKE      1e-4;
turbulentOmega   10;
```

Listing 5: `0.orig/include/initialConditions`

Table 12: Weir-only case: boundary conditions for **U**, **p_rgh**, and α.water

Patch	**U**	α.**water**
inlet	`flowRateInletVelocity` $Q = 0.0594$ m^3/s	`variableHeightFlowRate`
outlet	`pressureInletOutletVelocity`	`inletOutlet` (inletValue = 0)
atmosphere	`pressureInletOutletVelocity`	`inletOutlet` (inletValue = 0)
weir / walls	`noSlip`	`zeroGradient`
frontAndBack	`empty`	`empty`

Key Differences from the Main Jump Case

Table 13: Key differences between weir-only and main jump cases

Setting	**Weir-only case**	**Main jump case**
Channel length	5.1 m	19.1 m
Inlet BC (U)	`flowRateInlet-Velocity`	`fixedValue`
Inlet BC (α)	`variableHeight-FlowRate`	`fixedValue` 1
Outlet `p_rgh`	`fixedValue` 0	`totalPressure`
Laplacian scheme	`corrected`	`orthogonal`
snGrad scheme	`corrected`	`orthogonal`
End time	120 s	20 s
Write interval	0.5 s	0.1 s
k wall function	`kqRWallFunction`	`kLowReWallFunction`
Run mode	Serial (`Allrun`)	Parallel (12 cores)

Run Script

```
restore0Dir
blockMesh
setFields
interFoam
```

Listing 6: `Allrun` script for weir-only case

Validation Results

Table 14: Weir-only CFD validation: results across both iterations

Iteration	**Weir height h (m)**	**CFD y_2 (m)**	**Outcome**
1 (initial)	0.495	0.90	Submerged; h too large
2 (corrected)	0.386	0.81	$y_2 > y_{2,\text{anal.}}$; accepted

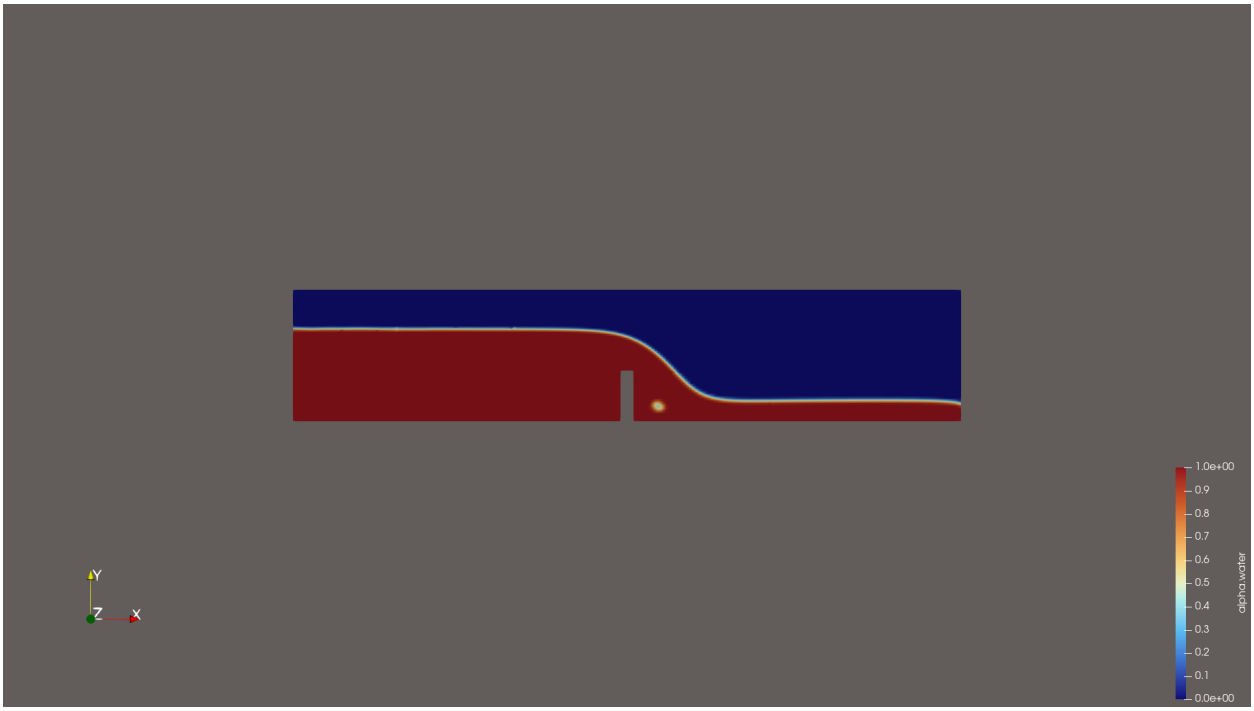

Figure 4: Phase fraction α in the weir-only case at steady state

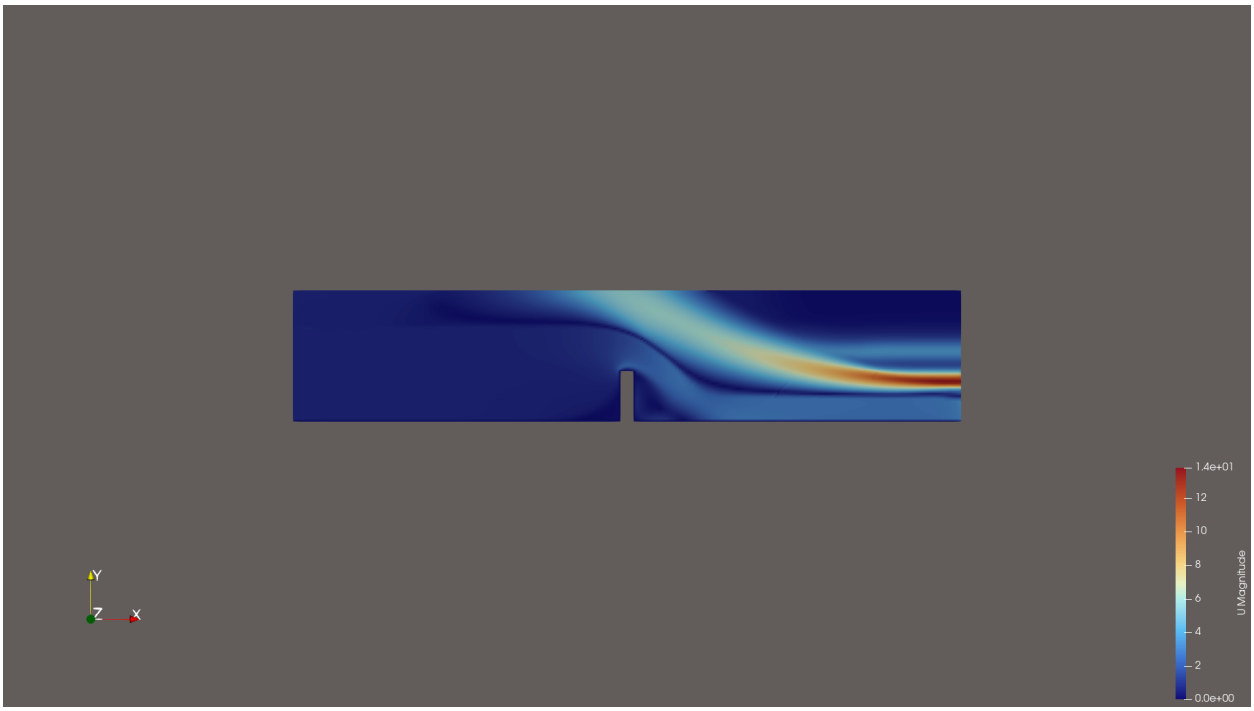

Figure 5: Velocity magnitude in the weir-only case showing the high-speed jet over the weir crest

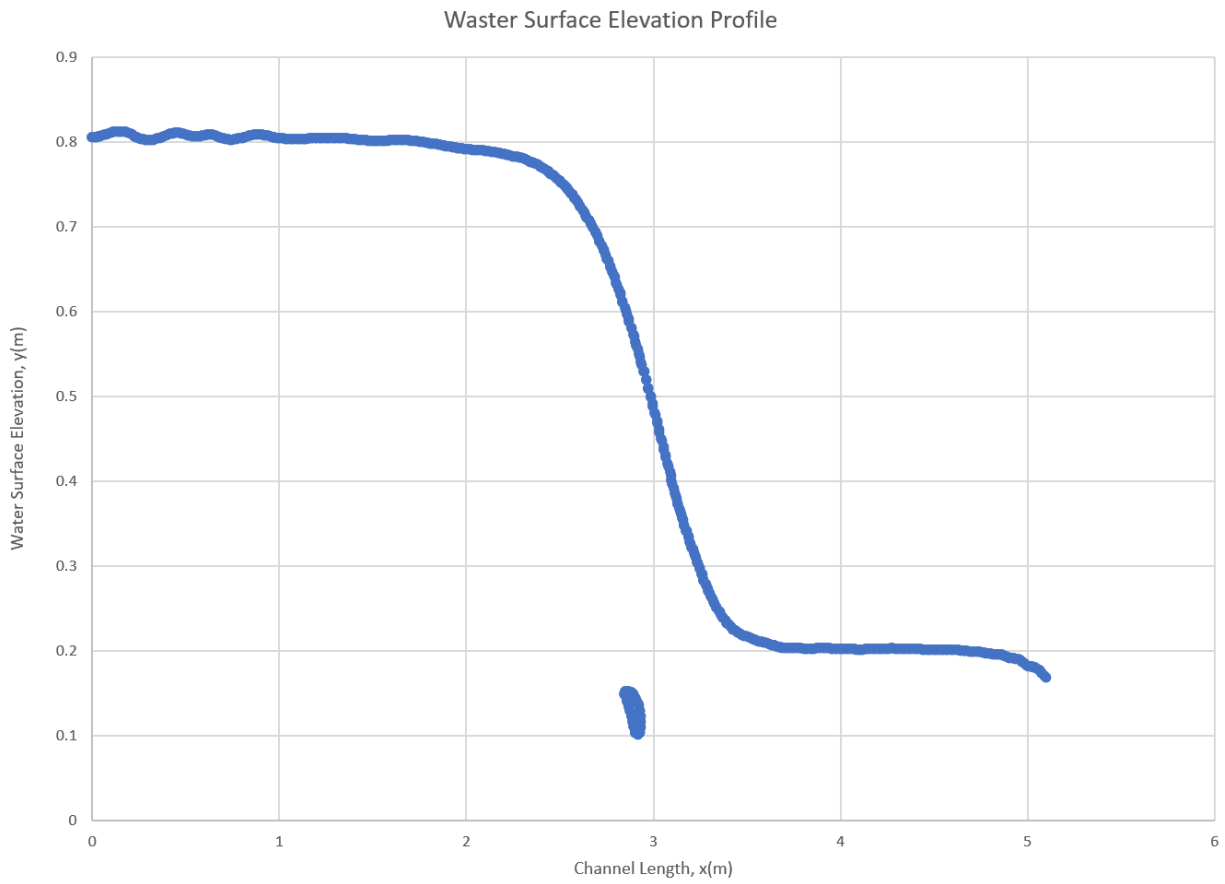

Figure 6: Water surface elevation profile for the weir-only case

5.3 Phase 2 — Hydraulic Jump Case Setup Structure

```
hydraulicJump9_1/
|-- 0/
|   |-- U, p_rgh, alpha.water, k, omega, nut
|-- constant/
|   |-- transportProperties, turbulenceProperties, g
|-- system/
|   |-- blockMeshDict, setFieldsDict, controlDict
|   |-- fvSchemes, fvSolution, decomposeParDict
|   `-- include/hydraulicParameters
```

Listing 7: OpenFOAM case directory (hydraulicJump9_1)

