

Waves and Oscillatons in Fluids

A Seminar Report
submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
by

Palas Kumar Farsoiya
(Roll No: 144026002)



Chemical Engineering
Indian Institute of Technology Bombay
Mumbai 460076 (India)

August 2016

Declaration

This is to certify that the annual progress seminar report titled **Waves and Oscillation in Fluids** which is submitted by me in partial fulfillment of the requirement for the award of Ph.D. (*philosophiae doctor*) to Department of Chemical Engineering, IIT Bombay, Mumbai comprises the work carried out by me and due acknowledgment has been made whenever the work described is based on the findings of other investigators.

Palas Kumar Farsoiya

ACCEPTANCE CERTIFICATE

Seminar Report

Department of Chemical Engineering
Indian Institute of Technology, Bombay

The Seminar report titled **Droplet Impact Dynamics** submitted by Palas Kumar Farsoiya (Roll No. 144026002) may please be accepted for being evaluated.

Prof. Ratul Dasgupta
(Research Supervisor)

Acknowledgement

I would like to acknowledge my supervisor Prof. Ratul Dasgupta for providing me VOF code for further development. Further, he has appreciated me in developing concepts and manifesting them in codes. He has time to time monitored my progress and provided with valuable aids in terms of suggestions, resources, lectures and even disagreement. His efforts have enormously benefited me.

I would also like to thank all my friends and lab mates for providing their valuable insights, friendly guidance and support whenever required. I would like to acknowledge Kim Motoyoshi Kalland (Software Engineer at Hue AS, Oslo, Norway) to help me in implementing surface tension and volume fraction generation algorithms.

My wonderful family, which has been my constant source of motivation and inspiration.

Abstract

Contents

1	Volume of Fluid method	1
1.1	Introduction	1
1.2	The Volume of Fluid method	1
1.2.1	Interface Reconstruction by Least Squares of Volume of Fluid In- terface Reconstruction Algorithm (LVIRA)	3
1.2.2	Advection	10
1.2.3	Volume Fraction Calculation	13
1.3	Verification	15
1.3.1	Advection of circle in translational flow	15
1.3.2	Advection test for solid body rotation	16
1.3.3	Shear Test	17
1.3.4	Calculation of error	17
1.4	Conclusion	21
2	Navier Stokes Solver for two phase flow	23
2.1	Governing Equations	23
2.1.1	Mass conservation	23
2.1.2	Momentum Conservation	24
2.1.3	Non-Dimensionalisation of governing equations	26
2.1.4	Integral form of governing equation	26
2.2	Discretisation of governing equations	27
2.2.1	Grid	27
2.2.2	Integration in time	27
2.2.3	Discretisation of advection terms	28
2.2.4	Diffusion terms	30

2.2.5	Boundary conditions for velocity	31
2.3	Pressure Poisson Equation(PPE)	32
2.3.1	Successive Over Relaxation method (SOR)	33
2.3.2	Boundary Conditions for PPE	34
2.4	Coupling VOF and Navier-Stokes solver	34
2.5	Verification	36
2.5.1	Lid Driven Cavity test	37
2.5.2	Falling Droplet	37
2.5.3	Rayleigh-Taylor Instability	37
2.6	Conclusion	42
3	Multigrid Methods	45
3.1	Introduction	45
3.2	General Iterative Methods	45
3.2.1	Convergence analysis of Gauss-Siedel method	47
3.3	Algorithm of Multigrid method	51
3.4	Sample Problem	51
3.4.1	One dimensional Poisson's equation	53
3.4.2	Two dimensional Poisson's equation	54
3.5	Discretization	55
3.6	Validation	55
3.7	Comparison with SOR Parallel	57
4	Surface Tension	59
4.0.1	Height Function	59

List of Tables

1.1	Errors for various tests	19
4.1	Comparison of spurious currents, The capillary number is computed by different surface tension methods	62
4.2	Comparison of growth rate n among the various numerical methods(PROST, CLSVOF, K_8 and HF) and analytical results with respect to relative im- portance of surface tension Φ	62

List of Figures

1.1	Interface normal direction	4
1.2	Area and angle region	5
1.3	Area of the triangle made by dark fluid in the cell	5
1.4	Area of a trapezium made by the dark fluid in the cell	6
1.5	When the cell is reoriented to make normal lie in first quadrant	6
1.6	Perpendicular distance in a triangle	7
1.7	Perpendicular distance for a trapezium	8
1.8	Perpendicular distance for compliment of the triangle	8
1.9	Normal form of a line in the cell	9
1.10	Extrapolation of the line in 3 X 3 cell	10
1.11	Cases for a triangle in flux calculation	11
1.12	Flux calculation for trapezium for $\theta < \frac{\pi}{4}$	11
1.13	Different cases for flux calculation for trapezium	12
1.14	Different cases for flux calculation for compliment of a triangle	12
1.15	Reconstruction of a circular interface by LVIRA	14
1.16	Advection of the interface by Youngs operator split algorithm	15
1.17	Advection test for velocity field $u=1, v=0$	16
1.18	Advection test for velocity field $u=2, v=1$	16
1.19	Advection test result for solid body rotation	18
1.20	Comparison with Rudman (1997) results. (Red LVIRA and Blue Rudman (1997) data)	19
1.21	Advection test result for shear velocity field	20
1.22	Comparison with Gerlach <i>et al.</i> (2006) results. (Red LVIRA and Blue Gerlach <i>et al.</i> (2006) data)	21

2.1	An arbitrary control volume Leal (2007)	23
2.2	Staggered grid for x-velocity(Blue),y-velocity(Green) and pressure(Gray)	27
2.3	Location of nodes, blue square:x-velocity, red square:boundary x-velocity, green triangle:y-velocity, red triangle:boundary y-velocity,black dot:pressure	28
2.4	Three sub stencils for reconstruction of $u_{i+1/2}$	29
2.5	Sequence to couple VOF and N-S solver	35
2.6	Blue:Ghia <i>et al.</i> (1982), Red:Present Study(LVIRA)	38
2.7	Stream function and vorticity contours	39
2.8	Falling Droplet test, (Blue-Gerris, Red-Present Study(LVIRA))	40
2.9	Comparison of Rayleigh-Taylor Instability test,(Blue-Rudman (1997), Red- Present Study(LVIRA))	41
2.10	Comparison of Rayleigh-Taylor Instability test,(Blue-Smolianski (2001), Red-Present Study(LVIRA))	43
3.1	The matrix A	46
3.2	Fourier modes	47
3.3	Profile after 10 Gauss-Siedel iterations, Blue - Inital guess, Orange - After 10 iterations	48
3.4	Eigenvalues λ with respect to k	49
3.5	Eigenvalues λ with respect to $\frac{k}{N}$	50
3.6	Eigenvalues λ with respect to N	50
3.7	Multigrid Algorithm	52
3.8	Comparison with Moin(2010)	54
3.9	2D uniform mesh featuring a discretization with a 5-point stencil.	55
3.10	Solver results of for $N = 500$ and $np = 4$; (a) Poisson solution and (b) solution error.	56
3.11	Speedup values for MPI parallel (SOR) Poisson equation solver.	57
3.12	Comparison of SOR and Multigrid Time of execution	58
4.1	Variable Stencil for calculation of height functions	60
4.2	Interface shapes for different values of Φ using Height Function at $\tau = 6$	61
4.3	Comparison with analytical results	63

Chapter 1

Volume of Fluid method

Introduction

In this chapter we present the development of in-house code for simulating a two phase flow. Mathematicians and physicists have studied the dynamics of multiphase flows and as discussed in the previous chapter literature is quite extensive. Governing equations for such flows are not only nonlinear but the position of the interface needs to be found out as a part of solution. Consequently, analytical solutions exist only for very simple problems such as oscillations of bubbles and droplets, inviscid linear waves and steady state motion of bubbles and droplets in Stokes flow. Thus there is a need for numerical solutions been felt by the multiphase research community since late fifties and early sixties. Hirt (1981) came up with the idea of Volume of Fluid interface (VOF) tracking to approximate the free boundaries in numerical simulations. This method is based on a concept of fractional volume of fluid. This is still widely used due to its flexibility and efficiency compared to other methods. The volume of fluid method conserves mass up to machine accuracy, level set methods conventionally have some issues of mass conservation but are better in the reconstruction of curved interfaces and thus easier for implementation of surface tension.

The Volume of Fluid method

This method starts with defining a quantity called as volume fraction F , for each cell in the computational domain. For a binary phase system there is a dark fluid and light fluid. F is the ratio of volume of the dark fluid to the volume of the cell itself. The cells with only light fluid are chosen to have $F = 0$ and whereas those with only dark fluid $F = 1$. For interfacial cells F will have a value between 0 and 1. The volume of fluid when advected

does not change with respect to the fluid parcel. Hence, it satisfies the following relation,

$$\frac{DF}{Dt} = 0 \quad (1.1)$$

The Eulerian form would be,

$$\frac{\partial F}{\partial t} + (u \cdot \nabla)F = 0 \quad (1.2)$$

As interface in Volume of fluid method can be reconstructed in many ways we choose the Least Squares Volume of Fluid Reconstruction Algorithm proposed by Pilliod and Puckett (2004) which is one of the most accurate algorithm available in literature. It is second order accurate in space and can be extended to the 3D. It represents the interface as a line in 2D and a plane in 3D. The VOF method achieves higher accuracy using geometrical techniques to reconstruct and advect the interface. Any interface reconstruction algorithm consists two basic steps:-

1. Interface Reconstruction
2. Advection of F field

Interface Reconstruction by Least Squares of Volume of Fluid Interface Reconstruction Algorithm (LVIRA)

Step I : Obtain F field

At first an F -field is obtained by knowing the initial free surface and then by calculating the F values for each cell. For reconstruction of interface a cell having a value of F between 0 and 1 is located.

Step II : Initial guess of slope

The initial slope of the normal to the interface away from the dark fluid can obtained using (Green-Gauss gradient Gerlach *et al.* (2006)) which is given by

$$N_x = -\frac{1}{\Delta x} [F_{r+1,c+1} + 2F_{r,c+1} + F_{r-1,c+1} - F_{r+1,c-1} - 2F_{r,c-1} - F_{r-1,c-1}] \quad (1.3)$$

$$N_y = -\frac{1}{\Delta y} [F_{r+1,c+1} + 2F_{r+1,c} + F_{r+1,c-1} - F_{r-1,c+1} - 2F_{r-1,c} - F_{r-1,c-1}] \quad (1.4)$$

Step III: Quadrant Identification

To get the interface as a line in 2D, we have to determine its shape and orientation in a cell. First step is to get the normal orientation in the quadrant. Signs of N_x and N_y will determine the quadrant in which the normal points away from the dark fluid present in the cell.

Step IV: Angle Calculation

The angle, θ is the positive acute angle made by the interface with x-axis if the normal points into first and the third quadrant and positive acute angle made by the interface by y-axis if the normal points into the second and fourth quadrant. This definition is necessary because we treat all other quadrants as the I^{st} quadrant by suitable rotation. So for the second and the fourth quadrant θ has to be redefined. Refer to Figure 1.1 for definition of theta. This is required to do since the algorithm is developed in way it works for normal in first quadrant and all other cases has to be modified according to algorithm.

$$\theta = \frac{\pi}{2} - \text{fabs} \left(\text{atan} \left(\frac{N_x}{N_y} \right) \right) \quad (1.5)$$

Step V: Shape Identification

There are four possible shapes in a 2D reconstruction If the volume fraction and θ is fixed then the shape is decided as in Figure 1.2. The various regions are identified as follows.

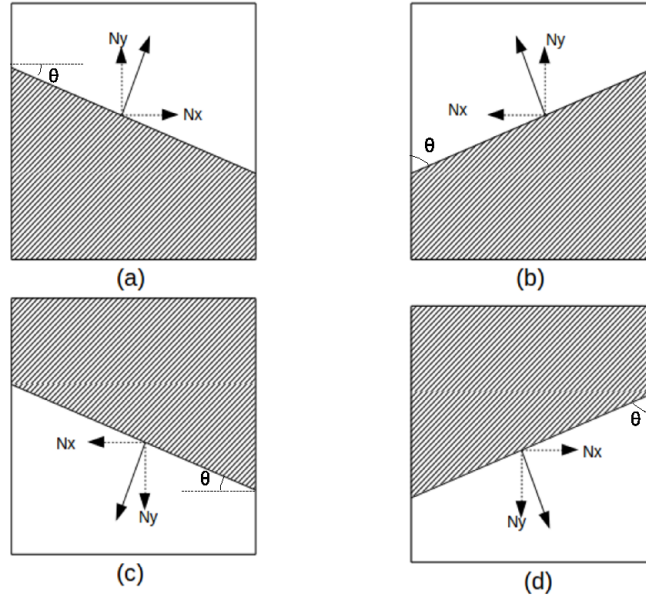


Figure 1.1: (a) I Quad (b) II Quad (c) III Quad (d) IV Quad

Triangle

Area of fluid, $A = \Delta^2 F$

From Figure 1.3, we get,

$$a = \Delta \tan \theta \quad \theta < \frac{\pi}{4}$$

$$b = \frac{\Delta}{\tan \theta} \quad \theta \geq \frac{\pi}{4}$$

$\frac{A}{\Delta^2} \leq \frac{1}{2} \tan \theta$	for, $\theta < \frac{\pi}{4}$
$\frac{A}{\Delta^2} \leq \frac{1}{2 \tan \theta}$	for, $\theta > \frac{\pi}{4}$

For trapezium,

In Figure 1.5, it can be seen that when the cell reoriented to make the normal in first quadrant the θ becomes the angle with the horizontal axis. Hence, the same calculations can be applied to this cell for what is done for first quadrant.

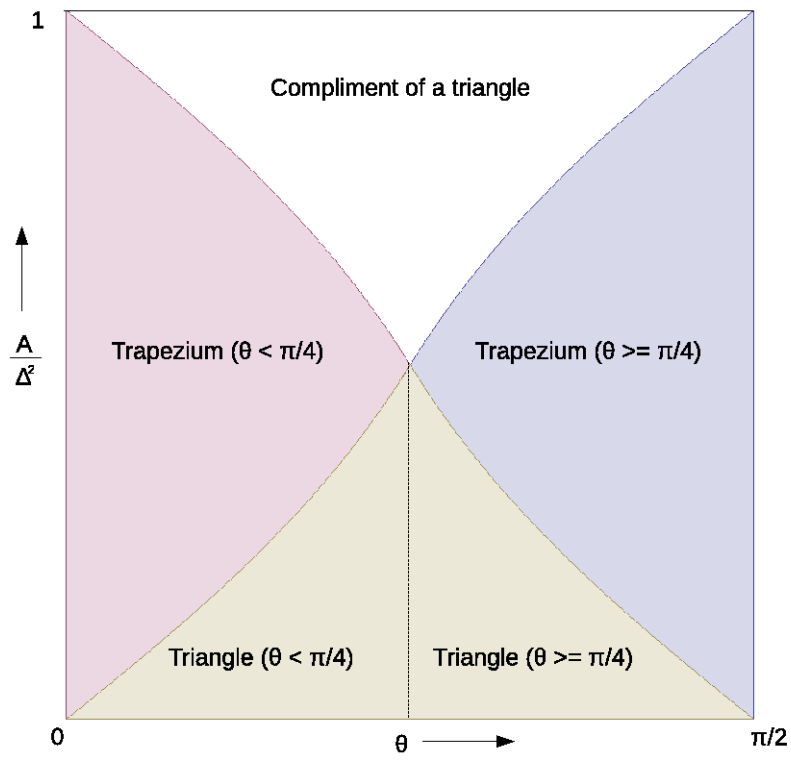


Figure 1.2: Area and angle region

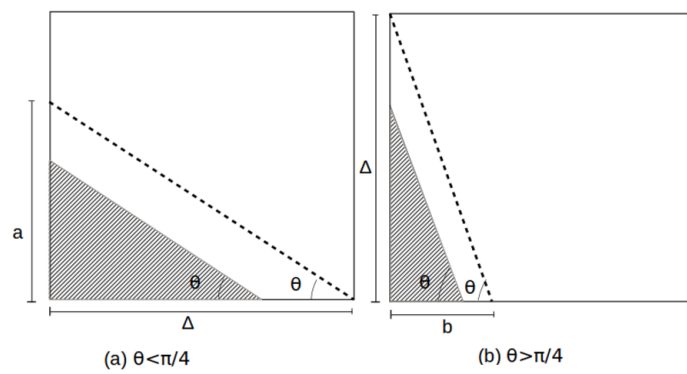


Figure 1.3: Area of the triangle made by dark fluid in the cell

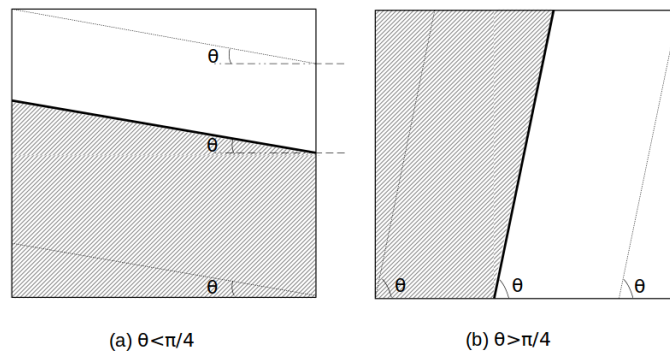


Figure 1.4: Area of a trapezium made by the dark fluid in the cell

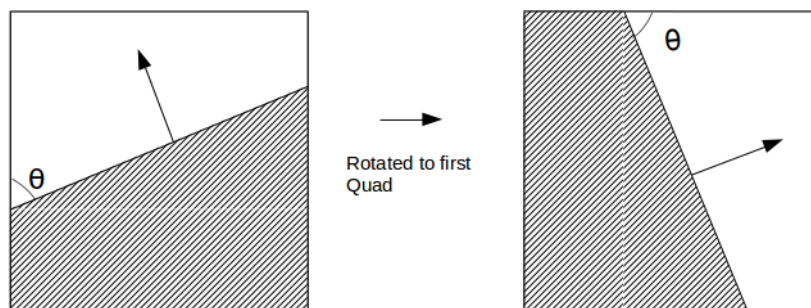


Figure 1.5: When the cell is reoriented to make normal lie in first quadrant

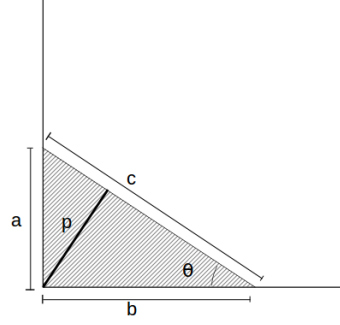


Figure 1.6: Perpendicular distance in a triangle

$$\begin{array}{ll} \frac{1}{2}\tan\theta < \frac{A}{\Delta^2} < \left(1 - \frac{1}{2}\tan\theta\right) & \text{for, } \theta < \frac{\pi}{4} \\ \frac{1}{2\tan\theta} < \frac{A}{\Delta^2} < \left(1 - \frac{1}{2\tan\theta}\right) & \text{for, } \theta \geq \frac{\pi}{4} \end{array}$$

For all other cases the shape becomes a complement of a triangle.

Step VI: Perpendicular distance Calculation

After reorientation, the perpendicular distance will now be always from the LHS corner of the cell.

