Numerical Simulation Of Two-Phase Viscoelastic Electrohydrodynamic Flow

THESIS

Submitted in partial fulfillment of the requirements of
BITS F421T Thesis

By

Yashas GANDHI
ID No. 2017A4TS0326P

Under the supervision of:

Dr. Shyam Sunder YADAV
Assistant Professor,
Mechanical Engineering Department

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, PILANI CAMPUS
May 2021

Acknowledgements

I would like to thank my supervisor Dr Shyam Sunder Yadav for guiding me throughout the course of the thesis. I would also like to thank Dr M S Dasgupta and other faculty members of Departmental Research Committee, Mechanical Engineering Department, BITS Pilani to let me take up this thesis.

Certificate

This is to certify that the thesis entitled, "*Numerical Simulation Of Two-Phase Viscoelastic Electrohydrodynamic Flow*" and submitted by Yashas GANDHI ID No. 2017A4TS0326P in partial fulfillment of the requirements of BITS F421T Thesis embodies the work done by him under my supervision.

Supervisor
Dr. Shyam Sunder YADAV
Assistant Professor,
Mechanical Engineering Department,
BITS Pilani, Pilani Campus
Date:

Thesis Title: Numerical Simulation Of Two-Phase Viscoelastic Electrohydrodynamic Flow
Supervisor: Dr. Shyam Sunder Yadav
Semester: Second **Session:** 2020-21
Name: Yashas Gandhi **ID No.:** 2017A4TS0326P

Abstract

Electrohydrodynamic flows are used in multiple industrial process like electro-spinning, electro-spray, etc. Thus, study of electrihydrodynamic flows bedcomes necessary for optimizing various industrial processes. A widely used computational fluid dynamics software *OpenFOAM*® can solve two-phase flows but lacks compatibilty for viscoelastic and elkectrohydrodynamic flows. A solver based on *rheoInterFoam*, an extension of *OpenFOAM*® that enables simulation of two-phase viscoelastic flows was developed to solve two-phase viscoelastic electrohydrodynamic flows. The solver uses Poisson equation to solve electric equations and volume of fluid method for describing the interface between the two phases. The solver is tested using an example of two water drops suspended in an Oldroyd-B fluid coalescing under the action of electric field.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The flows governed by electric forces are called Electrohydrodynamic (EHD) flows. Many industrial processes make use of electrospinning, electrospray and other such techniques where the fluid flow is manipulated by electric forces [1][2]. Thus, study of EHD flows becomes necessary to design and optimize several industrial processes.

OpenFOAM® is a widely used open source computational fluid dynamics software package. It is very flexible and allows users to make custom solvers and utilities for various type of flows. It can even solve two-phase flows. However, it lacks a solver for two-phase, viscoelastic EHD flows. Thus, the solver *rheoInterEHDFoam* was developed.

1.2 Literature Survey

J.M. López-Herrera, S. Popinet and M.A. Herrada have done extensive work on two-phase EHD flows and developed a solver for the same as an extension to Gerris flow solver but it lacks the capability to solve viscoelastic flows [3]. A third-party solver *rheoInterFoam*, a part of *rheoTool* [4] toolkit adds two-phase viscoelastic flow solving capabilities to the default *OpenFOAM*® package. However, it lacks the functionality to solve electric equations along with two-phase viscoelastic flows. Two other third-party solvers for *OpenFOAM*®- *interEHDFoam* [5] and *openFoamEHD* [6] are available, which solve two-phase EHD flows for Newtonian fluids but lack the capability to solve viscoelastic flows. Thus, the addition of capability to solve electric equations for two-phase viscoelastic flows to *rheoTool* was explored in this project.

1.3 Scope of Work

The solver uses Volume of Fluid (VOF) method to represent the surface between the interface of two phases [7]. Both Newtonian and non-Newtonaian viscosity models are available in *rheoTool*. Detailed information on each model can be found in Ref [4]. The solver uses a complete electrostatic model without any simplifications. The solver has been developed for *OpenFOAM v7*. A simple axis-symmetric case of coalescing of two water drops suspended in an Oldroyd B fluid under action of electric field has been used to test the solver.

1.4 Organization of Reoprt

The report is divided in five chapters. Chapter 1 being introduction, followed by chapter 2 on governing equations, chapter 3 on working of solver, chapter 4 on test case to verify the solver and finally chapter 5 containing conclusion and recommendations for future work.