```
weirHeight          0.386;
weirPosition        15;
weirStart           $weirPosition;
weirEnd             #calc "$weirStart + 0.1";

jumpLength          4.88;
```

```
afterWeirLength     4.0;
channelLength       #calc "$weirEnd + $afterWeirLength";

waterLevelStart     #calc "$weirPosition - $jumpLength";
```

Listing 8: system/include/hydraulicParameters

5.4 Control Dictionary

```
application     interFoam;
startFrom       startTime;
startTime       0;
stopAt          endTime;
endTime         20;
deltaT          0.001;
writeControl    adjustableRunTime;
writeInterval   0.1;
adjustTimeStep  yes;
maxCo           0.5;
maxAlphaCo      0.5;
maxDeltaT       0.01;
```

Listing 9: Key entries in `controlDict`

5.5 Running Procedure

The simulation was run in parallel on 12 cores using OpenMPI with Scotch domain decomposition.

```
blockMesh
setFields
checkMesh
decomposePar
mpirun -np 12 interFoam -parallel 2>&1 | tee log.interFoam
reconstructPar
paraFoam
```

Listing 10: Parallel run workflow

5.6 Total Simulation Time

The simulation ran to $t = 20$ s of physical time. With adaptive time-stepping constrained by $Co \leq 0.5$ and $\Delta t_{\max} = 0.01$ s, roughly 3000–5000 time steps were taken. The hydraulic jump reached a quasi-steady state by $t \approx 10$ s and held stable through the end.

5.7 OpenFOAM Version

OpenFOAM v2412 (released December 2024, OpenCFD Ltd / ESI Group) was used throughout. A few version-specific points worth noting:

- `nutkSSTWallFunction` is absent in v2412; `nutkWallFunction` is the correct replacement.
- `div(phi,alpha)` is the v2412 naming convention.
- `div(phirb,alpha)` with `interfaceCompression` is required for VOF interface sharpening.

Environment: WSL2 Ubuntu on Windows, 12-core MPI, ParaView 6.0.1.

6 Results and Discussions

6.1 Validation

Weir Calculator Validation (Phase 1)

At the final weir height $h = 0.386$ m, the CFD-predicted tailwater depth was $y_2 = 0.81$ m — 1.25% above the analytical conjugate depth of 0.80 m, putting the jump in submerged conditions as intended. The discharge error at convergence was -1.52×10^{-4} m^3/s, or 0.26% of Q.