Triangle

Note the following formulae for Figure 1.6

$$c = \frac{b}{\cos\theta}, \quad b = \frac{p}{\sin\theta}, \quad c = \frac{p}{\sin\theta\cos\theta}, \quad \frac{1}{2}pc = F\Delta^2, \quad \frac{p^2}{\sin 2\theta} = F\Delta^2$$

$$p = \sqrt{F\Delta^2 \sin 2\theta}$$

Trapezium,

Note the formulae for Figure 1.7

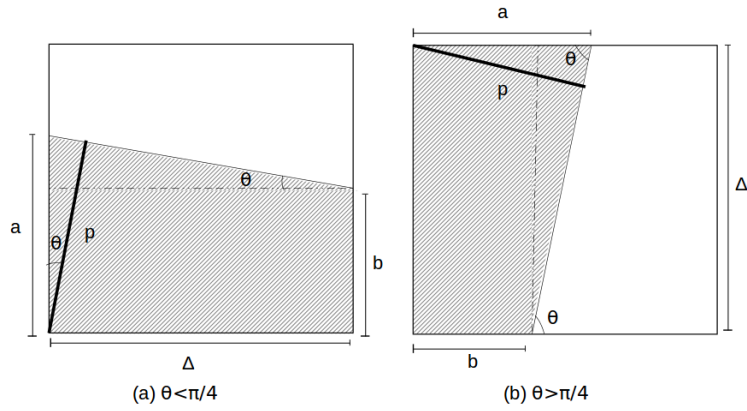


Figure 1.7: Perpendicular distance for a trapezium

$$\text{For } \theta < \frac{\pi}{4}, \quad \Delta^2 F = \frac{(a+b)\Delta}{2}, \quad a = \frac{p}{\cos\theta}, \quad \tan\theta = \frac{a-b}{\Delta}, \quad b = \frac{p}{\cos\theta} - \Delta \tan\theta,$$

$$\text{or, } \frac{1}{2}\Delta \left(\frac{2p}{\cos\theta} - \Delta \tan\theta \right) = \Delta^2 F, \quad \boxed{p = \Delta F \cos\theta + \frac{1}{2}\Delta \sin\theta}$$

$$\text{For } \theta \geq \frac{\pi}{4}, \quad \Delta^2 F = \frac{(a+b)\Delta}{2}, \quad a = \frac{p}{\sin\theta}, \quad \tan\theta = \frac{\Delta}{a-b}, \quad b = \frac{p}{\sin\theta} - \frac{\Delta}{\tan\theta},$$

$$\text{or, } 0.5\Delta \left(\frac{2p}{\sin\theta} - \frac{\Delta}{\tan\theta} \right) = \Delta^2 F, \quad \boxed{p = \Delta F \sin\theta + \frac{1}{2}\Delta \cos\theta}$$

For compliment of triangle,

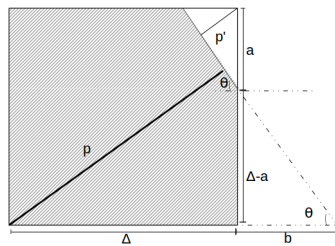


Figure 1.8: Perpendicular distance for compliment of the triangle

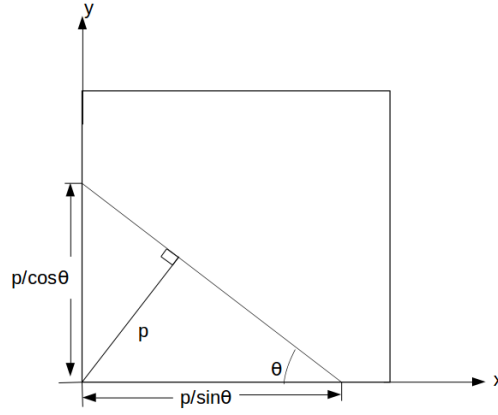


Figure 1.9: Normal form of a line in the cell

$$p' = \sqrt{F\Delta^2 \sin 2\theta}, \quad a = \frac{p'}{\cos \theta},$$

$$\text{or,} \quad b = \frac{\Delta - a}{\tan \theta}, \quad \boxed{p = \Delta(\sin \theta + \cos \theta) - p'}$$

Step VII : Line Extrapolation

After, reorientation of the line to the first quadrant. A line can be constructed with the θ and perpendicular distance using normal form, (See Figure 1.9)

$$x \sin \theta + y \cos \theta = p \quad (1.6)$$

Refer to Figure 1.10 for extrapolation,

After extrapolation the new volume fraction is calculated under each cell below the line, which is then used to calculate the norm.

Step VIII: Norm Calculation

Let the new volume fraction calculated be $\tilde{F}_{r,c}$. Now the norm which is here the difference of original volume fraction and new volume fraction calculated after the extrapolation are squared and then summed over 3 X 3 stencil. Also, called as L^2 norm,

$$\boxed{L^2(\theta) = \sum_{k,l=-1}^1 (\tilde{F}_{r+k,c+l} - F_{r+k,c+l})^2}$$

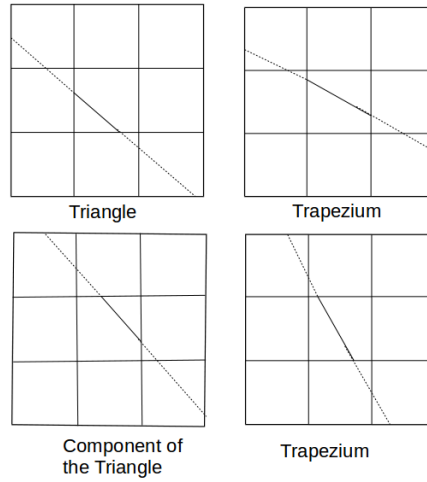


Figure 1.10: Extrapolation of the line in 3 X 3 cell

Step IX : Norm Minimization

After calculating norm the initial guess of slope and θ is changed with small step size but it should not be confused with rotation of interface or normal. All the steps are repeated to after modifying theta and norm is again calculated. This process repeats when a minima of norm reached. And the final values of θ , shape, and quadrant are assigned to the cell.

Advection

The Equation 1.2 is also solved using geometrical technique by calculating fluxes across the cells and then updating the F-field in each time step. Direct finite differencing of this equation will lose the discontinuous property of F-field and cause smearing of interface as shown in Figure ??

Flux Calculation

The flux of fluid through the walls is calculated using a graphical technique. A typical cell is shown in Figure 1.11 and the through its right wall is calculated . Various possibilities arise and these are discussed below.

Triangle

For a triangle there are two cases:-

if $(\Delta - udt) > x_0$, Nothing will leave from the right wall

if $(\Delta - udt) < x_0$, A triangle leaves.

here x_0 is the value of x at $y = 0$. The area of this triangle is given by,

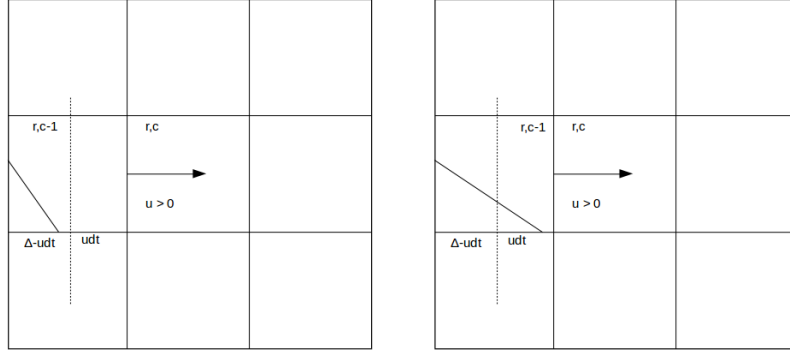
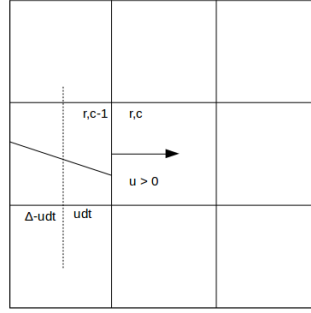


Figure 1.11: Cases for a triangle in flux calculation

Figure 1.12: Flux calculation for trapezium for $\theta < \frac{\pi}{4}$

$$Flux = \frac{1}{2}(x_0 - \Delta + udt)^2 \tan \theta$$

Trapezium

Two types of trapezium can be in the first quadrant, $\theta < \frac{\pi}{4}$, and $\theta > \frac{\pi}{4}$,
(See Figure 1.12 and 1.13)

For $\theta < \frac{\pi}{4}$, the only possibility is that a trapezium leaves and its area is given by,

$$Flux = \frac{1}{2}(y_0 - ((\Delta - udt) \tan \theta) + y_{del})udt$$

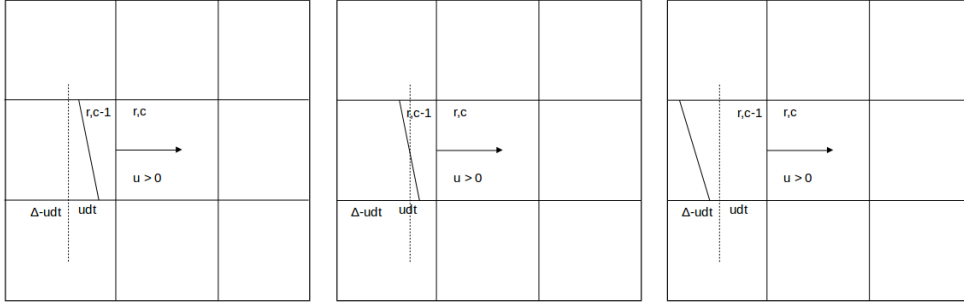


Figure 1.13: (a) $(\Delta - udt) < x_{del}$, (b) $x_{del} \leq (\Delta - udt) < x_0$, (c) $(\Delta - udt) \geq x_0$

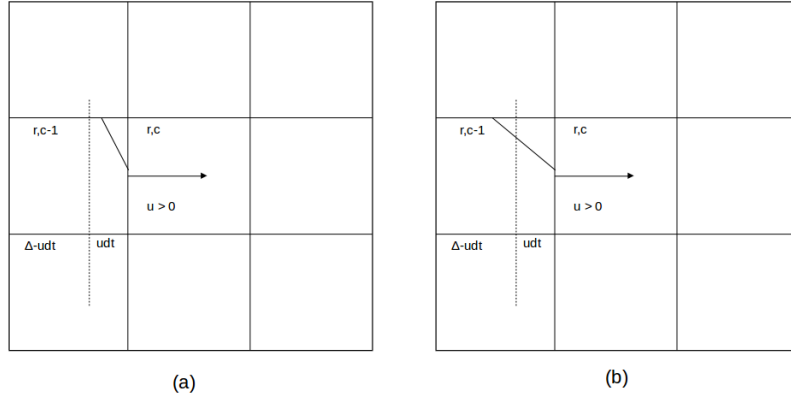


Figure 1.14: (a) $(\Delta - udt) < x_{del}$, (b) $(\Delta - udt) \geq x_{del}$

For $\theta > \frac{\pi}{4}$,

For $(\Delta - udt) < x_{del}$, a trapezium leaves the cell and the Flux is given by,

$$Flux = \frac{\Delta}{2}(x_{del} - (2(\Delta - udt)) + x_0)$$

For $x_{del} \leq (\Delta - udt) < x_0$, a triangle leaves the cell and the flux is given by,

$$Flux = \frac{1}{2 \tan \theta} (y_0 - (\Delta - udt) \tan \theta)^2$$

For $((\Delta - udt)) \geq x_0$, nothing leaves from the cell.

$$Flux = 0$$

Compliment of a triangle

For compliment of a triangle two cases arise, (See Figure 1.14)

if, $\Delta - udt < x_{del}$, A 5-Sided figure leaves, and Flux is given by,

$$Flux = \Delta(x_{del} - \Delta + udt) + \frac{1}{2}((\Delta + y_{del})(\Delta - x_{del})) \quad (\text{sum of rectangle and trapezium})$$

if, $\Delta - udt \geq x_{del}$, A trapezium leaves, and Flux is given by,

$$Flux = \frac{udt}{2}(y_{del} + y_0 - (\Delta - udt) \tan \theta)$$

Volume Fraction Calculation

The fluxes calculated in the previous section are used to update the volume fractions in the cells for the next time step. For this we use *Direction Split Young's method* (DSY). Youngs (1982). In this method the order of directions is interchanged after each time step in order to avoid systematic errors. To achieve mass conservation, for this the basic condition is velocity divergence. In this method, the mass cannot be conserved until all directions are taken into consideration. Hence the values of volume fraction larger than unity and less than zero may arise after first sweep which violates the restriction, $0 \leq F \leq 1$ for the next sweep. This difficulty can be solved by introducing effective volume of the cells (Rudman (1997)). But this induces risks of small under- or overshoots. Undershoot occurs when the all the volume fraction has to be fluxed out from the cell but it does not and cell cannot be emptied, overshoot occurs when the cell is fluxed with the volume fraction to get filled but it does not. Both occurs because of use of effective volume in calculation of fluxes through wall. This can be resolved by using flux correction in the second sweep Lörstad and Fuchs (2004). The following algorithm is used to calculate the volume fractions for the 2D flow.

Effective volume(non-dimensional, characteristic area Δ^2) for the I^{st} sweep,

$$\delta V_{r,c}^I = 1 - \Delta t \frac{V_{out} - V_{in}}{\Delta}$$

where, $\delta V_{r,c}^I$ is the effective volume after I^{st} sweep.

Net Flux out ΔJ_{out} , (non-dimensional) in or out in a cell is given by,

$$\Delta J_{out} = \frac{(J_{out} - J_{in})}{\Delta^2}$$

From which Volume Fraction after I^{st} of the cell is given by,

$$F_{r,c}^I = \frac{F^0 - \Delta J_{out}}{\delta V_{r,c}^1}$$

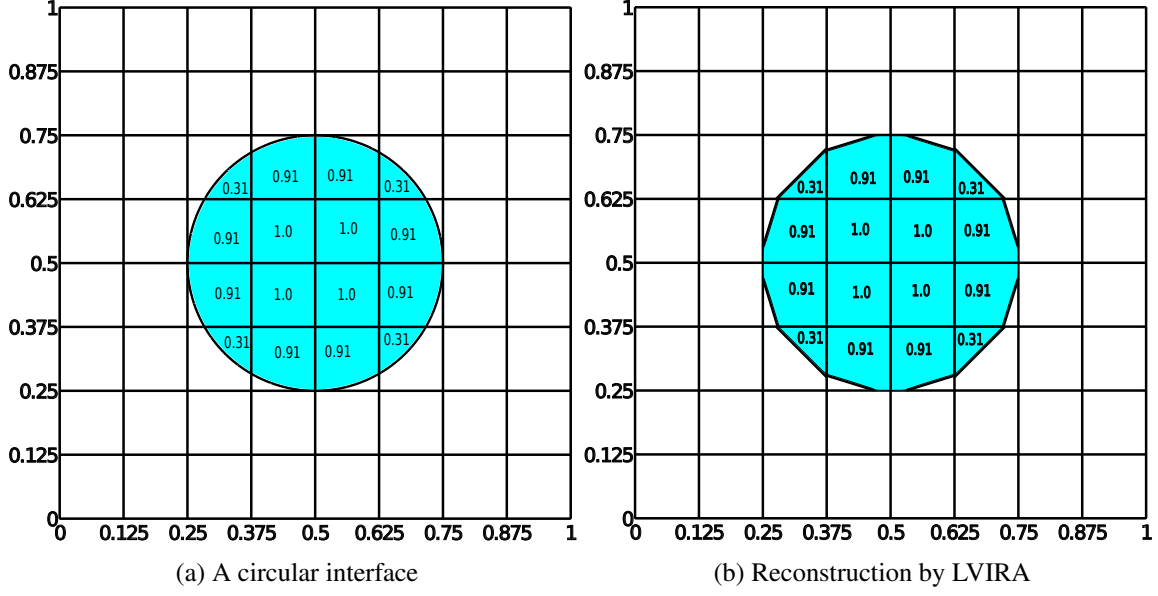


Figure 1.15: Reconstruction of a circular interface by LVIRA

where, F^0 and F^I are the volume fractions before I^{st} sweep and after I^{st} sweep respectively.

For II^{nd} sweep,

$$U_{out}^* = \frac{\Delta t}{\Delta} u_{out}$$

$$\zeta = \begin{cases} \zeta_1 = \frac{J_{out}}{U_{out}^*} & \text{if } |\zeta_1 - \frac{1}{2}| \geq |\zeta_2 - \frac{1}{2}| \\ \zeta_2 = \frac{F_{r,c} - J_{out}}{1 - U_{out}^*} & \text{if } |\zeta_1 - \frac{1}{2}| < |\zeta_2 - \frac{1}{2}| \end{cases}$$

Corrected volume is given by,

$$\delta V^{corr} = F_{r,c} + (1 - \delta V_{r,c}^I) \zeta$$

Corrected outgoing flux is given by,

$$J_{out}^{corr} = \delta V_{r,c}^I (J_{out} - U_{out}^*) + \delta V^{corr} U_{out}^*$$

Corrected Total flux out from the cell is given by

$$\Delta J_{out}^{corr} = J_{out}^{corr} - J_{in}$$

Final flux after last sweep is given by,

$$F_{r,c}^{II} = F_{r,c}^I \delta V_{r,c}^I - \Delta J_{out}^{corr}$$

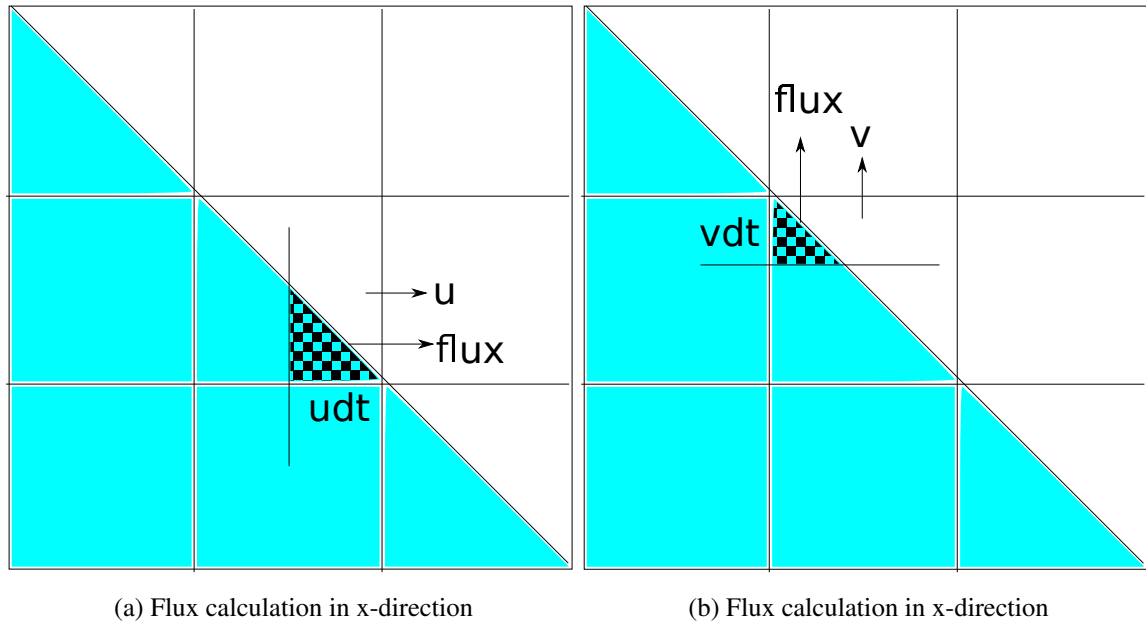


Figure 1.16: Advection of the interface by Youngs operator split algorithm

Verification

There are standard test cases available in the literature: Zalesak (1979), Puckett *et al.* (1997), Smolianski (2001), Gerlach *et al.* (2006) etc. These involve advecting the interface with a fixed underlying velocity field of varying degree of complexity. Volume of Fluid method is validated by choosing three test cases from Rudman (1997),

1. Circle in translational flow

In this test the velocity field has zero velocity gradient and it is the simplest of all the tests.

2. Solid body rotation of slotted circle

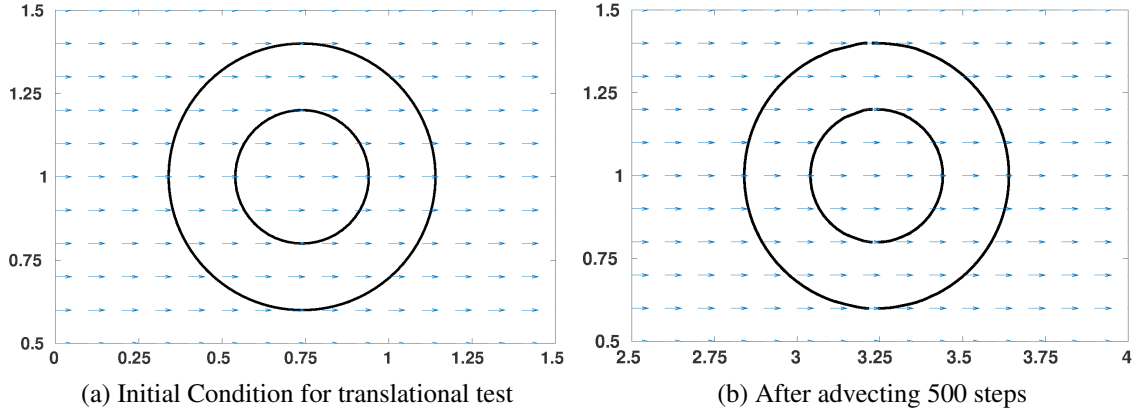
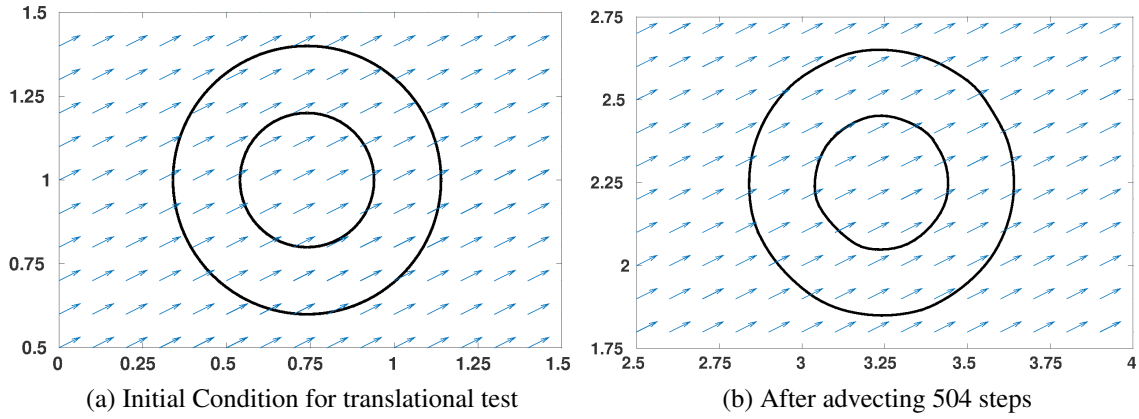
The test has rate of strain tensor zero and gradient of velocity only has the vorticity component with constant angular velocity at every point in space.

3. Circle in shear flow

Vorticity tensor in this test is zero and there is only the rate of strain tensor.

Advection of circle in translational flow

The algorithm is tested for the simplest case of unidirectional velocity field. Two concentric circles are used as the initial condition for translational test, with center at (0.75,1) and diameter of inner and outer circle is 0.4 and 0.8 respectively (Figure 1.17).

Figure 1.17: Advection test for velocity field $u=1, v=0$ Figure 1.18: Advection test for velocity field $u=2, v=1$

The volume fraction scalar field is advected by two velocity fields $u=1, v=0$ and $u=2, v=1$. The refinement of the domain which is $[0, 4] \times [0, 4]$ is 200×200 . The time step is 0.005 units and advection proceeds for 500 and 504 steps for case 1 and case 2 respectively.

Advection test for solid body rotation

For solid body rotation test a slotted circle configuration is taken from Zalesak (1979), the center of slotted circle is at $(2.0, 2.75)$ and diameter is 1.0. The length and width of slot is 0.6 and 0.12 respectively. (Figure 1.20 and 1.19) The refinement of the domain which is $[0, 4] \times [0, 4]$ is 200×200 . A velocity field is given by $u = -\Omega(y - y_0)$, $v = \Omega(x - x_0)$, where axis of rotation passes through the (x_0, y_0) and normal to the x - y plane. Ω is the angular velocity. Here $\Omega = 0.5$ and (x_0, y_0) is $(2, 2)$. The time step is 0.005.

1. Domain: $[0,4] \times [0,4]$
2. Grid Size: 200×200
3. Radius of circle :0.5
4. Center : (2.0,2.75)
5. Velocity field: $u = -0.5(y - 2), v = 0.5(x - 2)$

Shear Test

The real problems typically encounters the interface deformation, which includes merging and deformation. Hence the algorithm has to be tested for shear velocity field (Figure 1.21). The shear test problem verified with Gerlach *et al.* (2006). (Figure 1.22)

1. Domain: $[0,\pi] \times [0,\pi]$
2. Grid Size: 100×100
3. Radius of circle : $\frac{\pi}{5}$
4. Center : $(\pi/2, \pi/4)$
5. Velocity field(Forward): $u = \sin x \cos y, v = -\cos x \sin y$
6. Velocity field(Backward): $u = -\sin x \cos y, v = \cos x \sin y$

Calculation of error

The above results can be quantified by defining the error as

$$E = \frac{\sum |F_{r,c}^n - F_{r,c}^e|}{\sum F_{r,c}^0} \quad (1.7)$$

where E is the ratio of summation of difference between the volume fraction of cells of calculated solution and exact solution to the total sum initial volume fraction over all the cells. where F^n is the solution of volume fraction field after n time steps of computation, F^e is the exact solution, and F^0 is the initial solution. The initial solution can be calculated by the initial volume fraction field, exact solution of field for translational velocity fields can be easily calculated by recreating the circle at the center which has moved with the velocity field. For solid body rotation the exact solution is equal to the initial solution after one full rotation. For the shear test after 1000 backward steps the final solution should also be equal to initial solution. The errors calculated for various tests are shown in Table 1.1 and compared with Rudman (1997).

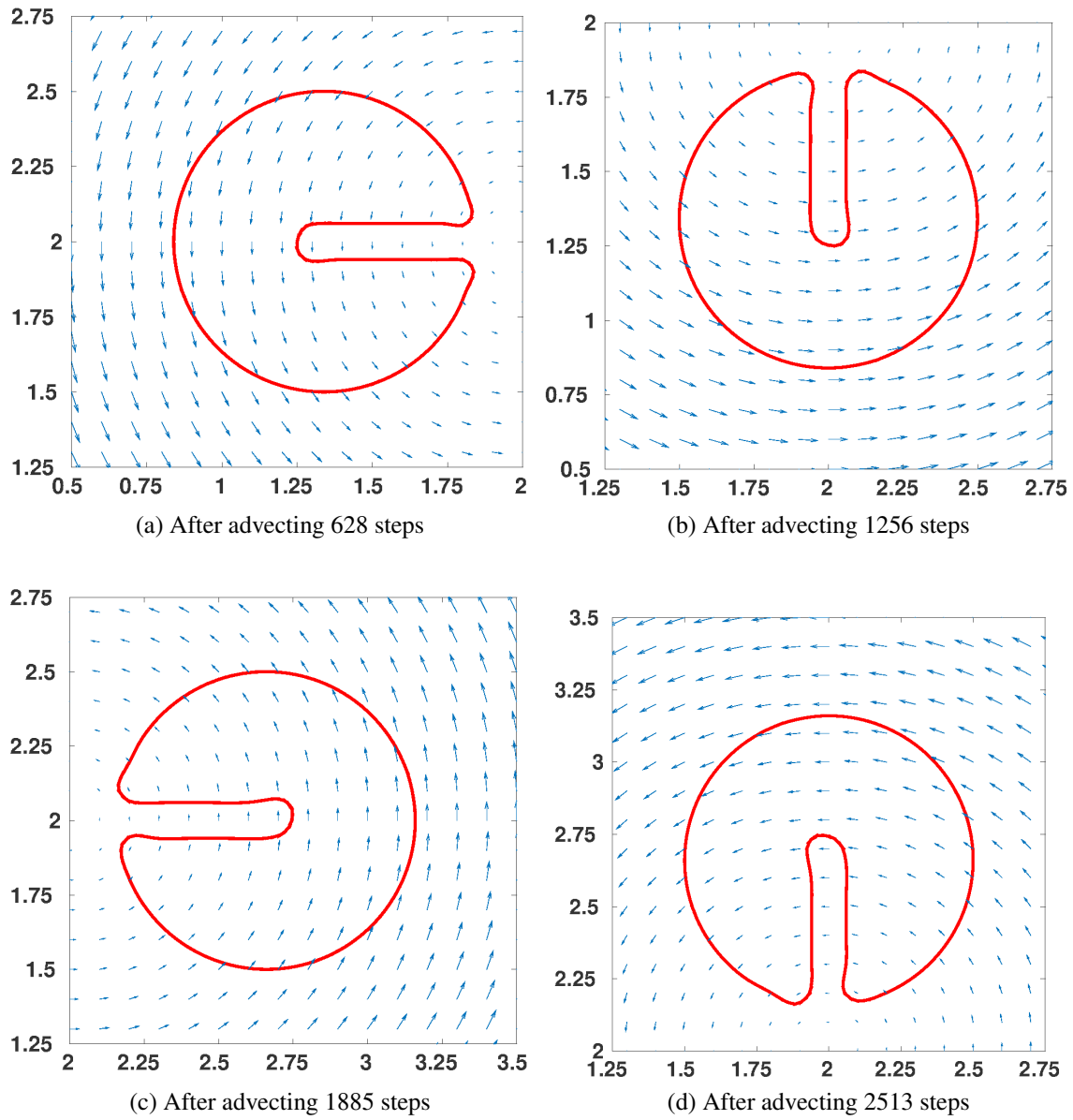


Figure 1.19: Advection test result for solid body rotation

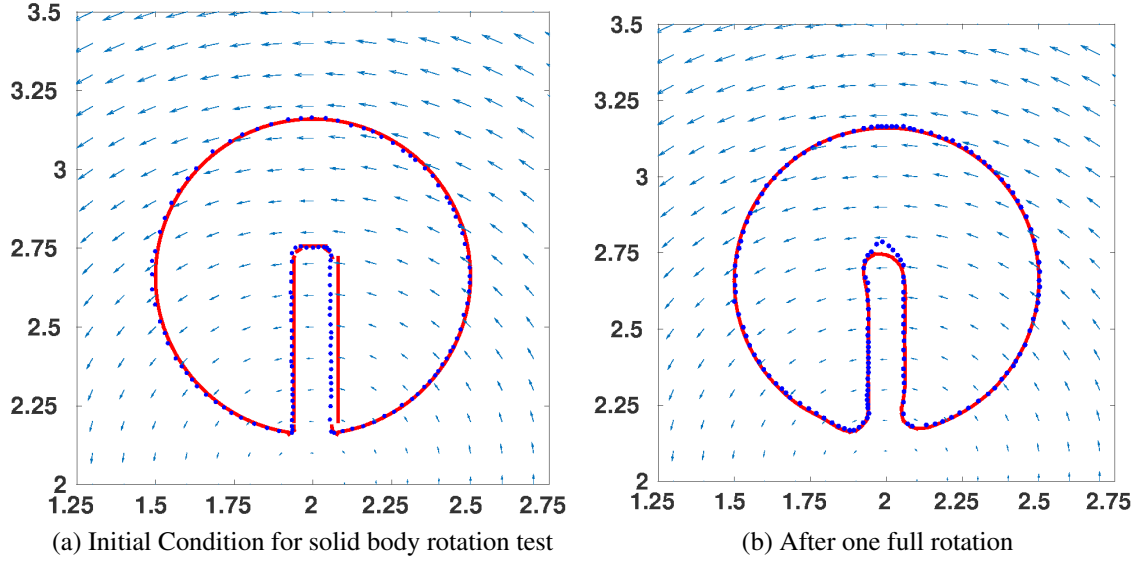


Figure 1.20: Comparison with Rudman (1997) results. (Red LVIRA and Blue Rudman (1997) data)

Table 1.1: Errors for various tests

Test	SLIC	Hirt-Nichols	FCT-VOF	Youngs	LVIRA (Present Study)
Translational (V(1,0))	1.30×10^{-2}	4.55×10^{-2}	1.28×10^{-2}	3.08×10^{-3}	1.5×10^{-3}
Translational (V(2,1))	9.18×10^{-2}	1.9×10^{-1}	3.99×10^{-2}	2.98×10^{-2}	1.05×10^{-2}
Shear Flow	4.59×10^{-2}	6.66×10^{-2}	3.14×10^{-2}	8.60×10^{-3}	6.90×10^{-3}
Solid Body Rotation (Slotted circle)	8.38×10^{-2}	9.62×10^{-2}	3.29×10^{-2}	1.09×10^{-2}	9.7×10^{-3}

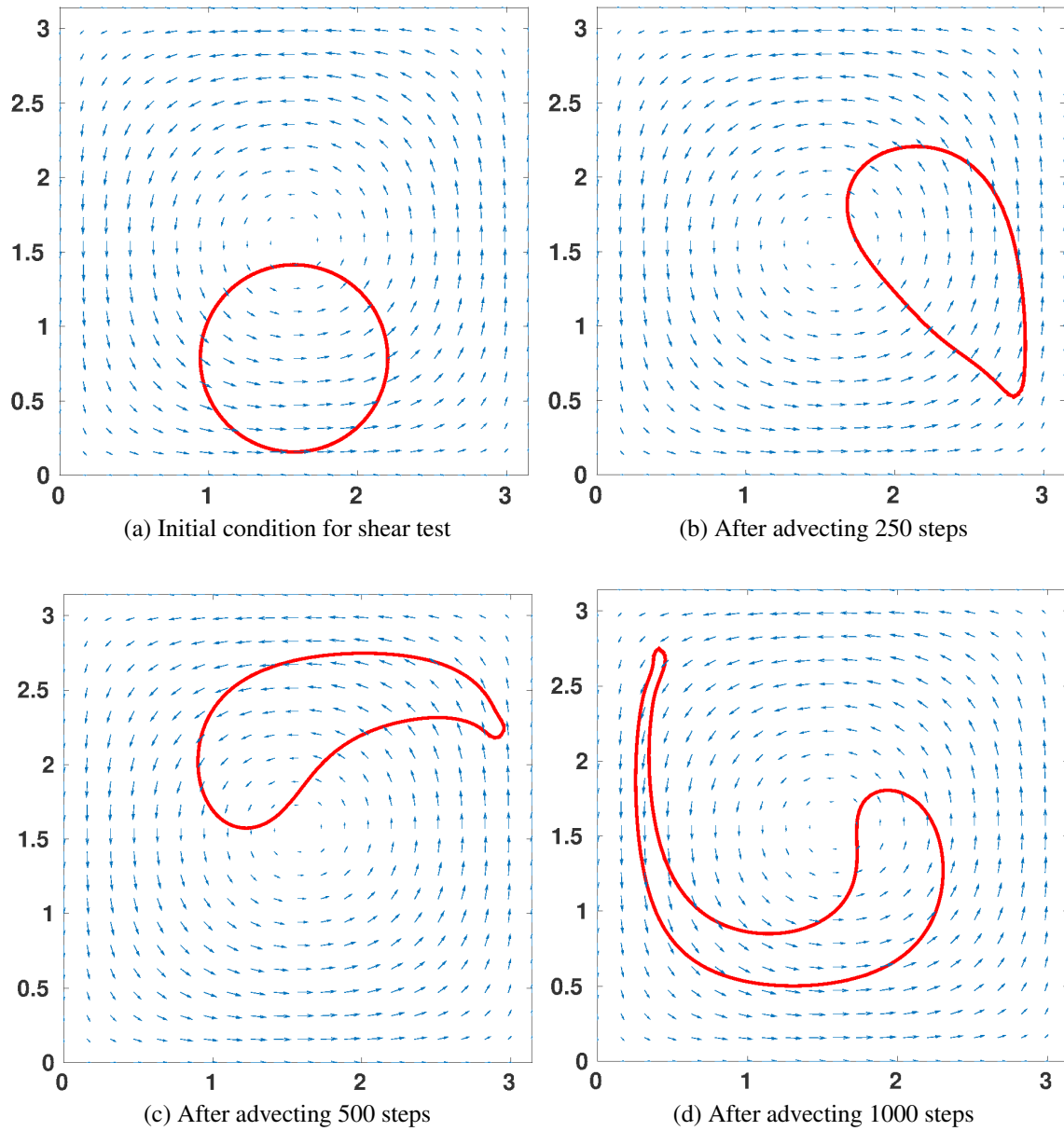


Figure 1.21: Advection test result for shear velocity field

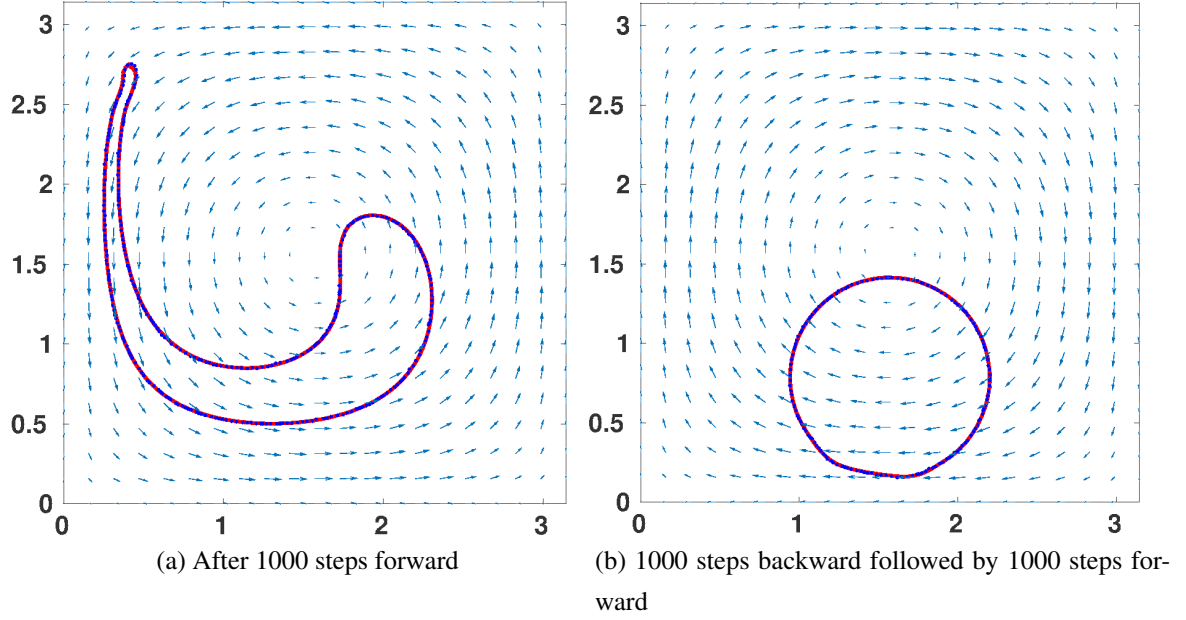


Figure 1.22: Comparison with Gerlach *et al.* (2006) results. (Red LVIRA and Blue Gerlach *et al.* (2006) data)

Conclusion

The reconstruction and advection of an interface is calculated mostly geometrical techniques and does not require extensive knowledge of computational techniques. We found LVIRA is more accurate than other methods available in the literature. We will couple this algorithm with a flow solver which we have discussed in the next chapter.

Chapter 2

Navier Stokes Solver for two phase flow

Governing Equations

As we adopt continuum hypothesis, we then apply conservation principles to obtain governing equations for the fluid flow. These are conservation of mass, momentum and energy which are briefly described here. This derivation is mostly adapted from Leal (2007).

Mass conservation

Consider a material volume element in Fig 2.1. A material volume is one whose shape at time $t = 0$, is arbitrary, and has a fixed set of material points. And therefore, moves with the local continuum velocity of the fluid at every point.