Analytical Targets (Phase 2)

Table 15: Analytical flow parameters and simulation targets

Parameter	**Formula / Source**	**Value**	**Unit**
Flow rate Q	given	0.0594	m^3/s
Channel width b	given	0.10	m
Unit discharge $q = Q/b$		0.594	m^2/s
Inlet depth y_1	given	0.10	m
Inlet velocity U_1		5.94	m/s
Froude number Fr_1	Eq. (7)	6.00	–
Jump type	classification	Steady	–
Conjugate depth y_2	Eq. (6)	0.80	m
Jump length L_j	Eq. (10)	4.88	m
Specific energy E_1	Eq. (9)	1.899	m
Energy dissipation ΔE	Eq. (8)	1.072	m
Jump efficiency η	Eq. (9)	56.4	%

Table 16: Hydraulic jump classification by inlet Froude number

Fr_1 **Range**	**Type**	**Description**
1.0 – 1.7	Undular	Gentle undulations, no roller
1.7 – 2.5	Weak	Small surface rollers
2.5 – 4.5	Oscillating	Unstable oscillating jet
4.5 – 9.0	**Steady**	**Well-defined roller, 45–70% dissipation**
> 9.0	Strong	Rough surface, up to 85% dissipation

6.2 Flow Field Visualisation

Post-processing was done in ParaView 6.0.1 on a mid-plane slice at $z = 0.05$ m.

6.2.1 Velocity

The velocity magnitude contours show the supercritical jet ($U \approx 5.94$ m/s) running along the channel floor for $0 < x < 10$ m. As it hits the subcritical tailwater pool upstream of the weir, it decelerates and spreads vertically through the hydraulic jump. Post-jump velocity drops to approximately 0.74 m/s at depth $y_2 = 0.80$ m by continuity.

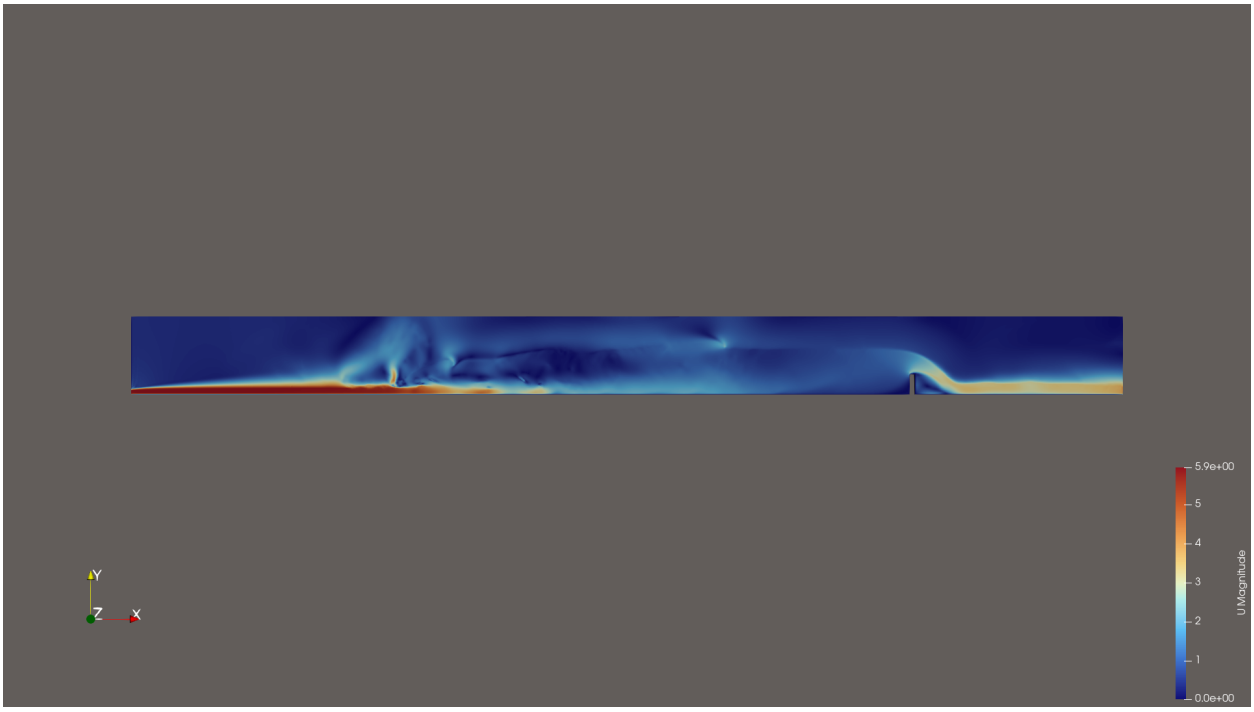

Figure 7: Velocity magnitude contours at $t = 10$ s

6.2.2 Pressure

The modified pressure p_{rgh} is low upstream of the jump and rises downstream as depth increases to $y_2 \approx 0.81$ m. A high-pressure stagnation zone appears on the upstream face of the weir.

6.2.3 Streamlines

Stream Tracer seeds placed at the inlet follow the supercritical jet along the floor, the upward deflection at the jump toe near $x \approx 10.1$–11.5 m, and recirculation in the roller. Streamlines over the weir crest show the free nappe expected from a sharp-crested weir.

6.2.4 Vortices and Wake

Velocity vectors from the Glyph filter confirm a recirculation roller spanning roughly 3.5–5 m, consistent with the predicted range of 3.46–4.88 m. No secondary rollers were found, as expected for a steady-jump classification.