Conservation of mass states that mass is neither created nor destroyed. It implies that the mass inside the material volume is constant with respect to time. Mathematically,

$$\frac{D}{Dt} \left[\int_{V_m(t)} \rho dV \right] = 0 \quad (2.1)$$

where,

$$\frac{DB}{Dt} = \frac{\partial B}{\partial t} + u \cdot \nabla B \quad (2.2)$$

Using the Reynolds transport theorem, which says

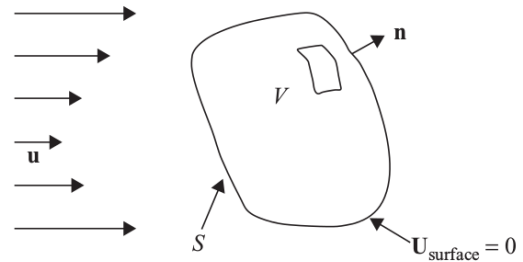


Figure 2.1: An arbitrary control volume Leal (2007)

$$\frac{D}{Dt} \left[\int_{V_m(t)} B(x, t) dV \right] = \int_{V_m(t)} \left[\frac{\partial B}{\partial t} + \nabla \cdot (Bu) \right] dV \quad (2.3)$$

2.1, can be written as,

$$\int_{V_m(t)} \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) \right] dV = 0 \quad (2.4)$$

where the choice of $V_m(t)$ is arbitrary, hence the integrand itself must be zero.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0 \quad (2.5)$$

For, incompressible fluids where density does not changes with time, 2.5 reduces to,

$$\nabla \cdot u = 0 \quad (2.6)$$

Momentum Conservation

Newton's second law states that,

Rate of change of linear momentum in an inertial frame = The sum of forces acting on the body (2.7)

When we apply this to a material control volume, we obtain

$$\frac{D}{Dt} \int_{V_m(t)} (\rho u) dV = \text{sum of the forces acting on } V_m(t) \quad (2.8)$$

From a continuum perspective, the forces which act on the control volume can be of two types. Body force and surface force. The body force which is common to most of the problems is gravitational force, which acts equally on all the volume elements. The surface forces are forces are short range forces which acts on the surface of the control volume.

RHS of 2.8, can be expressed as sum of these two forces,

$$\frac{D}{Dt} \int_{V_m(t)} (\rho u) dV = \int_{V_m(t)} \rho g dV + \int_{A_m(t)} J dA \quad (2.9)$$

where, g is acceleration due to gravity, $A_m(t)$ is the closed surface area of the control volume element and J is the **stress vector**. Using 2.3, we can write the RHS of 2.9 as,

$$\int_{V_m(t)} \left[\frac{\partial (\rho u)}{\partial t} + \nabla \cdot (\rho u u) \right] dV = \int_{V_m(t)} \rho g dV + \int_{A_m(t)} J dA \quad (2.10)$$

In the above equation stress vector J , can be found by linear vector operation on unit normal to the surface at any given point. The linear vector operator is \mathbf{T} , is called as stress tensor, which is a second order tensor. Thus,

$$J(n, x) = n \cdot \mathbf{T}(x) \quad (2.11)$$

Applying Gauss divergence theorem,

$$\int_{A_m(t)} n \cdot \mathbf{T} dS = \int_{V_m(t)} \nabla \cdot \mathbf{T} dV \quad (2.12)$$

and substituting in 2.10, we get

$$\int_{V_m(t)} \left[\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u u) \right] dV = \int_{V_m(t)} \rho g dV + \int_{V_m(t)} \nabla \cdot \mathbf{T} dV \quad (2.13)$$

After rearrangement,

$$\int_{V_m(t)} \left[\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u u) - \rho g - \nabla \cdot \mathbf{T} \right] dV = 0 \quad (2.14)$$

Again, the $V_m(t)$, is an arbitrary control volume, and the integrand must be then zero. Thus we obtain,

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u u) = \rho g + \nabla \cdot \mathbf{T} \quad (2.15)$$

The stress tensor \mathbf{T} can be expressed in terms of isotropic and non-isotropic parts,

$$\mathbf{T} = -p\mathbf{I} + \tau \quad (2.16)$$

Substitute this form in 2.15, we get

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u u) = \nabla \cdot (-p\mathbf{I} + \tau) + \rho g \quad (2.17)$$

The constitutive equation for a Newtonian fluid is,

$$\tau = 2\mu\mathbf{D} \quad (2.18)$$

where, D is the rate of strain tensor, the symmetric part of ∇u . Now,

$$\tau = \mu(\nabla u + (\nabla u)^T) \quad (2.19)$$

Substituting this in 2.17, we get

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u u) = -\nabla p + \nabla \cdot (\mu(\nabla u + (\nabla u)^T)) + \rho g \quad (2.20)$$

Subtracting 2.5 from 2.20, we get

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u = -\frac{\nabla p}{\rho} + \frac{1}{\rho} \nabla \cdot (\mu(\nabla u + (\nabla u)^T)) + g \quad (2.21)$$

Rewriting 2.21, in conservative form, using mass conservation

$$\frac{\partial u}{\partial t} + \nabla \cdot (uu) = -\frac{\nabla p}{\rho} + \frac{1}{\rho} \nabla \cdot (\mu(\nabla u + (\nabla u)^T)) + g \quad (2.22)$$

Non-Dimensionalisation of governing equations

For non-dimensionalisation we chose scale as characteristic length L , characteristic velocity U , characteristic time $\frac{L}{U}$, characteristic pressure $\rho_L U^2$, characteristic density ρ_L and characteristic viscosity μ_L . Substitute $u = U\tilde{u}$, $x = L\tilde{x}$, $y = L\tilde{y}$, $t = \frac{L}{U}\tilde{t}$, $p = \rho_L U^2 \tilde{p}$, $\rho = \rho_L \tilde{\rho}$, $\mu = \mu_L \tilde{\mu}$ in 2.22, where quantities with tilde are non-dimensional.

$$\frac{U^2}{L} \left[\frac{\partial \tilde{u}}{\partial \tilde{t}} + \tilde{\nabla} \cdot (\tilde{u}\tilde{u}) \right] = -\frac{\rho_L}{\rho_L \tilde{\rho}} \frac{U^2}{L} \tilde{\nabla} \tilde{p} + \frac{U^2}{L^2 \rho_L \tilde{\rho}} \tilde{\nabla} \cdot (\mu_L \tilde{\mu} (\tilde{\nabla} \tilde{u} + (\tilde{\nabla} \tilde{u})^T)) + g \quad (2.23)$$

Rearranging 2.23,

$$\left[\frac{\partial \tilde{u}}{\partial \tilde{t}} + \tilde{\nabla} \cdot (\tilde{u}\tilde{u}) \right] = -\frac{1}{\tilde{\rho}} \tilde{\nabla} \tilde{p} + \frac{\mu_L}{L \rho_L U \tilde{\rho}} \tilde{\nabla} \cdot (\tilde{\mu} (\tilde{\nabla} \tilde{u} + (\tilde{\nabla} \tilde{u})^T)) + \frac{gL}{U^2} \quad (2.24)$$

Now, dropping tilde from the quantities,

$$\frac{\partial u}{\partial t} + \nabla \cdot (uu) = -\frac{1}{\tilde{\rho}} \nabla p + \frac{1}{Re_L} \frac{1}{\tilde{\rho}} \nabla \cdot (\tilde{\mu} (\nabla u + (\nabla u)^T)) + \frac{1}{Fr^2} \quad (2.25)$$

where, $Re_L = \frac{L \rho_L U}{\mu_L}$, $We_L = \frac{L \rho_L U^2}{\sigma_{LG}}$ and $Fr = \frac{U}{\sqrt{gL}}$.

2.25 is the non-dimensional form for incompressible multiphase newtonian flow.

Integral form of governing equation

As the conservation equations are valid for a differential element, we can also integrate them over a control volume. This is the final step before the finite volume discretisation approach. 2.25 can be integrated on a control volume, V

$$\int_V \left[\frac{\partial u}{\partial t} + \nabla \cdot (uu) \right] dV = \int_V \left[-\frac{1}{\tilde{\rho}} \nabla p + \frac{1}{Re_L} \frac{1}{\tilde{\rho}} \nabla \cdot (\tilde{\mu} (\nabla u + (\nabla u)^T)) + \frac{1}{Fr^2} \right] dV \quad (2.26)$$

$$\int_V \frac{\partial u}{\partial t} dV + \int_V \nabla \cdot (uu) dV = - \int_V \frac{1}{\tilde{\rho}} \nabla p dV + \frac{1}{Re_L} \frac{1}{\tilde{\rho}} \int_V \nabla \cdot (\tilde{\mu} (\nabla u + (\nabla u)^T)) dV + \frac{1}{Fr^2} \int_V dV$$

Applying Gauss divergence theorem on advection and diffusion integrals, we get

$$\int_V \frac{\partial u}{\partial t} dV + \int_S u(u \cdot n) dS = - \int_V \frac{1}{\tilde{\rho}} \nabla p dV + \frac{1}{Re_L} \frac{1}{\tilde{\rho}} \int_S (\tilde{\mu} (\nabla u + (\nabla u)^T) \cdot n) dS + \frac{1}{Fr^2} \int_V dV$$

Now, we can average the individual quantities over the control volume, V and Equation 2.27 is used for discretisation in our code.

$$\begin{aligned} \frac{1}{V} \int_V \frac{\partial u}{\partial t} dV + \frac{1}{V} \int_S u(u \cdot n) dS = & -\frac{1}{V} \int_V \frac{1}{\tilde{\rho}} \nabla p dV + \frac{1}{Re_L} \frac{1}{\tilde{\rho}} \frac{1}{V} \int_S (\tilde{\mu}(\nabla u + (\nabla u)^T) \cdot n) dS \\ & + \frac{1}{Fr^2} \frac{1}{V} \int_V dV \end{aligned} \quad (2.27)$$

Discretisation of governing equations

Grid

Before discretisation, we first specify our domain, grid shape and approach. The grid is rectangular and uniform grid spacing has been used for both x and y directions. Figure 2.2 and 2.3 shows the grid and location of pressure and velocity nodes. We can see the nodes of pressure and velocities are not colocated because decoupling of pressure and velocity may occur since the discretised form of equation of continuity on colocated grids can lead to checkerboard patterns in the solution (Anderson (1995)). Staggered grid approach is used to avoid pressure-velocity decoupling. There are three different grids for x -velocity (u), y -velocity (v) and pressure (p). The momentum equations are discretised using finite volume approach using the conservative form of equations. This would first order in time and a predictor-corrector approach to compute velocity field in the domain.

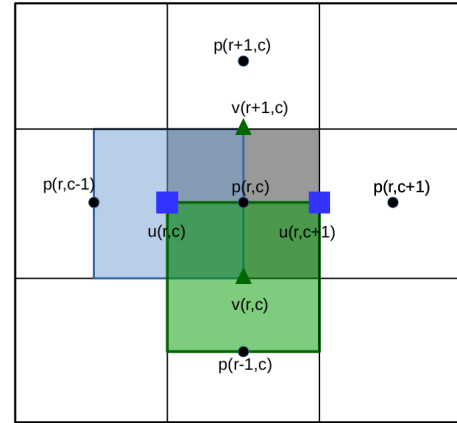


Figure 2.2: Staggered grid for x -velocity (Blue), y -velocity (Green) and pressure (Gray)

Integration in time

The first step is to compute projected velocity, ignoring the pressure terms,

$$\frac{u^* - u^n}{\Delta t} = -A^n + D^n + B^n \quad (2.28)$$

where u^n is the velocity at previous time step and u^* is the projected velocity without taking pressure into consideration, where $A = \frac{1}{V} \int_S u(u \cdot \hat{n}) dS$,

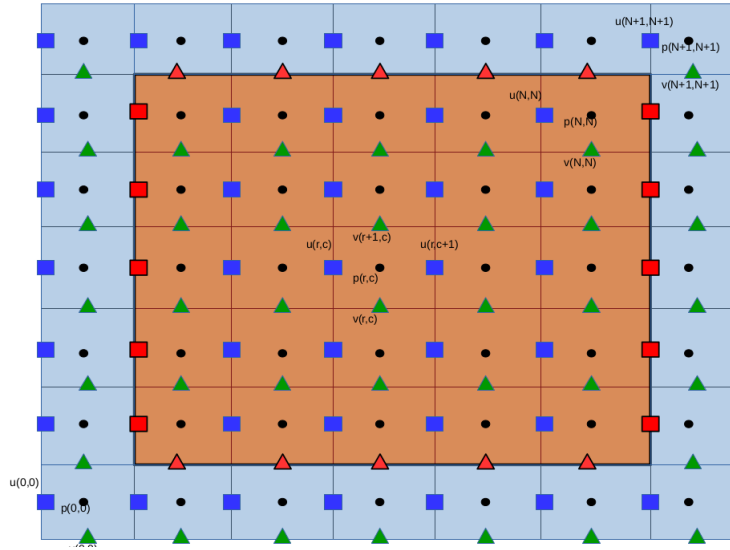


Figure 2.3: Location of nodes, blue square:x-velocity, red square:boundary x-velocity, green triangle:y-velocity, red triangle:boundary y-velocity,black dot:pressure

$$D = \frac{1}{V} \frac{1}{Re_L} \frac{1}{\bar{\rho}} \frac{1}{V} \int_S (\tilde{\mu}(\nabla u + (\nabla u)^T) \cdot \hat{n}) dS \text{ and}$$

$$B = \frac{1}{Fr^2} \frac{1}{V} \int_V dV \text{ And then adding the pressure component in the projected velocity,}$$

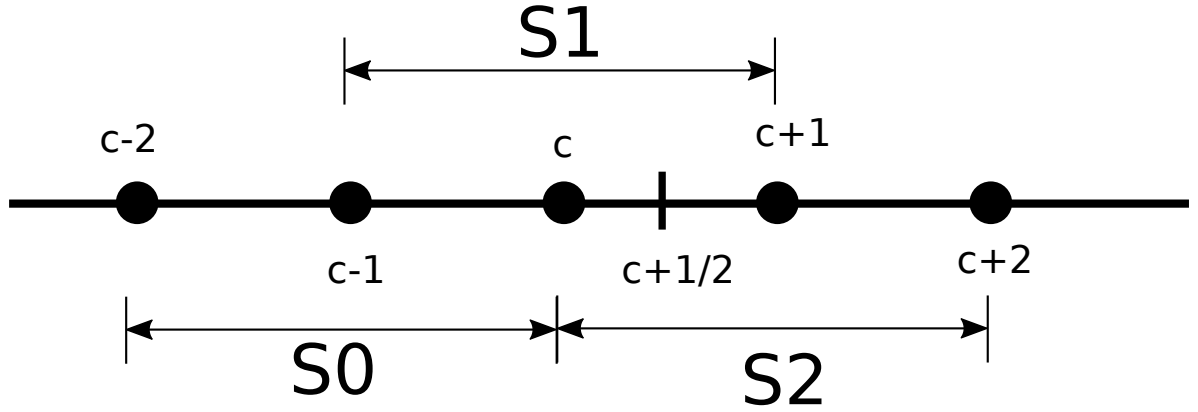
$$\frac{u^{n+1} - u^*}{\Delta t} = -\frac{1}{V} \int_V \frac{1}{\rho} \nabla p^{n+1} dV \quad (2.29)$$

Rearranging,

$$u^{n+1} = u^* - \frac{\Delta t}{V} \int_V \frac{1}{\rho} \nabla p^{n+1} dV \quad (2.30)$$

Discretisation of advection terms

To evaluate the advection terms, integrate it over the surface of the control volume. It is actually the sum of in and out fluxes through the faces of control volume. Here an approximation is made by assuming a uniform velocity on the faces of control volume and the fluxes are thus calculated by the value at the center of the boundary. Weighted Essentially Non-Oscillatory(WENO) [Liu *et al.* (1994), Jiang and Shu (1995)] scheme has been used to approximate the values of velocities at the face centers.

Figure 2.4: Three sub stencils for reconstruction of $u_{i+1/2}$

$$\begin{aligned}
 u_{c+1/2}^{(0)} &= \frac{1}{3}u_{c-2} - \frac{7}{6}u_{c-1} + \frac{11}{6}u_c \\
 u_{c+1/2}^{(1)} &= -\frac{1}{6}u_{c-1} + \frac{5}{6}u_c + \frac{1}{3}u_{c+1} \\
 u_{c+1/2}^{(2)} &= \frac{1}{3}u_c + \frac{5}{6}u_{c+1} - \frac{1}{6}u_{c+2}
 \end{aligned}$$

Once we get $u_{c+1/2}^k$ from the three sub-stencils, we can get,

$$u_{c+1/2} = \sum_{k=0}^2 \omega_k u_{c+1/2}^{(k)} \quad (2.31)$$

where ω_k is given by,

$$\begin{aligned}
 \omega_k &= \frac{\tilde{\omega}_k}{\sum_{k=0}^2 \tilde{\omega}_k}, \quad \tilde{\omega}_k = \frac{\gamma_k}{(\epsilon + \beta_k)^2} \\
 \gamma_0 &= \frac{1}{10}, \quad \gamma_1 = \frac{3}{5}, \quad \text{and} \quad \gamma_2 = \frac{3}{10} \\
 \epsilon &= 10^{-6}
 \end{aligned}$$

And, smoothness indicator β_k is given by,

$$\begin{aligned}
 \beta_0 &= \frac{13}{12}(u_{c-2} - 2u_{c-1} + u_c)^2 + \frac{1}{4}(u_{c-2} - 4u_{c-1} + 3u_c)^2 \\
 \beta_1 &= \frac{13}{12}(u_{c-1} - 2u_c + u_{c+1})^2 + \frac{1}{4}(u_{c-1} - u_{c+1})^2 \\
 \beta_2 &= \frac{13}{12}(u_c - 2u_{c+1} + u_{c+2})^2 + \frac{1}{4}(3u_c - 4u_{c+1} + u_{c+2})^2
 \end{aligned}$$

$$(A_x)_{net} = \frac{1}{V} \int_S u(u \cdot n) dS \quad (2.32)$$

$$= \frac{1}{\Delta x \Delta y} \left[\int_{right} u(u \cdot n) ds - \int_{left} u(u \cdot n) ds + \int_{top} u(u \cdot n) ds - \int_{bottom} u(u \cdot n) ds \right] \quad (2.33)$$

Diffusion terms

Before looking at the discretised form of diffusion terms, we shall derive the components for x and y directions.