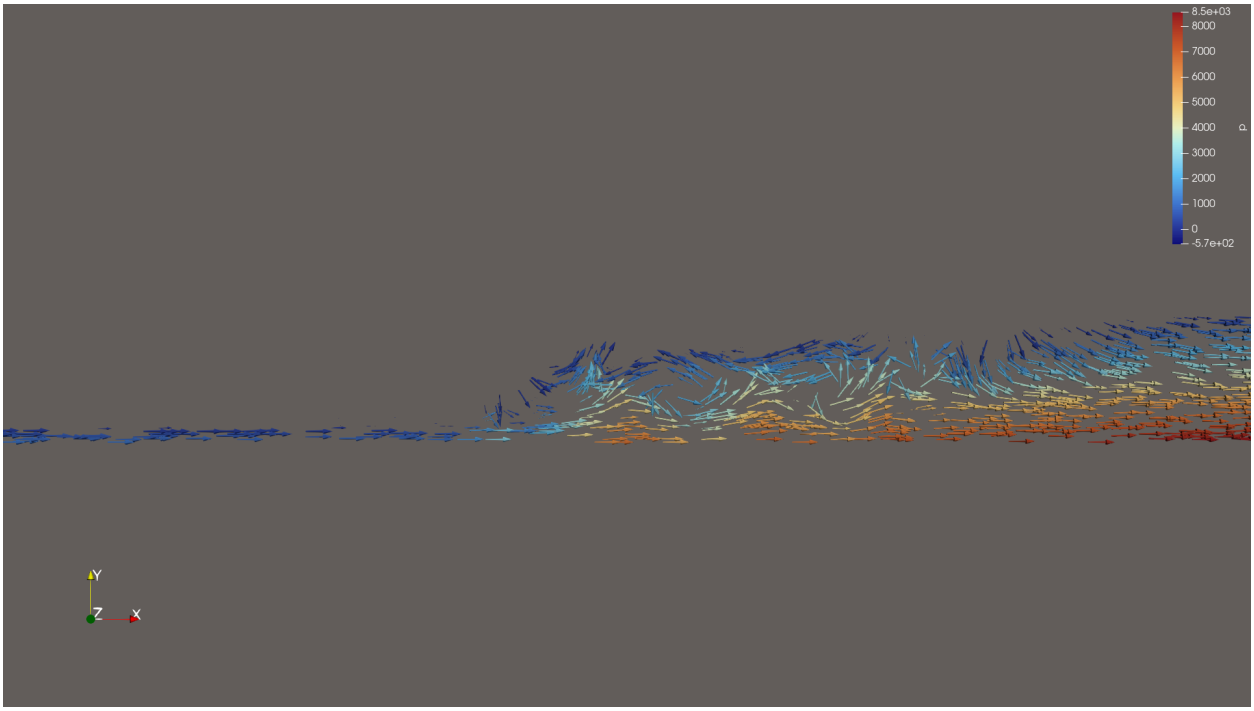

Figure 8: Velocity vectors (Glyph filter) at $t = 10$ s showing early-stage recirculation roller

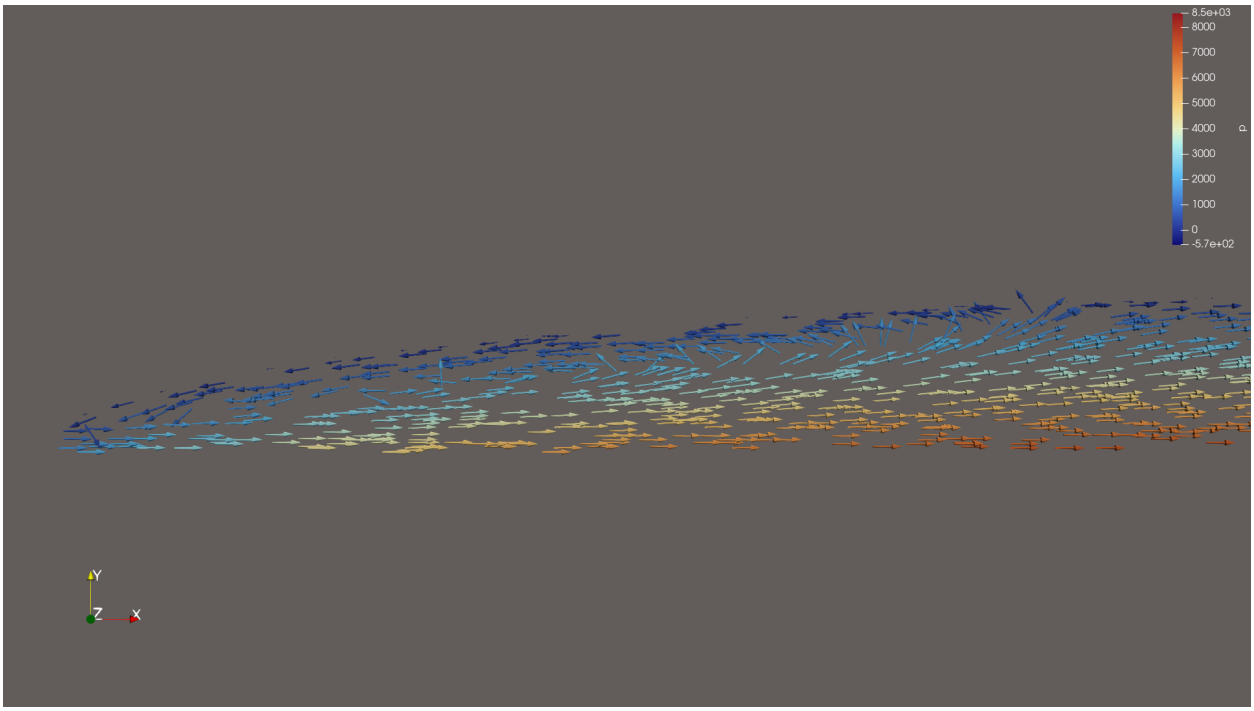

Figure 9: Velocity vectors at fully developed state showing established recirculation roller

6.3 Force Analysis

No aerodynamic bodies are present; thrust, lift, and force coefficient analysis do not apply. The primary engineering result is $\Delta E = 1.072$ m and $\eta = 56.4\%$.

6.4 Parametric Analysis

A formal parametric study was not conducted. All runs used fixed parameters ($Fr_1 = 6.0$, $y_1 = 0.10$ m, $h = 0.386$ m, $x_{\text{weir}} = 15.0$ m). Future work will vary Fr_1 across the steady-jump regime.

6.5 Submerged Jump Condition

The hydraulic jump was confirmed as **submerged**: the simulated tailwater depth $y_t = 0.81$ m exceeds the analytical conjugate depth $y_2 = 0.80$ m, a direct consequence of the weir raising the downstream water level. The academic guide confirmed this classification is physically correct.

7 Discussion and Conclusion

7.1 Summary of Findings

1. The weir height $h = 0.386$ m was determined through a two-iteration Excel procedure using the Rehbock formula, then validated against weir-only CFD.
2. A 2-D hydraulic jump at $Fr_1 = 6.0$ was simulated with `interFoam` and `kOmegaSST` in OpenFOAM v2412.
3. The VOF free surface reproduced the supercritical-to-subcritical transition upstream of the weir at $x = 15.0$ m.
4. Velocity vectors confirmed a recirculation roller spanning approximately 3.5–5 m, close to Peterka's prediction of 4.88 m.
5. Analytical energy dissipation efficiency was 56.4%, within the expected 45–70% range for steady jumps.

7.2 Limitations of the Study

- The 2-D setup ignores three-dimensional effects such as secondary currents and spanwise roller instabilities.
- High-Reynolds wall functions were used; near-wall meshes with $y^+ < 1$ would give better accuracy near boundaries.
- No direct comparison with experimental data was made.
- The Rehbock coefficient is valid for $H/h \leq 10$; the present ratio $H/h \approx 0.617$ is well within range.

7.3 Future Work

- Extend to a 3-D simulation to capture spanwise effects.
- Validate against published experimental data (e.g., Murzyn & Chanson 2009).
- Run a parametric study over $Fr_1 \in \{4.5, 6.0, 9.0\}$.
- Refine the mesh near the free surface and roller zone.

References

[1] Bélanger, J.-B. (1838). *Essai sur la solution numérique de quelques problèmes relatifs au mouvement permanent des eaux courantes*. Carilian-Goeury, Paris.

[2] Peterka, A. J. (1958). *Hydraulic design of stilling basins and energy dissipators* (Engineering Monograph No. 25). US Bureau of Reclamation.

[3] Rehbock, T. (1929). Wassermessung mit scharfkantigen Wehrschnellen. *Zeitschrift des Vereines Deutscher Ingenieure*, *73*(24), 817–823.

[4] Chaudhry, M. H. (2008). *Open-channel hydraulics* (2nd ed.). Springer.

[5] Menter, F. R. (1994). Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, *32*(8), 1598–1605.

[6] Bayon, A., Valero, D., García-Bartual, R., Valles-Morán, F. J., & Lopez-Jiménez, P. A. (2016). Performance of OpenFOAM in hydraulic jump simulation. *Journal of Hydro-environment Research*, *14*, 22–36.

[7] Witt, A., Gulliver, J., & Shen, L. (2018). Simulating air entrainment and vortex dynamics in a hydraulic jump. *International Journal of Multiphase Flow*, *103*, 301–317.

[8] OpenCFD Ltd. (2024). *OpenFOAM v2412 documentation*. https://www.openfoam.com/documentation/

[9] FOSSEE, IIT Bombay. (2024). *OpenFOAM case study projects*. https://fossee.in/

A Mesh Details and Additional Plots

A.1 Complete `blockMeshDict` — Main Jump Case

```
#include "include/hydraulicParameters"

scale   1;

vertices
(
    (0              0            0)   //  0
    (0              0.1          0)   //  1
    (0              $weirHeight  0)   //  2
    (0              1.5          0)   //  3
    ($weirStart     1.5          0)   //  4
    ($weirEnd       1.5          0)   //  5
    ($channelLength 1.5          0)   //  6
    ($channelLength $weirHeight  0)   //  7
    ($channelLength 0            0)   //  8
    ($weirStart     0            0)   //  9
    ($weirStart     0.1          0)   // 10
    ($weirStart     $weirHeight  0)   // 11
    ($weirEnd       $weirHeight  0)   // 12
    ($weirEnd       0            0)   // 13
    (0              0            0.1) // 14
    (0              0.1          0.1) // 15
    (0              $weirHeight  0.1) // 16
    (0              1.5          0.1) // 17
    ($weirStart     1.5          0.1) // 18
    ($weirEnd       1.5          0.1) // 19
    ($channelLength 1.5          0.1) // 20
    ($channelLength $weirHeight  0.1) // 21
    ($channelLength 0            0.1) // 22
    ($weirStart     0            0.1) // 23
    ($weirStart     0.1          0.1) // 24
    ($weirStart     $weirHeight  0.1) // 25
    ($weirEnd       $weirHeight  0.1) // 26
    ($weirEnd       0            0.1) // 27
);

blocks
(
    hex (0 9 10 1 14 23 24 15)  (800 5  1) simpleGrading (1 1 1)
    hex (1 10 11 2 15 24 25 16) (800 15 1) simpleGrading (1 1 1)
    hex (2 11 4 3 16 25 18 17)  (800 60 1) simpleGrading (1 1 1)
    hex (11 12 5 4 25 26 19 18) (5   60 1) simpleGrading (1 1 1)
    hex (12 7 6 5 26 21 20 19)  (180 60 1) simpleGrading (1 1 1)
    hex (13 8 7 12 27 22 21 26) (180 20 1) simpleGrading (1 1 1)
```

```
);

patches
(
    patch inlet   ( (0 1 15 14) )
    patch outlet  ( (7 8 22 21) (6 7 21 20) )
    wall weir
    (
        (9  10 24 23) (10 11 25 24)
        (11 25 26 12) (12 26 27 13)
    )
    wall walls
    (
        (2 16 15 1)  (3 17 16 2)
        (0 14 23 9)  (13 27 22 8)
    )
    patch atmosphere
    (
        (3 17 18 4) (4 18 19 5) (5 19 20 6)
    )
    empty frontAndBack
    (
        (0  1  10  9)  (1  2  11 10)
        (2  3   4 11)  (11  4  5 12)
        (12  5  6  7)  (13 12  7  8)
        (14 15 24 23)  (15 16 25 24)
        (16 17 18 25)  (25 18 19 26)
        (26 19 20 21)  (27 26 21 22)
    )
);
```

Listing 11: blockMeshDict (hydraulicJump9_1)

A.2 `checkMesh` Quality Report

Table 17: `checkMesh` output summary (hydraulicJump9_1)

Metric	Value
Total cells	78,700
Points	159,572
Internal faces	156,315
Boundary patches	6
Cell type	Hexahedral (100%)
Domain bounding box	(0 0 0) to (19.1 1.5 0.1)
Max aspect ratio	1.197
Min face area	3.48×10^{-4} m^2
Max face area	2.22×10^{-3} m^2
Max non-orthogonality	0°
Average non-orthogonality	0°
Max skewness	9.59×10^{-13}
Overall assessment	**Mesh OK**

A.3 Fixes Applied During Case Development (hydraulicJump3–9)

Table 18: Debugging fixes across case iterations

Fix Applied	Issue Resolved
`nutkWallFunction` (not `nutkSSTWallFunction`)	v2412 does not include an SST-specific wall function
`wallDist meshWave` added to `fvSchemes`	`kOmegaSST` requires a wall-distance field
`div(phi,alpha)` (not `div(phi,alpha.water)`)	Correct v2412 naming convention
`div(phirb,alpha) + interfaceCompression`	Required for VOF interface sharpening
`orthogonal` Laplacian & snGrad schemes	Fully orthogonal mesh; no correction needed
`nNonOrthogonalCorrectors = 0`	Consistent with fully orthogonal mesh
`fixedValue` inlet BC	`variableHeightFlowRateInletVelocity` crashed at $t = 18$ s
GAMG pressure solver	Replaced DICPCG; $\sim$10$\times$ fewer iterations

Table 18 continued

Fix Applied	Issue Resolved
Removed stray vertex at $(10.1, 1, 0)$	Non-ortho = 89°, skewness = 48.9; simulation crash
`inletValue 1` at outlet for α	Backflow must bring water, not air

A.4 ParaView Post-Processing Operations

- **Free surface** — Contour filter on `alpha.water` at iso-value 0.5.
- **Velocity recirculation** — Glyph filter (Arrow type); confirms roller with negative U_x above the jet.
- **Streamlines** — Stream Tracer seeded at inlet.
- **2-D slice** — Slice at $z = 0.05$ m (mid-plane).
- **WSE profile** — Plot Over Line along the channel centreline.