Components of Diffusion term

Gradient of velocity for 2D flow,

$$\nabla u = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix}$$

$$\nabla u + (\nabla u)^T = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix} + \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} \end{bmatrix}$$

$$\nabla u + (\nabla u)^T = \begin{bmatrix} 2\frac{\partial u}{\partial x} & \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & 2\frac{\partial v}{\partial y} \end{bmatrix}$$

$$(\tilde{\mu}(\nabla u + (\nabla u)^T)) = \begin{bmatrix} 2\tilde{\mu}\frac{\partial u}{\partial x} & \tilde{\mu}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ \tilde{\mu}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & 2\tilde{\mu}\frac{\partial v}{\partial y} \end{bmatrix}$$

$$\begin{aligned} (\tilde{\mu}(\nabla u + (\nabla u)^T))_x &= \begin{bmatrix} 2\tilde{\mu}\frac{\partial u}{\partial x} & \tilde{\mu}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ \tilde{\mu}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & 2\tilde{\mu}\frac{\partial v}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= 2\tilde{\mu}\frac{\partial u}{\partial x} + \tilde{\mu}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \end{aligned}$$

$$\begin{aligned} (\tilde{\mu}(\nabla u + (\nabla u)^T))_y &= \begin{bmatrix} 2\tilde{\mu}\frac{\partial u}{\partial x} & \tilde{\mu}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ \tilde{\mu}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & 2\tilde{\mu}\frac{\partial v}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= \tilde{\mu}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + 2\tilde{\mu}\frac{\partial v}{\partial y} \end{aligned}$$

Diffusion terms are integrated on the surface of control volume,

$$\begin{aligned}
 (D_x)_{net} &= \frac{1}{V} \int_S \left[2\tilde{\mu} \frac{\partial u}{\partial x} + \tilde{\mu} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \cdot \hat{n} \right] dS \\
 &= \frac{1}{\Delta x \Delta y} \left[\int_{right} 2\tilde{\mu} \frac{\partial u}{\partial x} + \tilde{\mu} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) ds \\
 &\quad - \int_{left} 2\tilde{\mu} \frac{\partial u}{\partial x} + \tilde{\mu} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) ds \\
 &\quad + \int_{top} 2\tilde{\mu} \frac{\partial u}{\partial x} + \tilde{\mu} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) ds \\
 &\quad - \int_{bottom} 2\tilde{\mu} \frac{\partial u}{\partial x} + \tilde{\mu} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) ds \right] \quad (2.34)
 \end{aligned}$$

Discretisation

Integrands in above equations are constant over the faces of control volume, and derivatives are approximated through central differencing,

$$\begin{aligned}
 (D_x)_{right} &= \frac{2}{\Delta x} \left[\mu_{r,c} \left(\frac{u_{r,c+1} - u_{r,c}}{\Delta x} \right) \right] \\
 (D_x)_{left} &= \frac{2}{\Delta x} \left[\mu_{r,c-1} \left(\frac{u_{r,c} - u_{r,c-1}}{\Delta x} \right) \right] \\
 (D_x)_{top} &= \frac{\mu_{top}}{\Delta y} \left[\frac{u_{r+1,c} - u_{r,c}}{\Delta y} + \frac{v_{r+1,c} - v_{r+1,c-1}}{\Delta x} \right] \\
 (D_x)_{bottom} &= \frac{\mu_{bottom}}{\Delta y} \left[\frac{u_{r,c} - u_{r-1,c}}{\Delta y} + \frac{v_{r,c} - v_{r,c-1}}{\Delta x} \right]
 \end{aligned}$$

mean viscosity is used where it is not defined, and therefore,

$$\begin{aligned}
 \mu_{top} &= \frac{1}{4} [\mu_{r+1,c} + \mu_{r+1,c-1} + \mu_{r,c-1} + \mu_{r,c}] \\
 \mu_{bottom} &= \frac{1}{4} [\mu_{r,c} + \mu_{r,c-1} + \mu_{r-1,c-1} + \mu_{r-1,c}]
 \end{aligned}$$

Boundary conditions for velocity

Boundary conditions are directly imposed by setting the values of velocity on the boundary and ghost control volumes. At the boundary where the variable is not defined we use the ghost cells.

Dirichlet Boundary Conditions

The most commonly used boundary condition is the no-slip on the boundaries, we can set the velocity variables on the boundary to zero or any other value. On the left boundary of the domain, the x-velocities are defined so we can set them zero directly, but to set y-velocities we set the ghost control volumes y-velocities to change so that the average of ghost and the boundary control volume value becomes zero which is at the wall. This has to be implemented inside the time loop, and ghost value has to be updated at every time step

$$\begin{aligned} v_{wall} &= \frac{v_{r,0} + v_{r,1}}{2} \\ v_{r,0} &= 2v_{wall} - v_{r,1} \end{aligned}$$

The similar operation is used for x-velocities at bottom and top boundaries.

Neumann Boundary Conditions

Conditions are where normal derivative for tangential component of velocity on the wall is zero, and normal component of velocity on the wall is zero. This is equivalent to free slip, impermeable boundary, also called as symmetry condition. Mathematically, for left wall

$$\begin{aligned} \frac{\partial v}{\partial x} &= 0 & \text{Neumann} \\ u_{wall} &= 0 & \text{Dirichlet} \end{aligned}$$

To implement the above Neumann condition, on say left wall, we set the following in our code,

$$v_{r,0} = v_{r,1}$$

Pressure Poisson Equation(PPE)

For incompressible fluids, it can be followed from 2.6 that,

$$\nabla \cdot u^{n+1} = 0 \quad (2.35)$$

$$\nabla \cdot u^{n+1} = \nabla \cdot \left(u^* - \frac{\Delta t}{V} \int_V \frac{1}{\rho} \nabla p^{n+1} dV \right) = 0$$

$$\nabla \cdot \left(\frac{\Delta t}{V} \int_V \frac{1}{\rho} \nabla p^{n+1} dV \right) = \nabla \cdot u^*$$

$$\frac{u_{r,c+1}^{n+1} - u_{r,c}^{n+1}}{\Delta x} + \frac{v_{r+1,c}^{n+1} - v_{r,c}^{n+1}}{\Delta y} = 0 \quad (2.36)$$

It essentially states that, any solution of velocity field should comply mass conservation, and the velocity field thus obtained must be divergence free. substituting (33) in (46), we get

$$\begin{aligned} & \frac{1}{\Delta x} \left[u_{r,c+1}^* - \frac{2\Delta t}{(\rho_{r,c} + \rho_{r,c+1})} \left(\frac{p_{r,c+1} - p_{r,c}}{\Delta x} \right) \right. \\ & \quad \left. - u_{r,c}^* + \frac{2\Delta t}{(\rho_{r,c} + \rho_{r,c-1})} \left(\frac{p_{r,c} - p_{r,c-1}}{\Delta x} \right) \right] \\ & + \frac{1}{\Delta y} \left[v_{r+1,c}^* - \frac{2\Delta t}{(\rho_{r+1,c} + \rho_{r,c})} \left(\frac{p_{r+1,c} - p_{r,c}}{\Delta y} \right) \right. \\ & \quad \left. - v_{r,c}^* + \frac{2\Delta t}{(\rho_{r,c} + \rho_{r-1,c})} \left(\frac{p_{r,c} - p_{r-1,c}}{\Delta y} \right) \right] = 0 \end{aligned}$$

From which we obtain,

$$\begin{aligned} p_{r,c} = & \left[\frac{1}{(\Delta x)^2} \left(\frac{1}{\rho_{r,c+1} + \rho_{r,c}} + \frac{1}{\rho_{r,c} + \rho_{r,c-1}} \right) + \frac{1}{(\Delta y)^2} \left(\frac{1}{\rho_{r+1,c} + \rho_{r,c}} + \frac{1}{\rho_{r,c} + \rho_{r-1,c}} \right) \right]^{-1} \\ & \left\{ \frac{1}{(\Delta x)^2} \left(\frac{p_{r,c+1}}{\rho_{r,c+1} + \rho_{r,c}} + \frac{p_{r,c-1}}{\rho_{r,c} + \rho_{r,c-1}} \right) + \frac{1}{(\Delta y)^2} \left(\frac{p_{r+1,c}}{\rho_{r+1,c} + \rho_{r,c}} + \frac{p_{r-1,c}}{\rho_{r,c} + \rho_{r-1,c}} \right) \right. \\ & \quad \left. - \frac{1}{2\Delta t} \left(\frac{u_{r,c+1}^* - u_{r,c}^*}{\Delta x} + \frac{v_{r+1,c}^* - v_{r,c}^*}{\Delta y} \right) \right\} \end{aligned}$$

Successive Over Relaxation method (SOR)

Successive over relaxation method is used to solve a linear system of equations derived by extrapolating the Gauss-Sidel method. This extrapolation takes the form of a weighted average between the previous iterate and the computed Gauss-Seidel iterate successively for each component.

$$y_i^k = \omega \bar{y}_i^k + (1 - \omega) y_i^{k-1}$$

where \bar{y}_i^k , denotes the Gauss-Sidel iterate and ω is the relaxation parameter which is used to accelerate the convergence.

$$\begin{aligned} p_{r,c}^{\alpha+1} = & \omega \left[\frac{1}{(\Delta x)^2} \left(\frac{1}{\rho_{r,c+1} + \rho_{r,c}} + \frac{1}{\rho_{r,c} + \rho_{r,c-1}} \right) + \frac{1}{(\Delta y)^2} \left(\frac{1}{\rho_{r+1,c} + \rho_{r,c}} + \frac{1}{\rho_{r,c} + \rho_{r-1,c}} \right) \right]^{-1} \\ & \left\{ \frac{1}{(\Delta x)^2} \left(\frac{p_{r,c+1}}{\rho_{r,c+1} + \rho_{r,c}} + \frac{p_{r,c-1}}{\rho_{r,c} + \rho_{r,c-1}} \right) + \frac{1}{(\Delta y)^2} \left(\frac{p_{r+1,c}}{\rho_{r+1,c} + \rho_{r,c}} + \frac{p_{r-1,c}}{\rho_{r,c} + \rho_{r-1,c}} \right) \right. \\ & \quad \left. - \frac{1}{2\Delta t} \left(\frac{u_{r,c+1}^* - u_{r,c}^*}{\Delta x} + \frac{u_{r,c+1}^* - u_{r,c}^*}{\Delta x} \right) \right\} \\ & + (1 - \omega) p_{r,c}^\alpha \end{aligned}$$

where α is the previous iteration step and ω is the relaxation parameter, which can take values from 0-2, for overrelaxation it must be greater than 1. For stability reasons it must be below 2. A choice of $\omega = 1.2 - 1.5$ is usually a good compromise between stability and convergence. The advantage to use SOR is its simplicity but it converges very slowly. For faster runs we have to more use advanced methods. (Tryggvason *et al.* (2011))

Boundary Conditions for PPE

There is no explicit boundary condition for pressure needed, rather we use velocity boundary conditions to get pressure at boundary control volumes. (Tryggvason *et al.* (2011)) thus 2.35 at boundary reduces to

$$\frac{u_{r,c+1}^{n+1} - u_{b,r}}{\Delta x} + \frac{v_{r+1,c}^{n+1} - v_{r,c}^{n+1}}{\Delta y} = 0 \quad (2.37)$$

where $u_{b,r}$, the boundary value of velocity is known by velocity boundary conditions. From 2.37 the $p_{r,c}$ is computed on the boundary and corner control volumes.

Coupling VOF and Navier-Stokes solver

The volume of fluid interface capturing is coupled with the Navier-stokes solver. Figure 2.5 shows the schematic to couple the interface capturing and Navier-stokes solver for a time step. The steps followed are discussed in detail below:-

Input

The solver is fed with following to start the time marching:-

1. Initial Volume Fraction field which described the initial condition of the interface
2. Density ratio of light to the dark fluid, $\tilde{\mu} = \frac{\rho_G}{\rho_L}$
3. Viscosity ratio of light to dark fluid, $\tilde{\mu} = \frac{\mu_G}{\mu_L}$
4. Reynolds number defined with dark fluid scales $Re = \frac{L\mu\rho_L}{\mu_L}$
5. CFL number
6. Domain size
7. Mesh - No of Cells

Update time step and fluid properties

The time step is varied to follow the CFL criteria to keep the advection scheme numerically stable. The time step is calculated by

$$\Delta t = CFL \left(\frac{\Delta}{2u_{max}} \right) \quad (2.38)$$

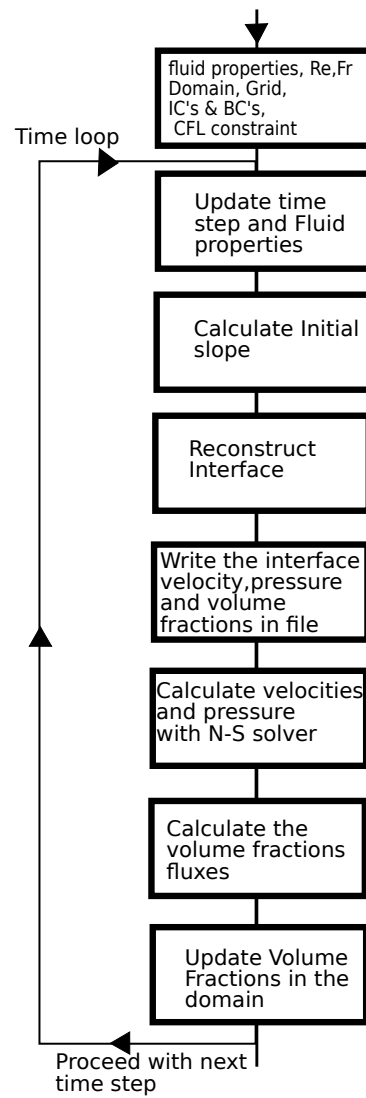


Figure 2.5: Sequence to couple VOF and N-S solver

where u_{max} is the maximum velocity in the actual domain. CFL number should be less than 1.

The fluid properties i.e. density $\tilde{\rho}$, and viscosity $\tilde{\mu}$ field is updated at every time step using the volume fraction field, which describes the properties of the fluid in the cells where velocities has to be computed.

$$\begin{aligned}\rho_{r,c} &= F + (1 - F)\tilde{\rho} \\ \mu_{r,c} &= F + (1 - F)\tilde{\mu}\end{aligned}\tag{2.39}$$

Calculate Initial slope

Initial guess is calculated using Green-Gauss gradient given by the Equation 1.4

Reconstruct Interface

Reconstruct the interface using the LVIRA in all the cells in the actual domain.

Write the data

Following data is stored in a file at required intervals. This file can be used to restart the simulation and march further in time.

1. x-velocity
2. y-velocity
3. Pressure
4. Volume Fraction

Interface information i.e. the coordinates of the lines in the domain is stored in a different file.

Calculate velocities

The x and y velocities are calculated using the projection method as discussed above in this chapter.

Calculate the volume fraction fluxes

The fluxes are calculated in all the cells actual domain using the velocity field calculated in previous step.

Update the volume fractions

The volume fractions are updated using the fluxes calculated above and the proceed to next time step in the loop by updating the fluid properties with updated volume fractions.

Verification

The test problems were set without any surface tension model and compared with respective available data.

Lid Driven Cavity test

A problem was set test lid driven cavity to get a steady state solution. A square domain of unit length, single fluid and Reynolds number 100, 400, 1000. The results are validated by Ghia *et al.* (1982). Figure 2.6 shows the x-velocity along vertical center-line and y-velocity along horizontal center-line respectively in the domain. Stream function and vorticity contours are shown in Figure 2.7.

Falling Droplet

Falling droplet problem was set up and compared with Gerris simulations with exactly same input conditions (See Figure 2.8).

- Domain: $[0,3] \times [0,3]$
- Droplet: Diameter = 0.3, Center (1.5,2.5)
- Grid size: 96 x 96
- Time Step 0.00125
- Density ratio $\frac{\rho_G}{\rho_L}, \tilde{\rho} = 0.5$
- Viscosity ratio $\frac{\mu_G}{\mu_L}, \tilde{\mu} = 1.0$
- $Re_L = 100$
- $Fr = 0.1$
- Boundary Condition : All walls no slip

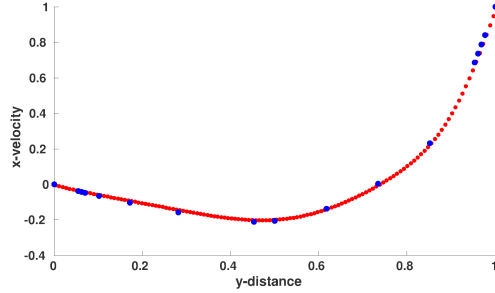
Rayleigh-Taylor Instability

Rayleigh-Taylor Instability is one of the most common test problems which are studied for verification, the problem was set up as in Rudman (1997)

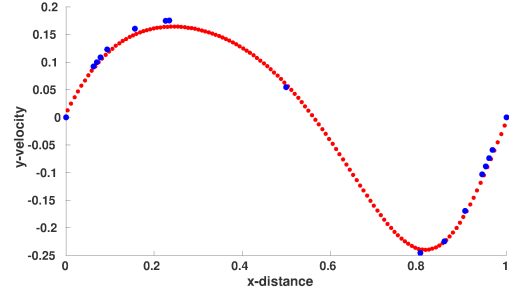
1. Domain: $[0,1] \times [0,3]$
2. Initial Interface: $y = 2 - 0.02\cos(\pi x)$
3. Grid size: 64 x 192
4. Time Step 0.005
5. Density ratio $\frac{\rho_G}{\rho_L}, \tilde{\rho} = \frac{5}{6}$
6. Viscosity ratio $\frac{\mu_G}{\mu_L}, \tilde{\mu} = 1.0$
7. $Re_L = 500$
8. $Fr = 0.5$
9. Boundary Condition : top,bottom and right wall as no slip, left wall free slip

Results are shown in Figure 2.9

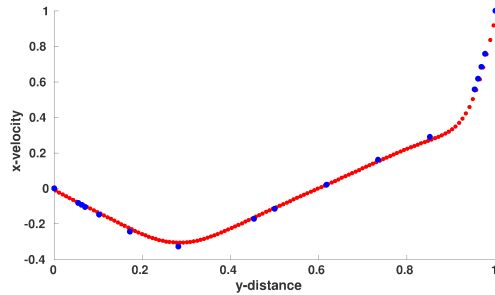
Similar problem with different initial and boundary conditions was studies and com-



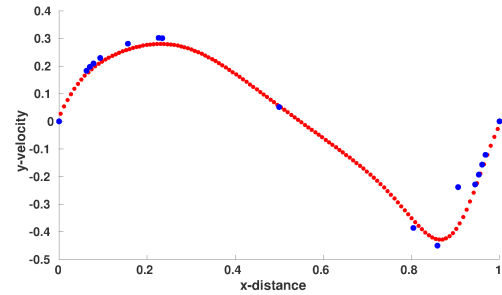
(a) x-velocity along vertical centerline (Re=100)



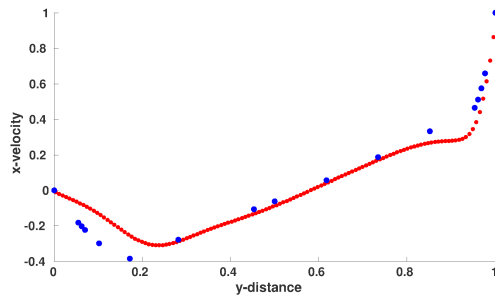
(b) y-velocity along horizontal centerline (Re=100)



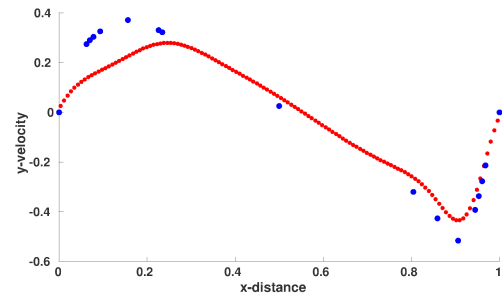
(c) x-velocity along vertical centerline (Re=400)



(d) y-velocity along horizontal centerline (Re=400)



(e) x-velocity along vertical centerline (Re=1000)



(f) y-velocity along horizontal centerline (Re=1000)

Figure 2.6: Blue:Ghia *et al.* (1982), Red:Present Study(LVIRA)