B Extra Contour Plots

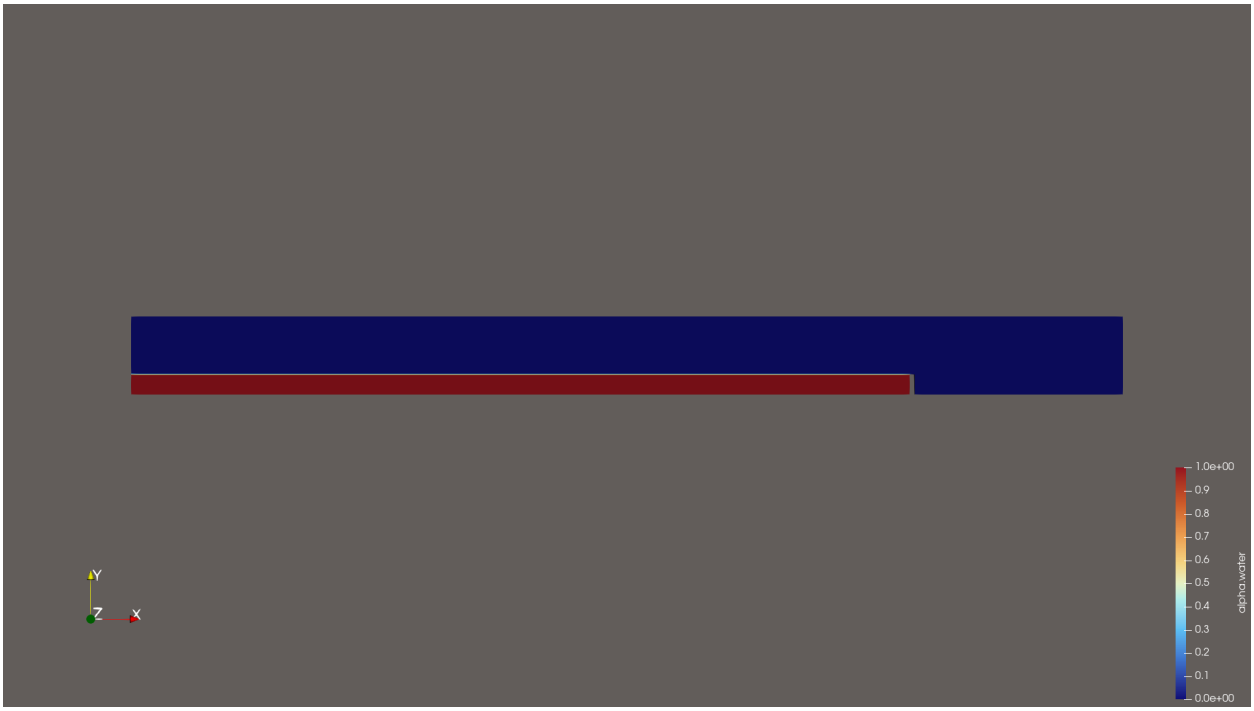

Figure 10: Phase fraction α at $t = 0$ s showing the pre-seeded water regions from `setFields`

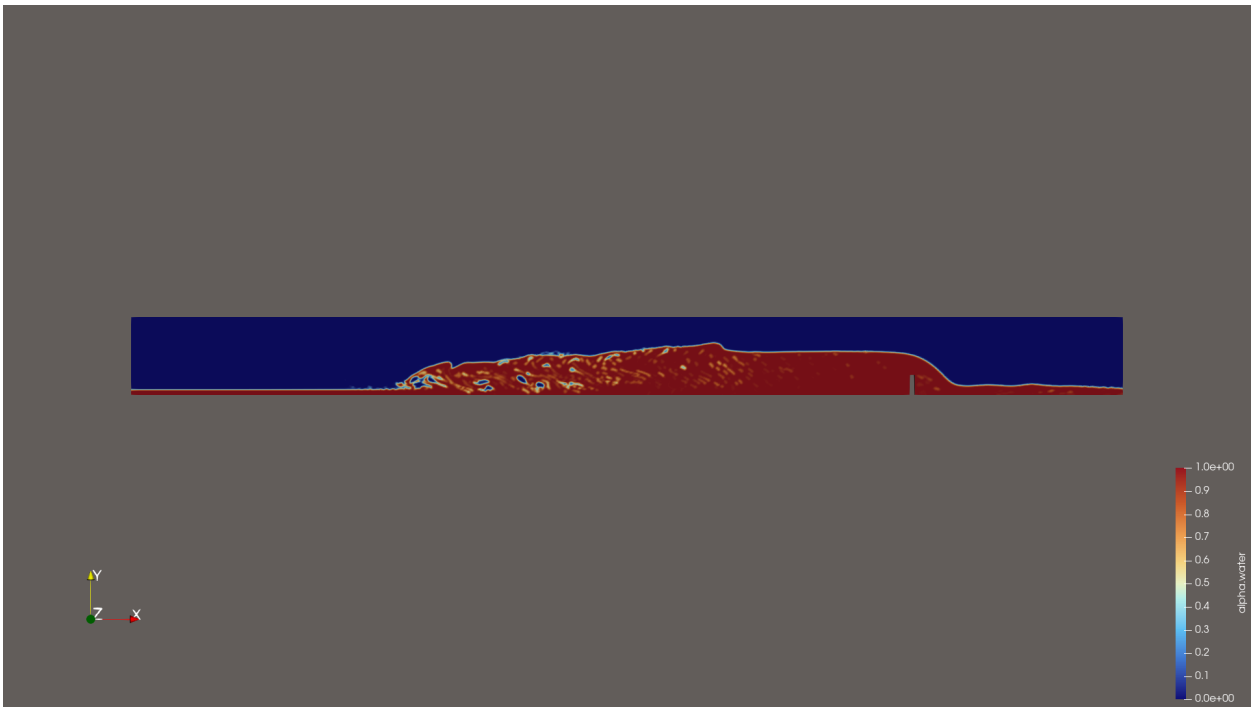

Figure 11: Phase fraction α at $t = 10$ s showing hydraulic jump formation upstream of the weir

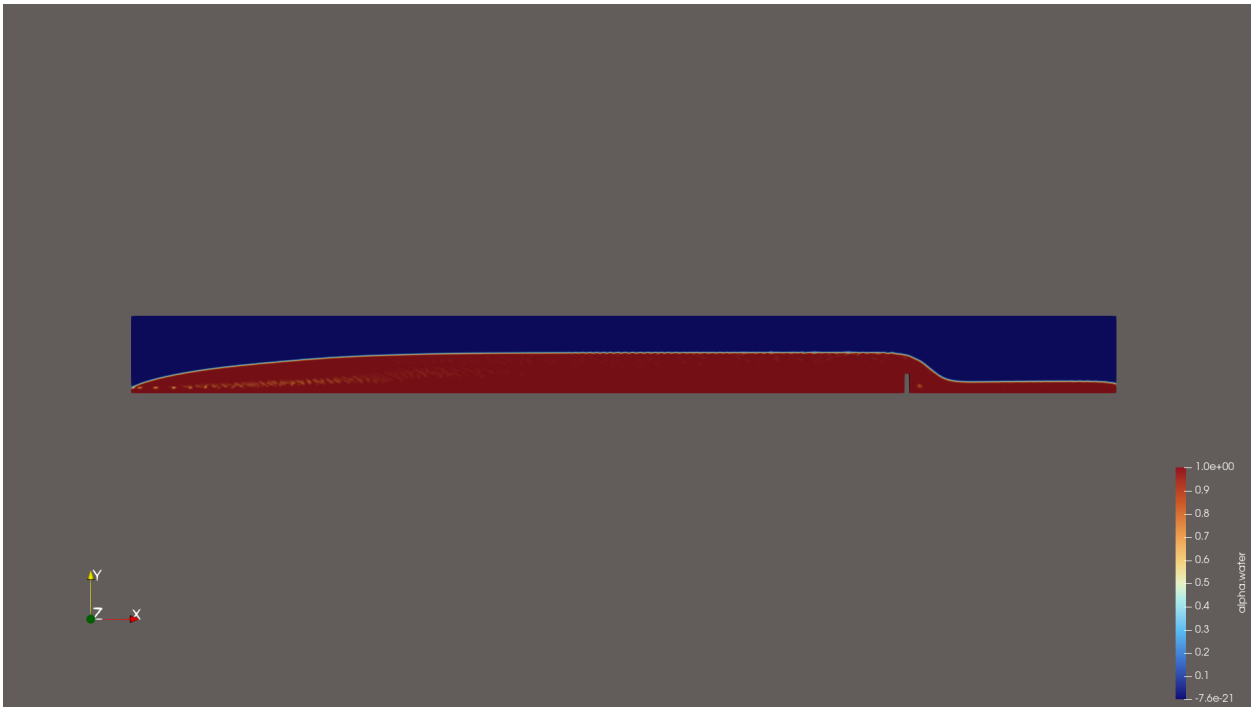

Figure 12: Phase fraction α at $t = 7.5$ s showing transitional jump development

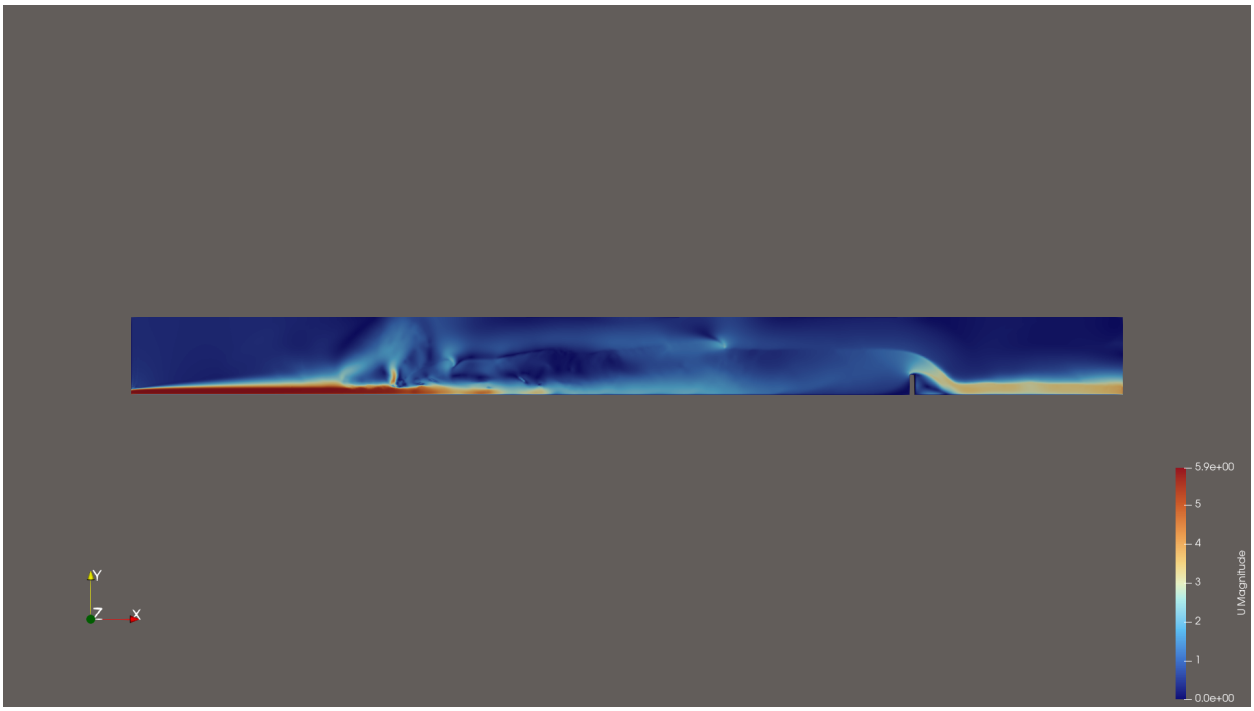

Figure 13: Velocity magnitude contours at $t = 10$ s

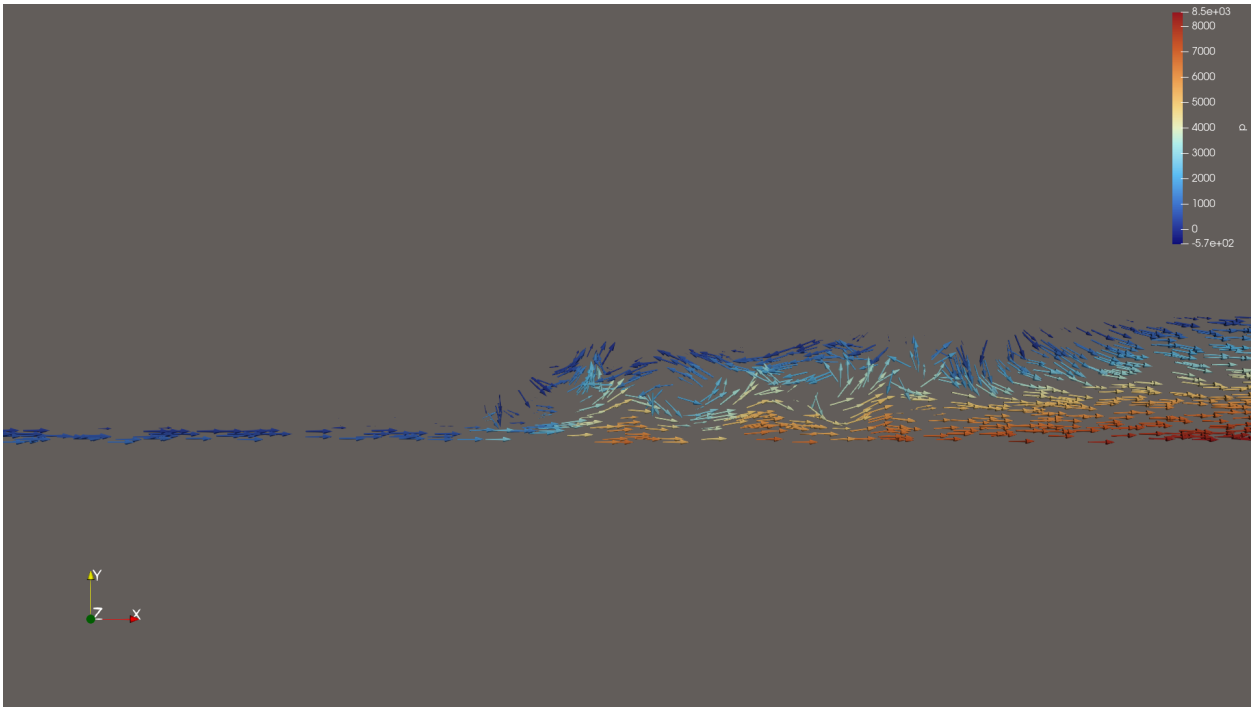