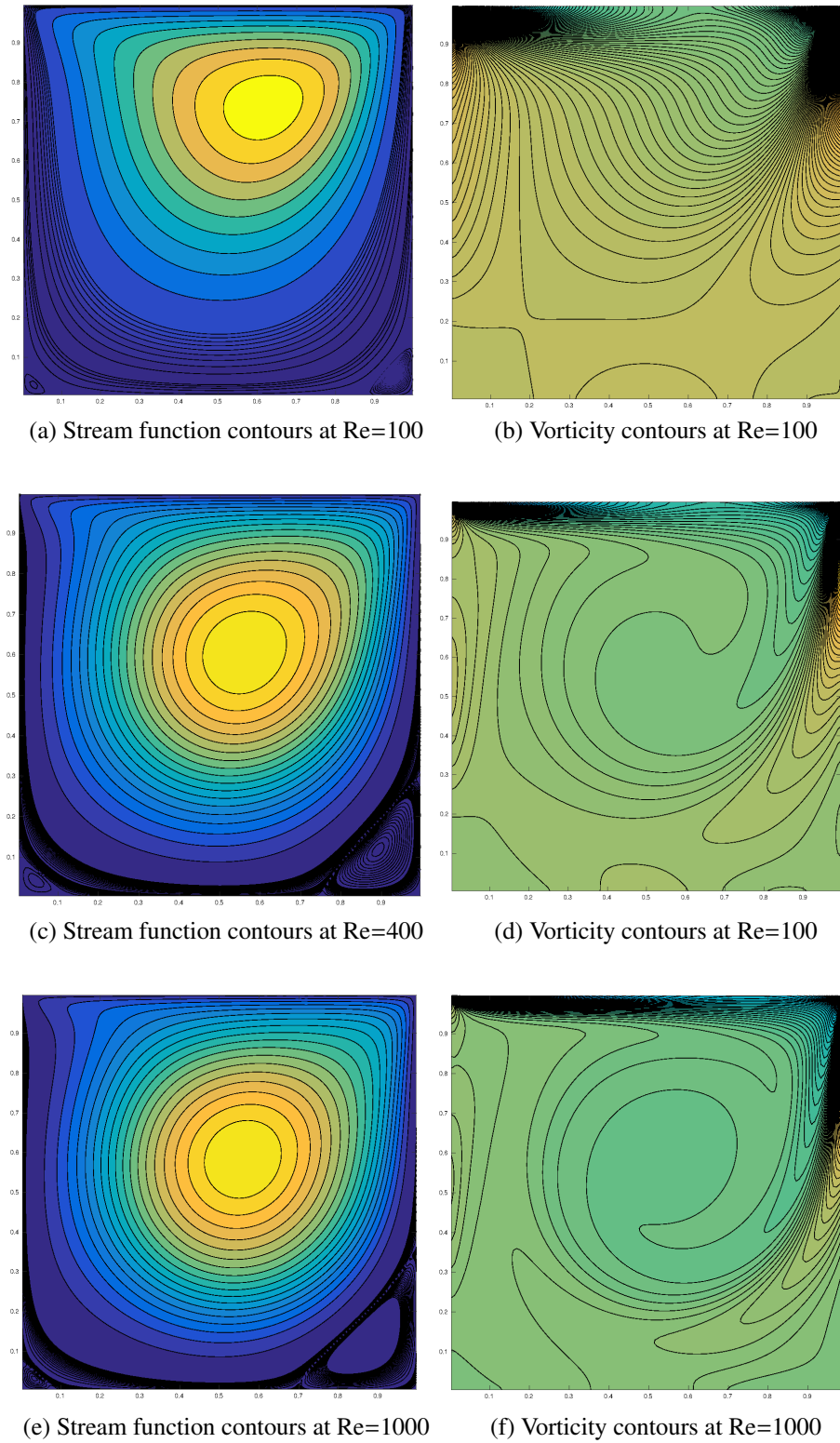


Figure 2.7: Stream function and vorticity contours

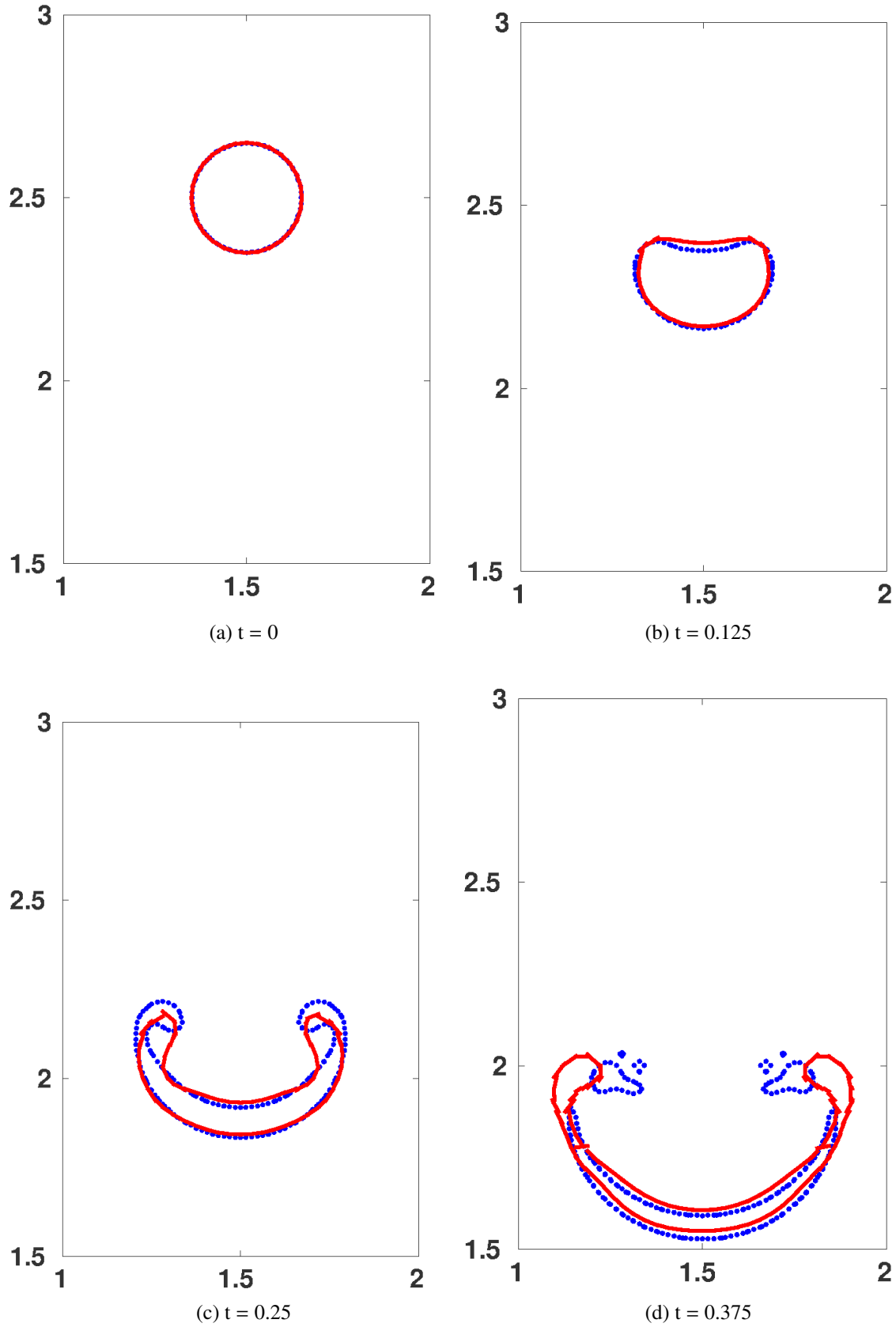


Figure 2.8: Falling Droplet test, (Blue-Gerris, Red-Present Study(LVIRA))

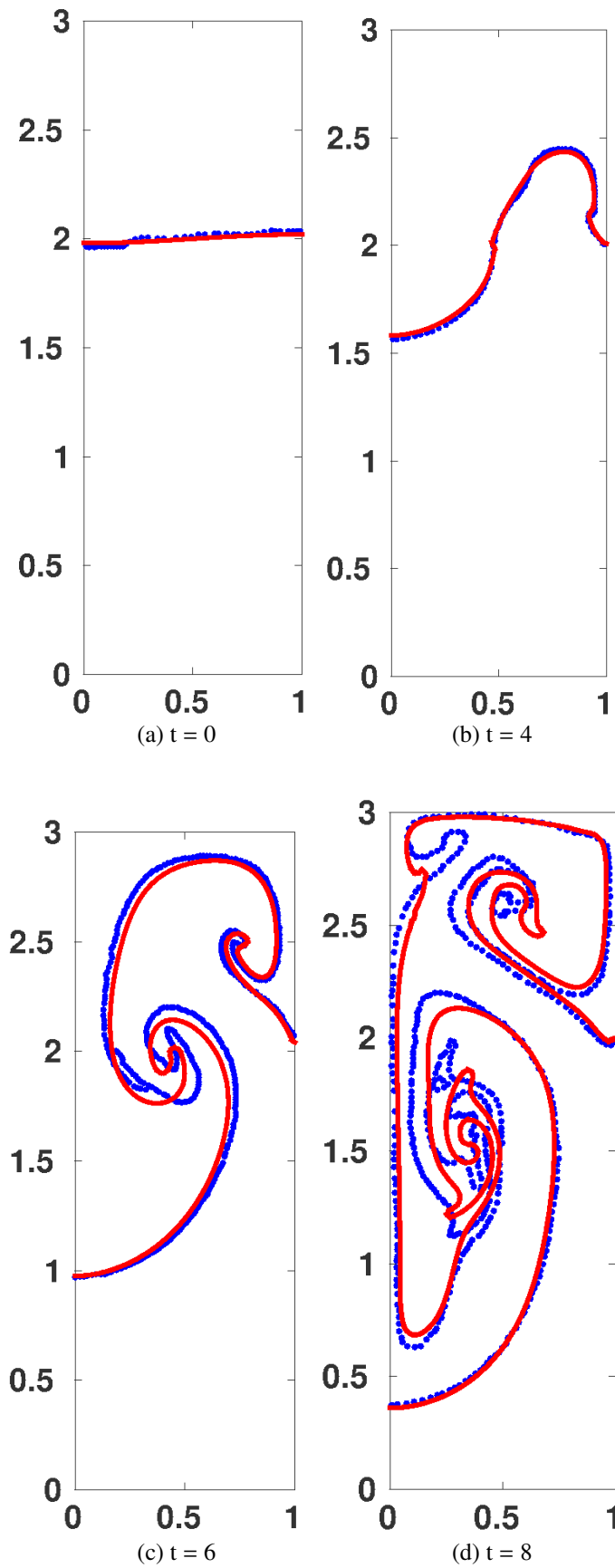


Figure 2.9: Comparison of Rayleigh-Taylor Instability test,(Blue-Rudman (1997), Red-Present Study(LVIRA))

pared with Smolianski (2001) as shown in Figure 2.10

1. Domain: $[0,1] \times [0,4]$
2. Initial Interface: $y = 2 + 0.05\cos(2\pi x)$
3. Grid size: 40×160
4. Time Step 0.001
5. Density ratio $\frac{\rho_G}{\rho_L}, \tilde{\rho} = \frac{17}{120}$
6. Viscosity ratio $\frac{\mu_G}{\mu_L}, \tilde{\mu} = 1.0$
7. $Re_L = \frac{1000}{3}$
8. $Fr = 1.0$
9. Boundary Condition : top and bottom wall as no slip, left and right wall free slip

Conclusion

The solver however produce satisfactory results while benchmarking but there is a need to improve the accuracy, for which we plan to implement advance advection schemes such as Godunov scheme. Also, Gauss-Sidel method used to solve pressure poisson equation converges very slowly and results in extensive use of computational resources and time. To improve the efficiency, we plan to use multigrid methods to solve the pressure poisson equation.

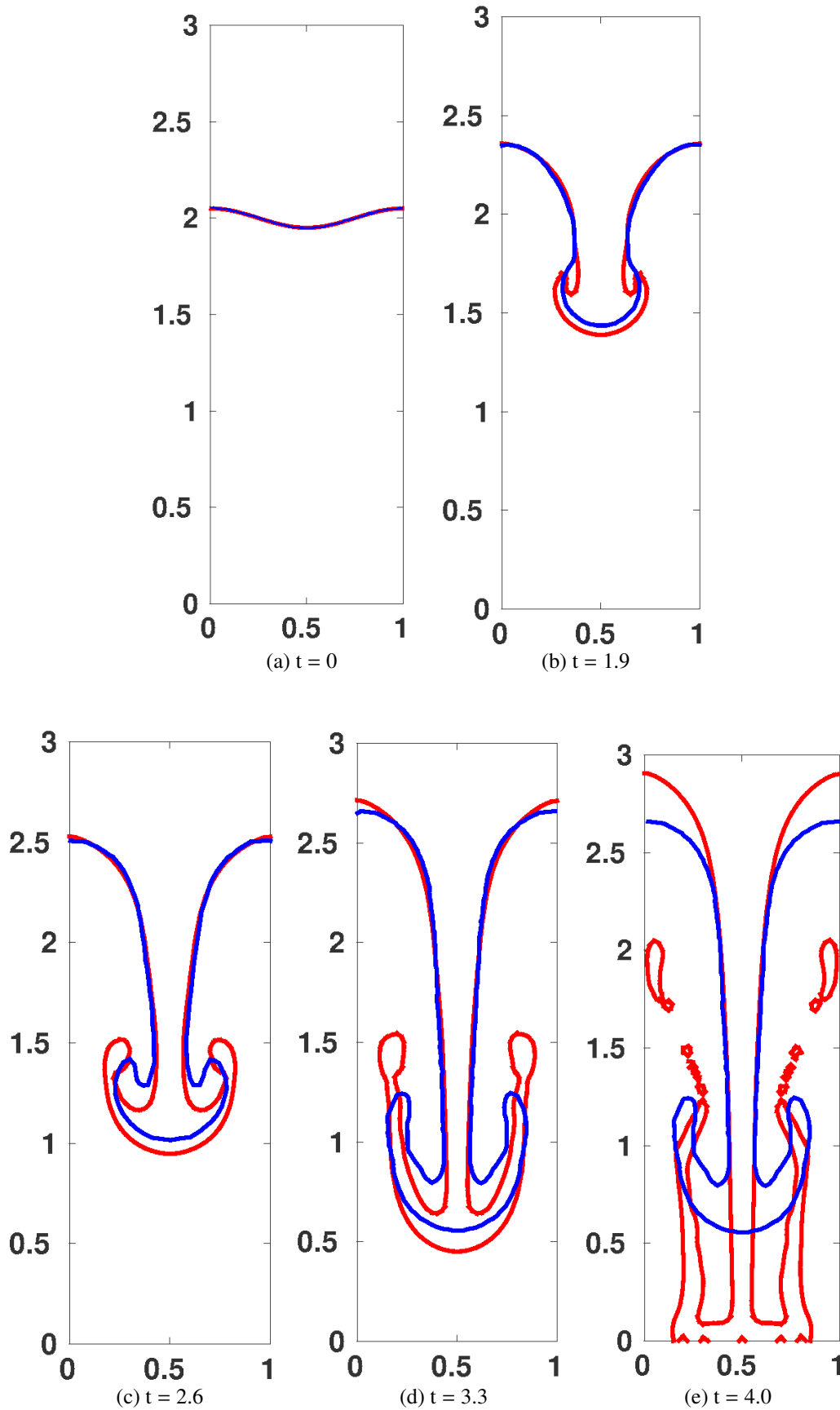


Figure 2.10: Comparison of Rayleigh-Taylor Instability test, (Blue-Smolianski (2001), Red-Present Study(LVIRA))

Chapter 3

Multigrid Methods

Introduction

In the previous chapter, we encountered Pressure Poisson's Equation (PPE) and used Successive Over Relaxation (SOR) method to reach the solution for pressure. However, we concluded that SOR has very slow convergence and faster methods should be used to solve the PPE. Fast Poisson solvers use advanced techniques, such as discrete Fast Fourier transform or cyclic reduction, multigrid methods and iterative Krylov-subspace methods.

Developed in late eighties (?), Multigrid methods has been used by many authors to solve PPE. We first discuss the characteristics of classic iterative methods and then show how the concepts from the analysis can be used to accelerate the convergence, and subsequently state the algorithm which we will implement in the solver to solve the PPE. The PPE has to be solved while solving the Navier-Stokes equation which is the most expensive part of the solution as it takes most of the time. Hence there is a need to solve this equation efficiently. Multigrid method has been developed for serial processor.

General Iterative Methods

As the discretised form of Poisson's equation has a system of linear algebraic equations. We consider a one dimensional Laplace equation as it is easier to study analytically.

$$\frac{d^2 u}{dx^2} = 0 \quad (3.1)$$

having boundary conditions, $u(0) = 0$ and $u(L) = 0$, which has the exact solution simply $u(x) = 0$. We now look into a numerical method to get the solution of this equation. First

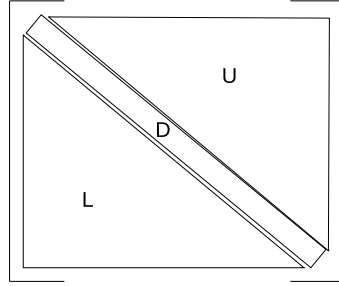


Figure 3.1: The matrix A

we discretise the differential equation and get algebraic equations.

$$\begin{aligned} \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} &= 0 \\ \text{or, } u_{i+1} - 2u_i + u_{i-1} &= 0 \\ \text{of form, } Ax &= b \end{aligned} \tag{3.2}$$

But the iterative methods don't solve $x = A^{-1}b$, rather the problem is formulated as $x = Px + Q$. These generally differ in the forms of P and Q . For Gauss-Seidel method, we can split the operator A , as shown in Fig.3.1 in strictly lower, diagonal and strictly upper parts i.e. $A = L + D + U$.

$$\begin{aligned} (D + L + U)x &= b \\ (D + L)x &= -Ux + b \\ x^{j+1} &= (D + L)^{-1}(-U)x^j + (D + L)^{-1}b \end{aligned} \tag{3.3}$$

From 3.3 we can see that $P = (D + L)^{-1}(-U)$ and $Q = (D + L)^{-1}b$.

For problem 3.2, the error is defined as $e^j = u^e - u^j$, where e^j is the error and u^j is the value of u at j^{th} iteration. The exact answer to the problem is known $u^e = 0$ at all x , hence the error, is simply $e^j = -u^j$.

We can now discuss the behavior of Gauss-Seidel method by starting with arbitrary initial guesses. To find the characteristics for different error profiles, we solve the problem with initial guesses given by,

$$u_i = \sin\left(\frac{k\pi x_i}{L}\right) \tag{3.4}$$

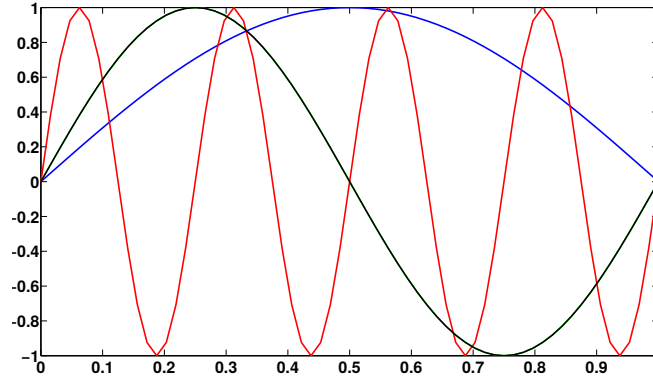


Figure 3.2: Fourier modes

In Eq. 3.4, Fourier modes and k is the wavenumber. Fig. 3.2 shows the modes over the domain for $k=1, 2$ and 8 . We can see for low values of k the error is smooth and for higher wavenumbers the profiles are oscillatory.

In Fig. 3.3, we can see for a given number of iterations the method converges faster for higher values of wavenumber in the error. To know the dependence of wavenumber on the convergence we have to analyse the characteristics of Gauss-Sidel method.

Convergence analysis of Gauss-Siedel method

We know that while approaching to the solution the error approaches to zero. We might want to look how the error decimates with respect to iterations.

$$e^n = P^n e^0 \quad (3.5)$$

Now, if we expand e^0 in eigen basis of P , we get,

$$e^0 = \sum C_k v_k \quad (3.6)$$

where C_k are the components of e^0 in eigen basis of and v_k are eigenvectors of iteration operator P . We know that P in its eigenbasis will only stretch or contract the components when operated on a vector (here e^0) by corresponding eigenvalues λ . Hence, 3.5 can be expressed as,

$$e^n = \sum \lambda_k^n C_k v_k \quad (3.7)$$

From 3.7, it can be seen that largest value of $|\lambda_k|$ should be less than 1, for the error to approach zero in successive iterations i.e. for convergence. This value is known as

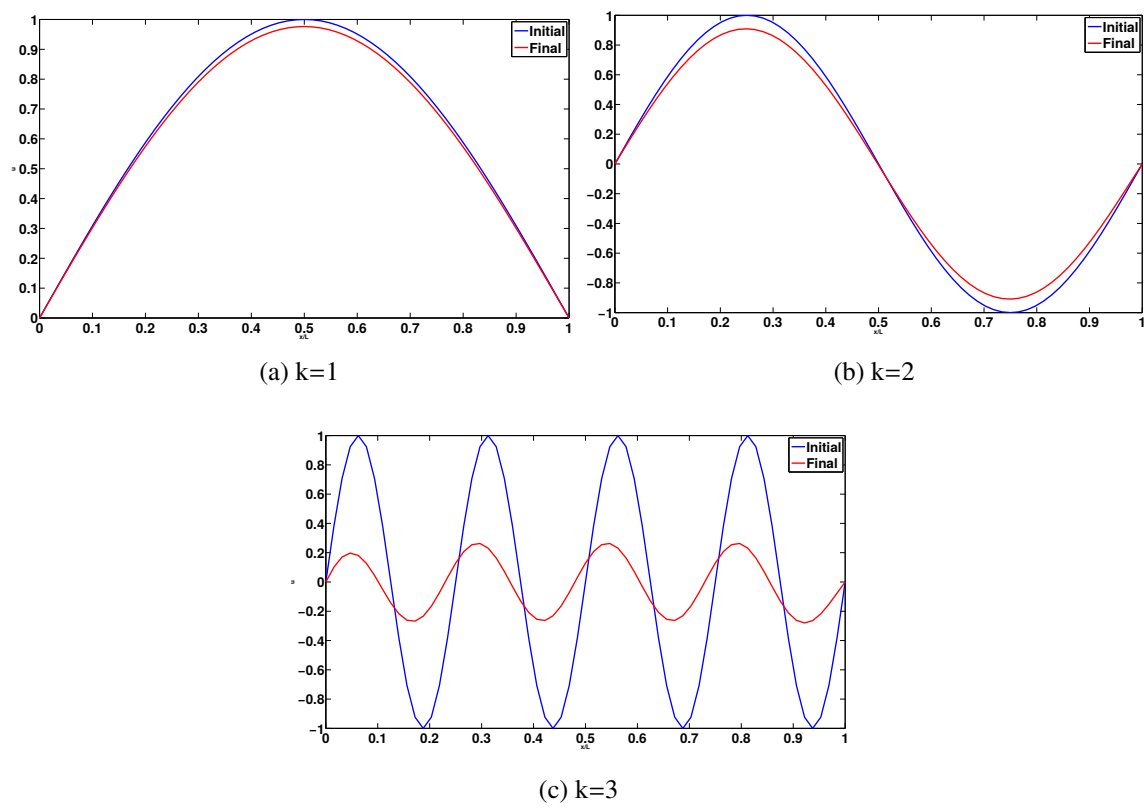
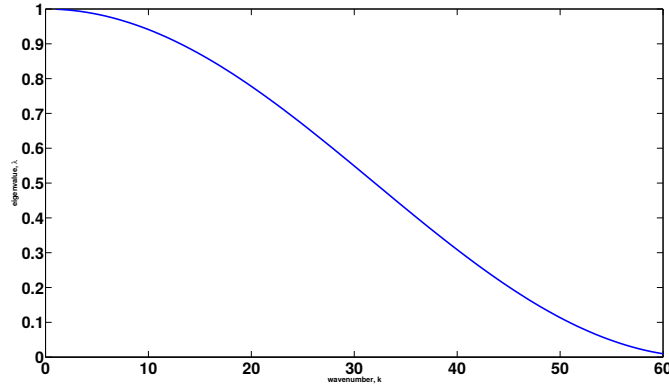


Figure 3.3: Profile after 10 Gauss-Siedel iterations, Blue - Initial guess, Orange - After 10 iterations

Figure 3.4: Eigenvalues λ with respect to k

the spectral radius of the iteration matrix P . It can also be concluded from here that the smaller the spectral radius faster the convergence. We have observed in the previous section that there is faster convergence for high wavenumber errors and we know that convergence is directly affected by the eigenvalues of the P operator. We now look what is the relation between the wavenumber and the eigenvalues of P .

The eigenvalues of Jacobi iteration operator is given by,

$$\lambda_k = 1 - \sin^2\left(\frac{k\pi}{2N}\right) \quad k = 1, 2, \dots, N \quad (3.8)$$

where N is the number of nodes in the grid. From 3.8 we plot λ vs k in 3.4, and we can now understand reasons for the results in previous section i.e. the eigenvalue is low for high wavenumber modes and hence we found faster convergence for high wavenumbers. In practical problems we don't have the solution, we need to find it and most of the times it is not even possible to give a good guess. Most of the times we start with zero values of the variable to be calculated. Hence when we start with an arbitrary guess and it can have errors of many wavenumbers. With this argument we cannot do much about increasing the wavenumber or the oscillatory behavior of the error. But we can again look at 3.8, the convergence depends on low values of λ , but it is not entirely dependent on wavenumber k but the ratio $\frac{k}{N}$ (See Fig. 3.5). For a given wavenumber $k = 1$, for which convergence is low, See Fig. 3.6 it can be said that low wavenumber component will converge faster on coarser grids. This result lays down the foundation for multigrid methods. In literature we can find many variants of multigrid methods but the prime reason is buried in the dependence of eigenvalues of the iteration operator on grid size.

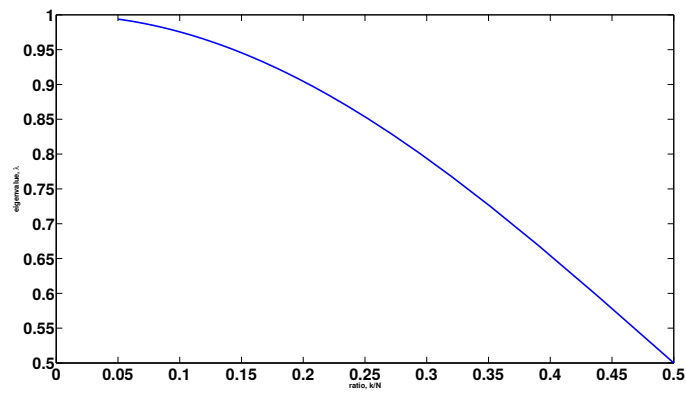


Figure 3.5: Eigenvalues λ with respect to $\frac{k}{N}$

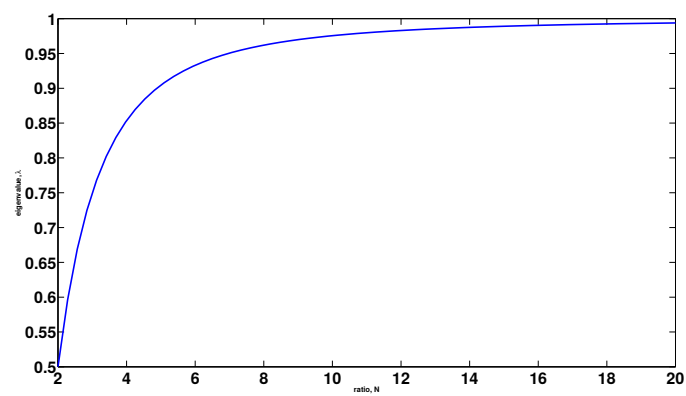


Figure 3.6: Eigenvalues λ with respect to N

Algorithm of Multigrid method

In the light of above results, we can think of an strategy to exploit the faster convergence characteristics of the method. It could be to solve the problem on coarser grids and interpolate the result on finer grid. This would be a better guess for finer grid and we can repeat the process till the finest grid when we want the solution. The disadvantages of this strategy are:

1. The problem is solved fully on coarser grids even we are only interested in the solution on finest grid.
2. It does not make use of any guess which we can have from finer grids.
3. Sometimes the problem is not well resolved on coarser grids and hence the guess provided by the coarser grid solution might not be a good guess.

A better strategy is to coarse grid correction, which calculates the error on coarser grids and provide a correction on the finest grid. We have defined the error as:

$$\begin{aligned}
 \text{exact solution} \quad Ax^e &= b \\
 \text{solution at } k^{\text{th}} \text{ iteration} \quad Ax^k &= b - r \\
 \text{error at } k^{\text{th}} \text{ iteration} \quad e^k &= x^e - x^k \\
 \text{from above, we get,} \quad A(x^e - x^k) &= r \\
 \text{or,} \quad Ae^k &= r
 \end{aligned} \tag{3.9}$$

We solve the given problem $Ax = b$ on the finest grid, and $Ae = r$ on the coarser grids. The basic steps in the algorithm are:

1. Iterate for few steps the given problem on the finest grid. This step is **relaxation**.
2. Transfer the residue from finest grid to coarser grid. This step is called as **restriction**.
3. Interpolate the values of error and add it previous values to make a correction. This step is **prolongation**.

The algorithm for multigrid is followed as in Fig. 3.7. This is V cycle algorithm, where the restriction is done till the coarsest grid and then prolongation till the finest grid.

Sample Problem

We try to solve to some problems to understand the implementation details of the algorithm. Consider a one dimensional Poisson's equation 3.10,

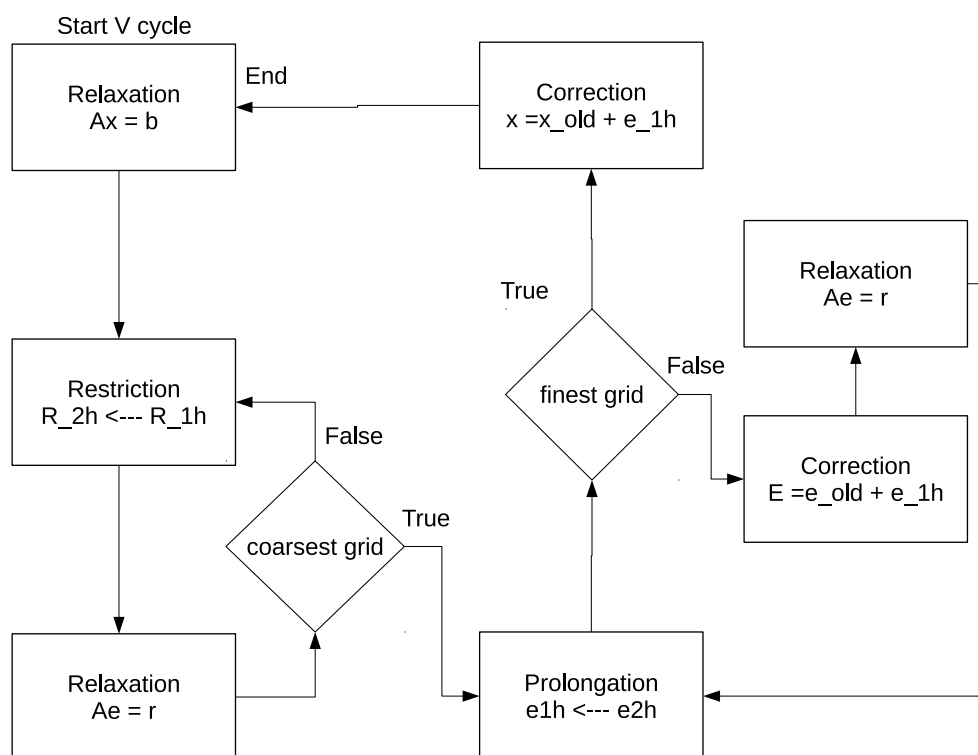


Figure 3.7: Multigrid Algorithm

One dimensional Poisson's equation

$$\frac{d^2u}{dx^2} = \frac{1}{2}[\sin\pi x + \sin 16\pi x] \quad (3.10)$$

having boundary conditions $u(0) = u(1) = 0$. We can discretise Eq. 3.9 as,

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = \frac{1}{2}[\sin\pi x_i + \sin 16\pi x_i] \quad (3.11)$$

Let the RHS be q_i at i^{th} node. Then we can write for u_i from 3.11 as,

$$u_i^{n+1} = \frac{1}{2}(u_{i+1}^n + u_{i-1}^{n+1} - h^2 q_i) \quad (3.12)$$

Eq. 3.12 is iteration equation for Gauss-Sidel method but for multigrid we iterate only few times. Here we will iterate only once in each step in a V-cycle. We have the finest grid having $N = 64$ with total nodes $N + 1$ i.e. 65. We now restrict the residue to the coarser grid which has $N/2+1$ nodes. We can use average restriction as,

$$r_i^{2h} = \frac{1}{4}(r_{2i-1}^h + 2r_{2i}^h + r_{2i+1}^h) \quad (3.13)$$

Then iterate for error on the coarser grid as,

$$e_i^{n+1} = \frac{1}{2}(e_{i+1}^n + e_{i-1}^{n+1} - h^2 r_i) \quad (3.14)$$

restriction and iteration continues till the coarsest grid. At coarsest grid the iteration the error equation is solved exactly. Then the error is interpolated backwards for prolongation step as,

$$\begin{aligned} e_i^h &= e_i^{2h} \quad i = 0, \dots, N \\ e_{2i+1}^h &= \frac{e_i^{2h} + e_{i+1}^{2h}}{2} \quad i=0, \dots, N \end{aligned} \quad (3.15)$$

While prolongation iteration step is taken at every grid level and at finest grid, we add the interpolated error to the u , again start with the V-cycle with the new u as a guess. The results for this problem are compared with ? (See Fig. 3.8).

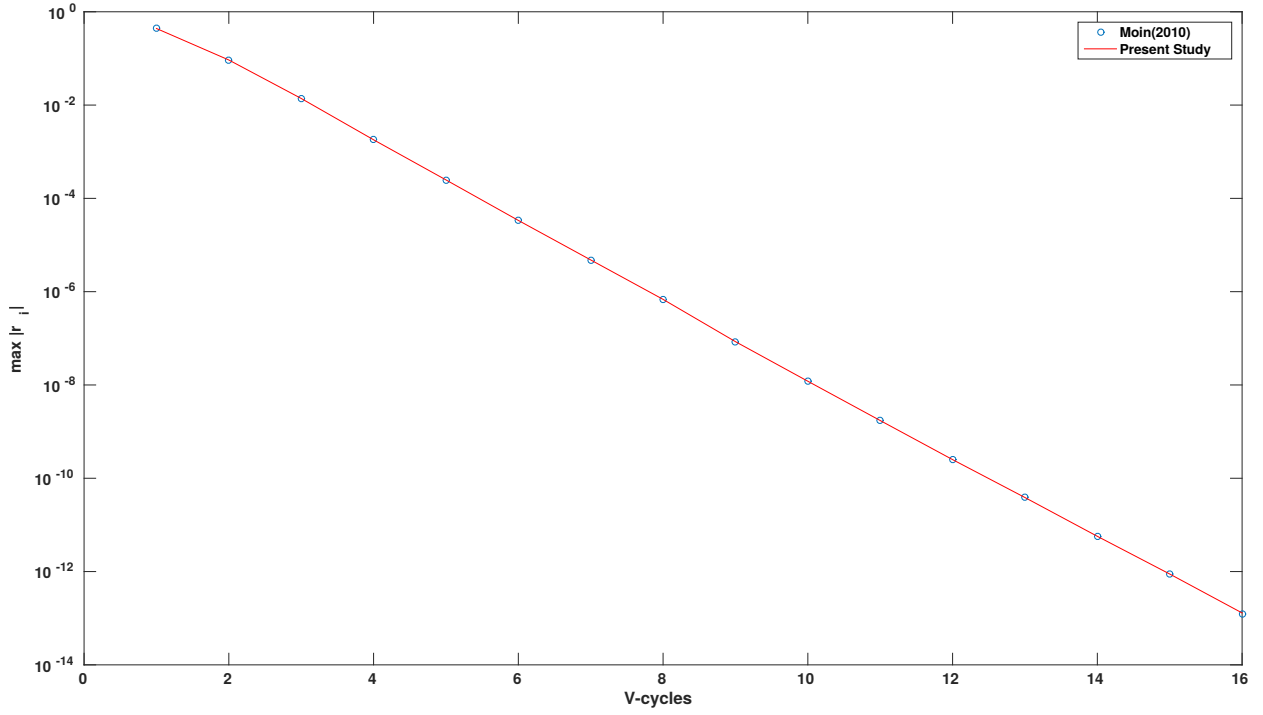


Figure 3.8: Comparison with Moin(2010)

Two dimensional Poisson's equation

We are interested in solving the Poisson equation, Equation 3.16, in a square domain of size unity centered on unity with Neumann boundary conditions on all sides (?).

$$\nabla^2 \phi = f(x, y) \quad (3.16)$$

Source term $f(x, y)$ is given by

$$f(x, y) = -\pi^2(k^2 + l^2) \sin(\pi kx) \sin(\pi ly) \quad (3.17)$$

with $k = l = 3$. Exact solution of the Poisson equation with this source term is

$$\phi(x, y) = \sin(\pi kx) \sin(\pi ly) + \kappa \quad (3.18)$$

where κ is an arbitrary constant.

\bullet $p(N+1,0)$	\bullet $p(N+1,1)$	\bullet $p(N+1,c-1)$	\bullet $p(N+1,c)$	\bullet $p(N+1,c+1)$	\bullet $p(N+1,N)$	\bullet $p(N+1,N+1)$
\bullet $p(N,0)$	\bullet $p(N,1)$	\bullet $p(N,c-1)$	\bullet $p(N,c)$	\bullet $p(N,c+1)$	\bullet $p(N,N)$	\bullet $p(N,N+1)$
\bullet $p(r+1,0)$	\bullet $p(r+1,1)$	\bullet $p(r+1,c-1)$	\bullet $p(r+1,c)$	\bullet $p(r+1,c+1)$	\bullet $p(r+1,N)$	\bullet $p(r+1,N+1)$
\bullet $p(r,0)$	\bullet $p(r,1)$	\bullet $p(r,c-1)$	\bullet $p(r,c)$	\bullet $p(r,c+1)$	\bullet $p(r,N)$	\bullet $p(r,N+1)$
\bullet $p(r-1,0)$	\bullet $p(r-1,1)$	\bullet $p(r-1,c-1)$	\bullet $p(r-1,c)$	\bullet $p(r-1,c+1)$	\bullet $p(r-1,N)$	\bullet $p(r-1,N+1)$
\bullet $p(1,0)$	\bullet $p(1,1)$	\bullet $p(1,c-1)$	\bullet $p(1,c)$	\bullet $p(1,c+1)$	\bullet $p(1,N)$	\bullet $p(1,N+1)$
\bullet $p(0,0)$	\bullet $p(0,1)$	\bullet $p(0,c-1)$	\bullet $p(0,c)$	\bullet $p(0,c+1)$	\bullet $p(0,N)$	\bullet $p(0,N+1)$

Figure 3.9: 2D uniform mesh featuring a discretization with a 5-point stencil.