Figure 14: Velocity vectors (Glyph filter) at $t = 10$ s

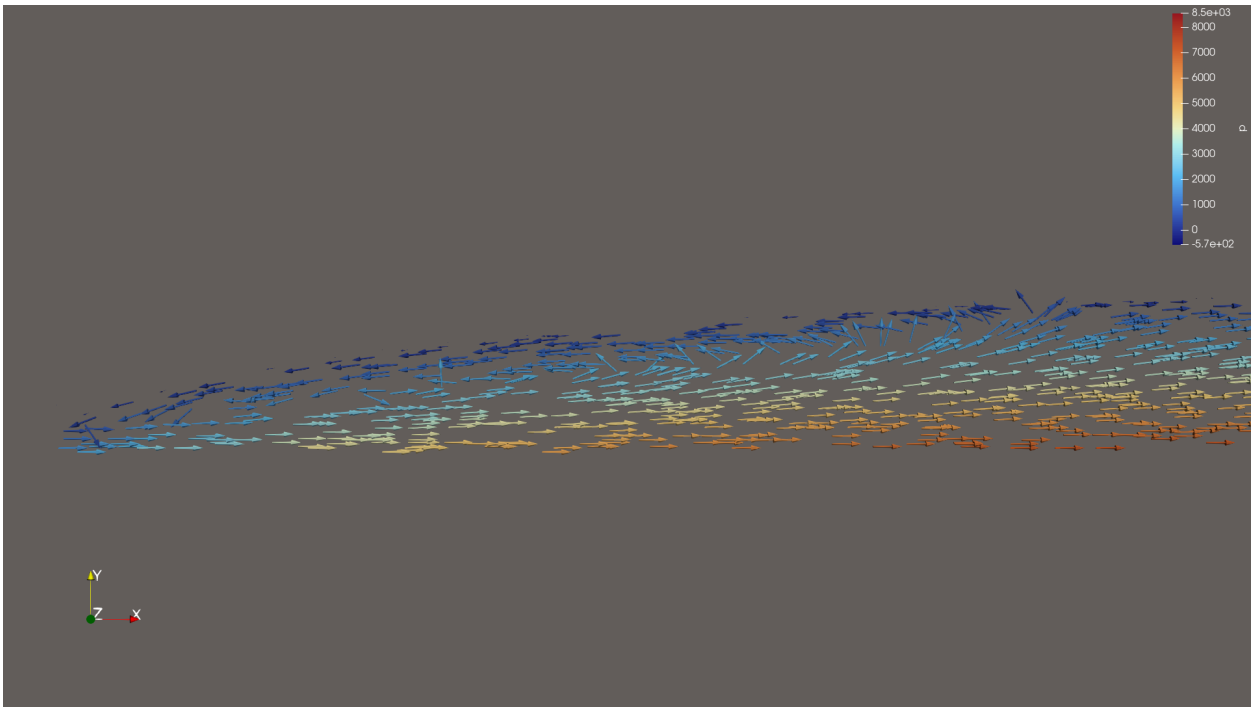

Figure 15: Velocity vectors (Glyph filter) at fully developed state

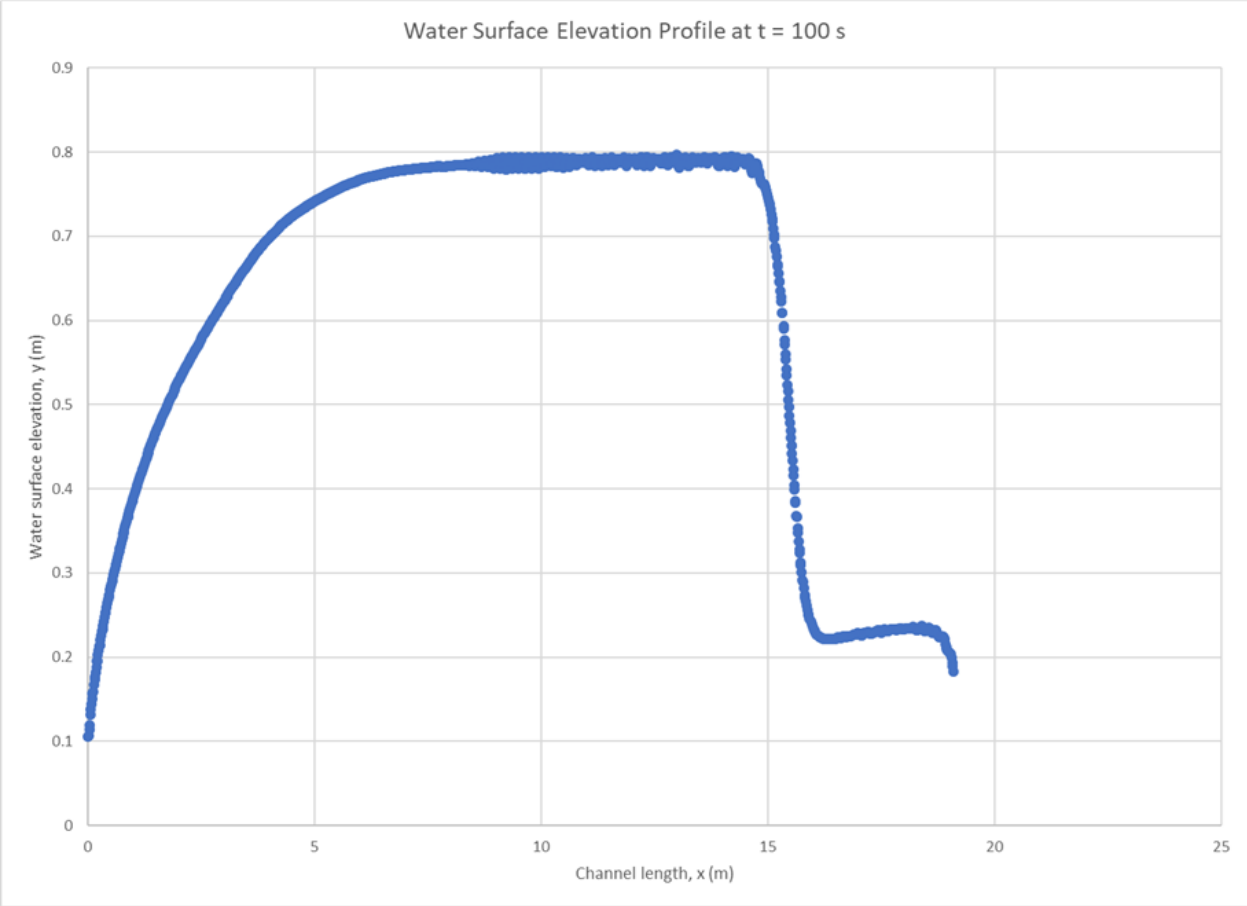

Figure 16: Water surface elevation profile along the channel centreline