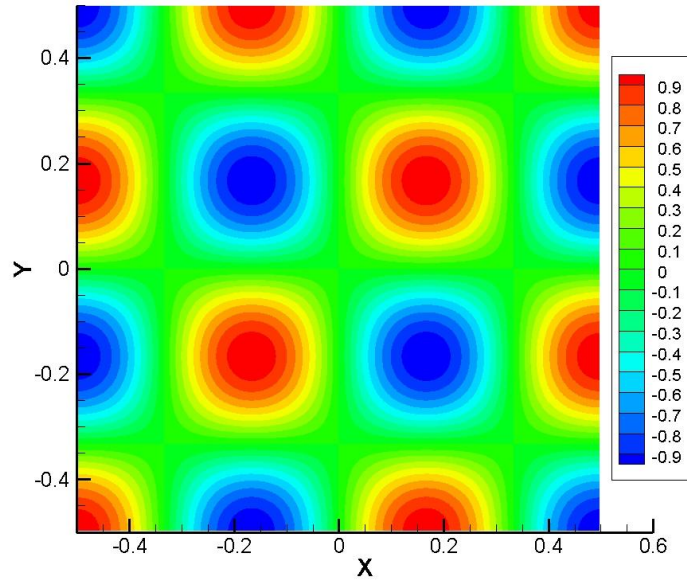
Discretization

Problem domain, shown in Figure 3.9, is discretized using a 2nd-order finite difference approximation on a cartesian grid having N nodes in both x - and y -directions which corresponds to a uniform grid spacing h . The value of ϕ on the 2D cartesian mesh can then be approximated for each node i,j in the interior of the computational domain as

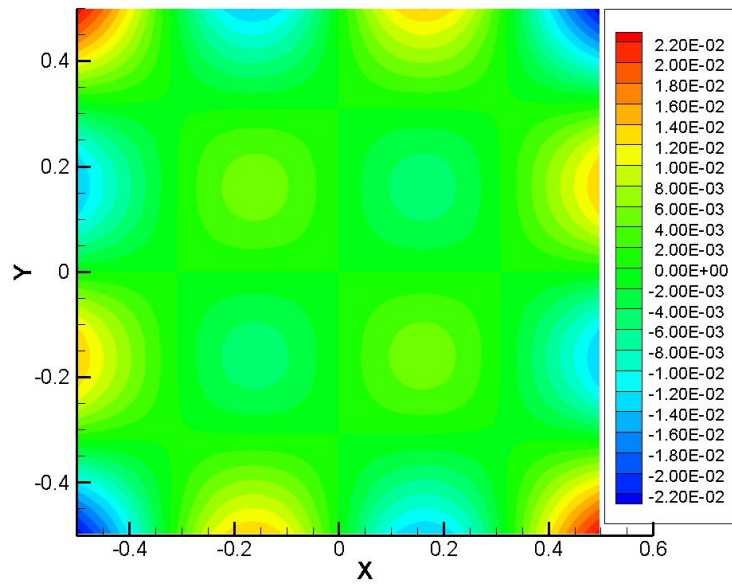
$$\frac{1}{h^2} (\phi_{i-1,j}^n + \phi_{i,j-1}^n - 4\phi_{i,j}^{n+1} + \phi_{i+1,j}^n + \phi_{i,j+1}^n) = f_{i,j}, \quad i, j = 2, \dots, N-1 \quad (3.19)$$

where the subscripts i and j represent the indices of the current node in the computational domain for the x and y directions, respectively.

Validation



(a)



(b)

Figure 3.10: Solver results of for $N = 500$ and $np = 4$; (a) Poisson solution and (b) solution error.

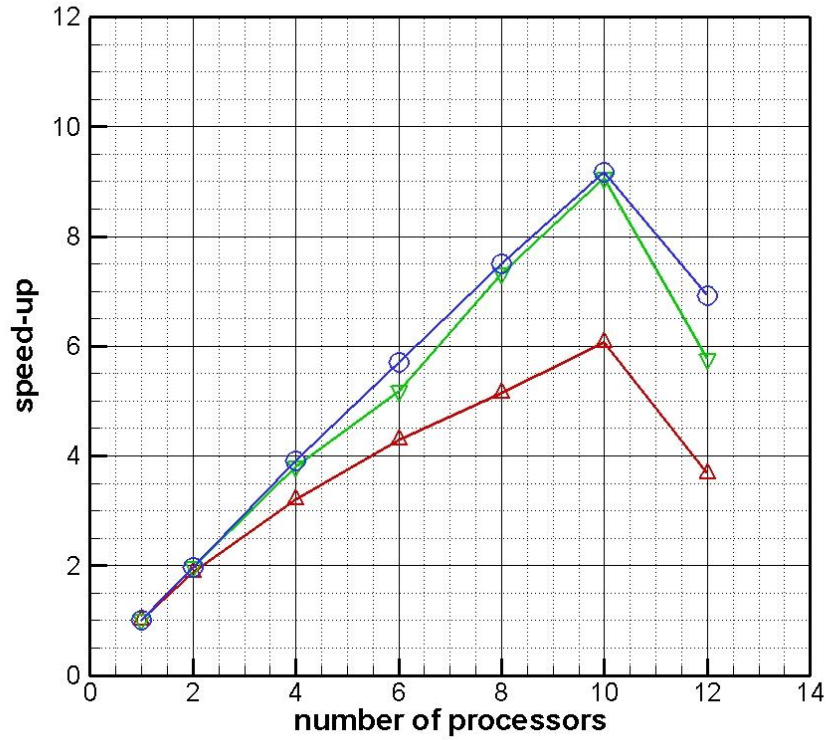


Figure 3.11: Speedup values for MPI parallel (SOR) Poisson equation solver.

Comparison with SOR Parallel

The same problem is solved using SOR (MPI) and the results are compared.

The following tests performed on the a machine having the following specifications:

1. Processor: Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.50GHz
2. CPU Cores: 6 (6 more with hyperthread-Virtual cores)
3. RAM: 64 GB
4. Cache: 15360 KB

From Fig.3.11 it can be observed that time of execution has decreased with increase in number of processors and speed up is more for larger size problem. A decrease in speed up is seen when the number of processors were more than the physcial cores and the threads took more time to process. Fig. 3.12 has emphasizes the fact that multigrid on a single processor is far more efficient on a multicore SOR algorithm. Hence this work

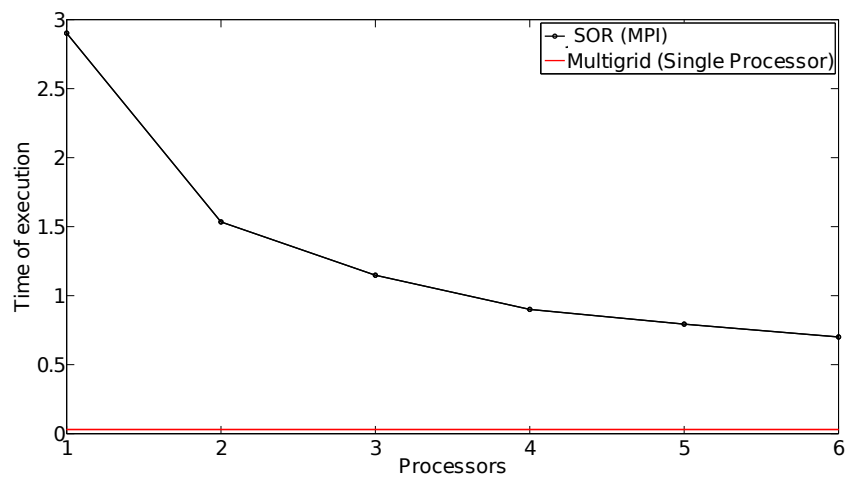


Figure 3.12: Comparison of SOR and Multigrid Time of execution

can be further extended for parallelizing the multigrid algorithm.

Chapter 4

Surface Tension

Height Function

The height function is calculated as shown in Fig. 4.1 (when absolute value of y-normal of interface is greater than x-normal), as

1. Start the summation from the middle row
2. Sum upwards as long as the cells contain dark fluid
3. Sum downwards as long as the cells contain light fluid, but subtract one for each cell added
4. The sums result in a histogram which approximates the interface

The radius of curvature is given by,

$$\kappa = -\frac{h''(x)}{[1 + (h'(x))^2]^{3/2}} \quad (4.1)$$

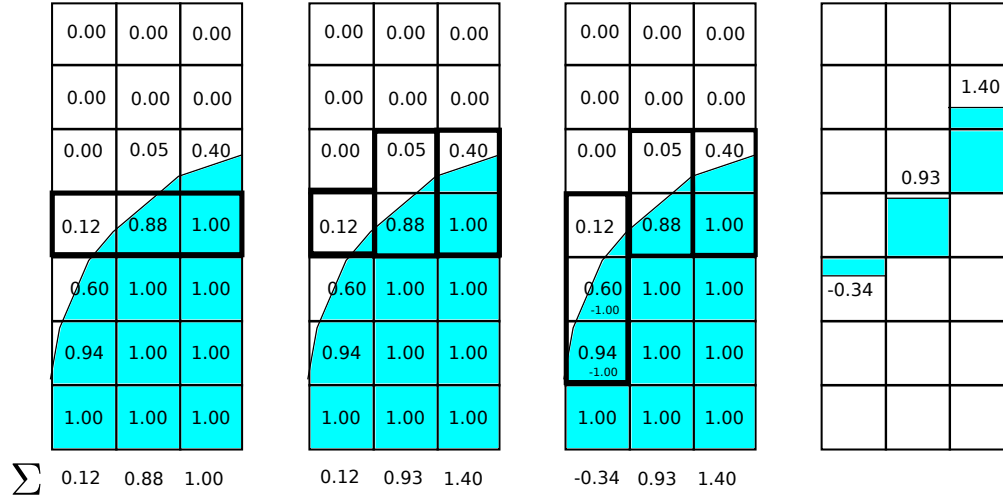


Figure 4.1: Variable Stencil for calculation of height functions

Algorithm 1 Height Function algorithm

```

1: procedure HEIGHT( $F$ )                                ▶ Volume Fractions as argument
2:   for Traverse through all cells do
3:
4:     if  $0 < F < 1$  then                                ▶ if cell contains the interface
5:        $d = \max(|nx|, |ny|)$                                 ▶ find the direction of maximum normal
6:
7:       while  $F > 0$  do
8:          $H = \sum_d F_d$ 
9:          $d = d + 1$ 
10:      end while
11:
12:      while  $F > 0$  do
13:         $H = \sum_d (F - 1)$ 
14:         $d = d - 1$ 
15:      end while
16:
17:    end if
18:  end for
19:  return  $H$                                               ▶ The Height Function is  $H$ 
20: end procedure

```

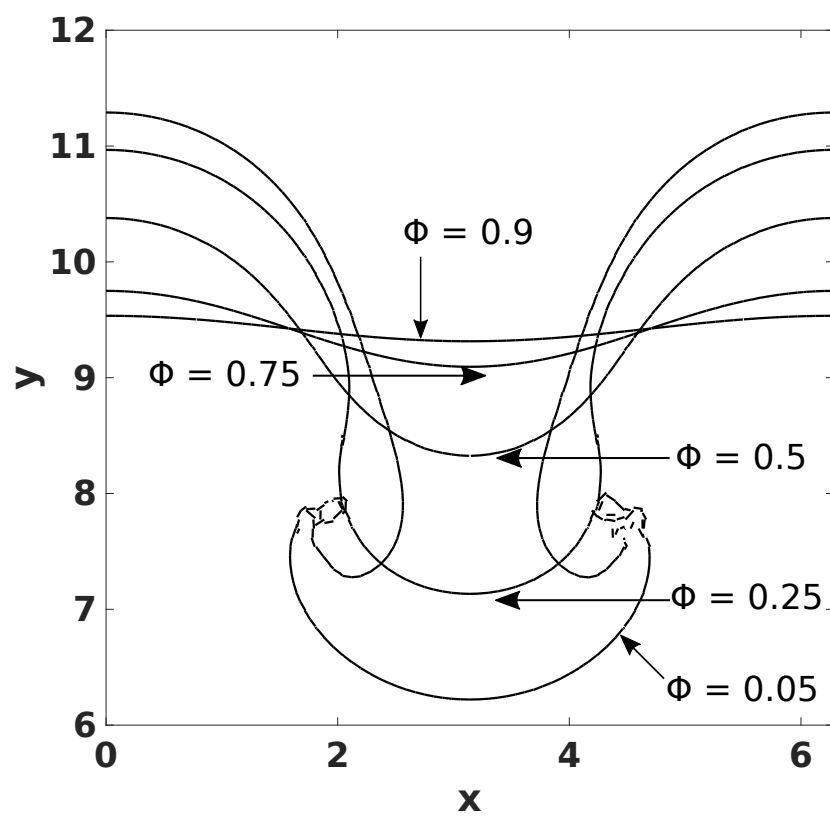


Figure 4.2: Interface shapes for different values of Φ using Height Function at $\tau = 6$

Table 4.1: Comparison of spurious currents, The capillary number is computed by different surface tension methods

Case	La	$\frac{\rho_{liq}}{\rho_{gas}}$	$\frac{\mu_{liq}}{\mu_{gas}}$	$\frac{D}{h}$	CSS	CSF	FT	CSF ¹	HF ²
A	0.357	1	1	32	3×10^{-3}	1.2×10^{-2}	3.0×10^{-4}	1.22×10^{-2}	NA
B	2×10^6	1	1	32	3×10^{-4}	5.0×10^{-4}	1.5×10^{-4}	disintegrates	NA
C	2×10^6	10^3	1	32	disintegrates	3.5×10^{-3}	1.1×10^{-3}	1.5×10^{-3}	NA
D	2×10^6	10^3	50	32	disintegrates	4.5×10^{-3}	blows up	1.6×10^{-3}	NA

Table 4.2: Comparison of growth rate n among the various numerical methods (PROST, CLSVOF, K_8 and HF) and analytical results with respect to relative importance of surface tension Φ

Method	Grid	$\Phi = 0.05$	$\Phi = 0.25$	$\Phi = 0.5$	$\Phi = 0.75$	$\Phi = 0.9$
PROST	20 x 60	5.7%	6.2%	6.1%	6.0%	3.5%
	40 x 120	2.3%	2.4%	2.7%	2.7%	0.9%
	80 x 240	1.0%	1.0%	0.9%	1.9%	0.9%
CLSVOF	20 x 60	7.4%	7.7%	8.5%	10.1%	15.3%
	40 x 120	2.7%	3.4%	4.4%	5.2%	7.5%
	80 x 240	0.8%	1.5%	2.1%	2.7%	3.5%
K_8	20 x 60	8.7%	8.1%	9.1%	1.4%	26.0%
	40 x 120	3.6%	3.4%	3.8%	2.2%	26.0%
	80 x 240	1.5%	1.0%	2.1%	3.0%	29.0%
HF	20 x 60	5.7%	6.2%	6.1%	6.0%	3.5%
	40 x 120	2.3%	2.4%	2.7%	2.7%	0.9%
	80 x 240	1.0%	1.0%	0.9%	1.9%	0.9%
Exact n		2.365	2.101	1.716	1.213	0.767

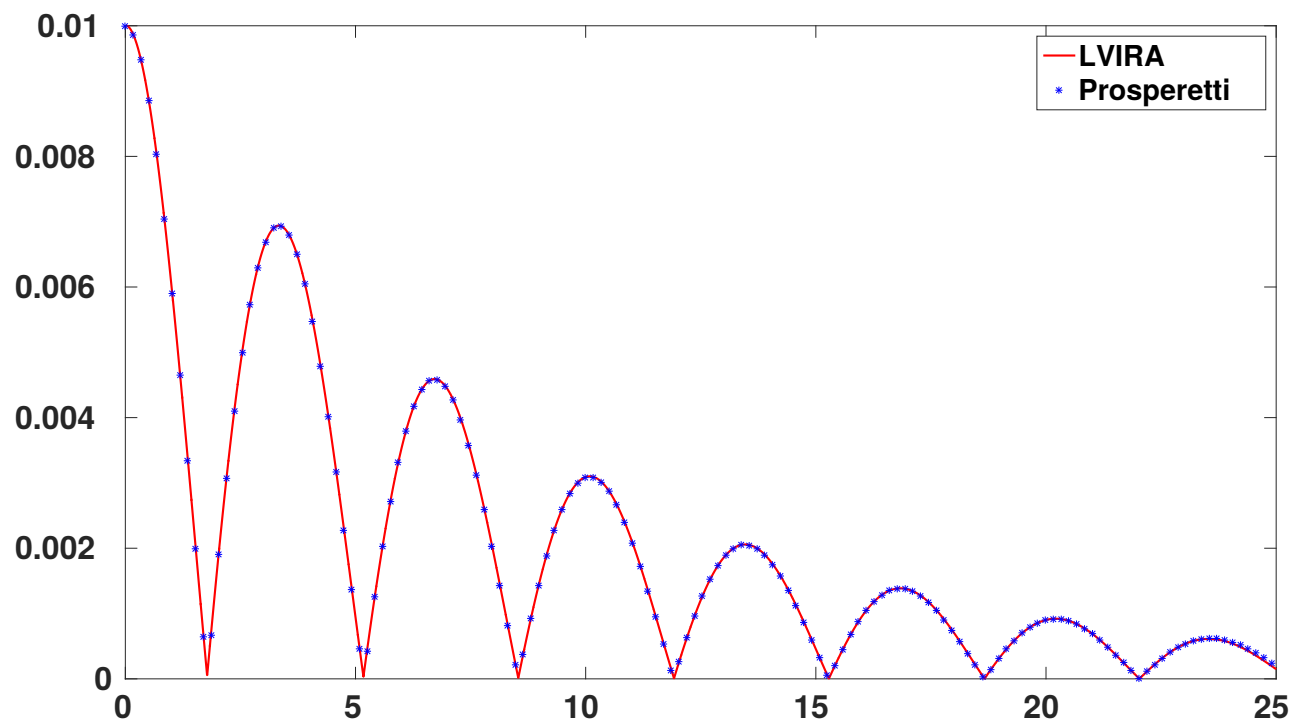


Figure 4.3: Comparison with analytical results

References

- Anderson, J., 1995, *Computational Fluid Dynamics*, Computational Fluid Dynamics: The Basics with Applications (McGraw-Hill Education) ISBN 9780070016859
- Gerlach, D, G Tomar, G Biswas, and F Durst, 2006, “Comparison of volume-of-fluid methods for surface tension-dominant two-phase flows,” *International Journal of Heat and Mass Transfer* **49**, 740–754
- Ghia, UKNG, Kirti N Ghia, and CT Shin, 1982, “High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method,” *Journal of computational physics* **48**, 387–411
- Hirt, C W, 1981, “Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries,” *J COMPUT PHYS* **225**, 201–225
- Jiang, Guang-Shan, and Chi-Wang Shu, 1995, *Efficient implementation of weighted ENO schemes.*, Tech. Rep. (DTIC Document)
- Leal, L.G., 2007, *Advanced Transport Phenomena: Fluid Mechanics and Convective Transport Processes*, Cambridge Series in Chemical Engineering (Cambridge University Press) ISBN 9781139462068
- Liu, Xu-Dong, Stanley Osher, and Tony Chan, 1994, “Weighted essentially non-oscillatory schemes,” *Journal of computational physics* **115**, 200–212
- Lörstad, Daniel, and Laszlo Fuchs, 2004 Oct., “High-order surface tension VOF-model for 3D bubble flows with high density ratio,” *Journal of Computational Physics* **200**, 153–176

- Pilliod, James Edward, and Elbridge Gerry Puckett, 2004 Sep., “Second-order accurate volume-of-fluid algorithms for tracking material interfaces,” *Journal of Computational Physics* **199**, 465–502
- Puckett, Elbridge Gerry, Ann S. Almgren, John B. Bell, Daniel L. Marcus, and William J. Rider, 1997 Jan., “A High-Order Projection Method for Tracking Fluid Interfaces in Variable Density Incompressible Flows,” *Journal of Computational Physics* **130**, 269–282
- Rudman, Murray, 1997, “Volume-tracking methods for interfacial flow calculations,” *International journal for numerical methods in fluids* **24**, 671–691
- Smolianski, Anton, 2001, *Numerical modeling of two-fluid interfacial flows*, PhD thesis (University of Jyväskylä)
- Tryggvason, G., R. Scardovelli, and S. Zaleski, 2011, *Direct Numerical Simulations of Gas–Liquid Multiphase Flows* (Cambridge University Press) ISBN 9781139496704
- Youngs, David L, 1982, “Time-dependent multi-material flow with large fluid distortion,” *Numerical methods for fluid dynamics* **24**, 273–285
- Zalesak, Steven T, 1979, “Fully multidimensional flux-corrected transport algorithms for fluids,” *Journal of computational physics* **31**, 335–362