Chapter 2

Governing Equations

2.1 General Two-Phase Flow

A two-phase, incompressible, immiscible, isothermal, transient and laminar flow is governed by mass conservation Eq. (2.1) and momentum conservation Eq. (2.2),

$$\nabla \cdot \mathbf{u} = 0 \tag{2.1}$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \nabla \cdot \boldsymbol{\tau}' + \mathbf{f} + \sigma \kappa \delta_s \mathbf{n} \tag{2.2}$$

where, $\mathbf{u}$ is the velocity vector, ρ is the density of fluid, t is time, p is pressure, $\boldsymbol{\tau}'$ is the extra-stress tensor, $\mathbf{f}$ is the external body-force, $sigma$ is surface tension coeffecient, κ is interface curvature and $\mathbf{n}$ is the normal to the interface. δ_s is Dirac delta function, whose value is 1 at interface and 0 everywhere else in the domain. This ensures that the surface tension acts only on the interface of the fluids. $\boldsymbol{\tau}'$ is the extra-stress tensor and is calculated differently by different viscosity models.

2.2 Electrohydrodynamic Two-Phase Flow

In EHD flows, in the Maxwell's electromagnetic equations, the magnetic effects can be ignored and approximated as electrostatic equations [8]. Thus the electrical phenomena can be described completely by Eq. (2.3) and Eq. (2.4),

$$\nabla \cdot (\epsilon \mathbf{E}) = \rho_e \tag{2.3}$$

$$\nabla \times \mathbf{E} = 0 \tag{2.4}$$

where, $\mathbf{E}$ is the electric field, ϵ is electric permitivitty of fluid and ρ_e is the electric charge per unit volume. In terms of electric potential in electrostatic limit, ϕ, electric field $\mathbf{E}$ can be written as $\mathbf{E} = -\nabla\phi$ and Eq. (2.3) can be written as Poisson equation Eq.(2.5),

$$\nabla \cdot (\epsilon \nabla \phi) = -\rho_e \tag{2.5}$$

The free charge conservation equation is given by Eq. (2.6),

$$\frac{\partial \rho_e}{\partial t} + \nabla \cdot (K\mathbf{E} + \rho_e \mathbf{u}) = 0 \tag{2.6}$$

where, K is the conductivity of the fluid. The body-force term $\mathbf{f}$ in Eq. (2.2) for electrohydrodynamic flow in absence of any other external forces is given by Eq. (2.7),

$$\mathbf{f} = \nabla \cdot \mathbb{T} \tag{2.7}$$

$$\mathbb{T} = \epsilon(\mathbf{E}\mathbf{E} - 0.5|\mathbf{E}|^2\mathbf{I}) \tag{2.8}$$

where, $\mathbb{T}$ is the electrostatic Maxwell Stress Tensor and $\mathbf{I}$ is 3×3 identity tensor.

For detailed derivation of the above equations see Ref. [3].

Chapter 3

rheoInterEHDFoam Solver

3.1 Accessing the Solver

The solver *rheoInterEHDFoam*'s source code and test case *dropCoalescence* developed as a part of this project can be accessed at `https://github.com/Yashas17/rheoInterEHDFoam`. The installation of the solver is explained in the *README.md* file published with the solver files. The solver has been developed for *OpenFOAM v7* only as it is the latest version that the base solver *rheoInterFoam* supports, although the latest version of *OpenFOAM*® is *OpenFOAM v8*.

3.2 Working of the Solver

It is assumed that the reader has basic knowledge of *OpenFOAM*® and *rheoTool*, if not the reader is advised to go through *OpenFOAM*® and *rheoTool* manual [9][4] before proceeding further.

Fig 3.1 shows a brief outline of the solver. The solver starts by verifying case files and throws an error message and terminates the program if the files are not in proper format or some file is missing. Once the case files are verified, the solver creates time variables followed by reading the mesh files and finally creates field variables such as pressure p, velocity $\mathbf{u}$, electric permitivitty ϵ, electric conductivity K, etc. Finally, the solver enters the time-stepping loop. The solver exits the loop when the Run Time of simulation equals the End Time. The maximum and minimum time step is defined in the *controlDict* file. Inside the loop, the solver calculates time-step, updates Run Time and enters Pressure-Velocity PIMPLE corrector loop which solves the governing equations. This loop is run *nInIter* times, refining the solution *nInIter* times per time-step. The variable *nInIter* is specified in *fvSolutions* file.

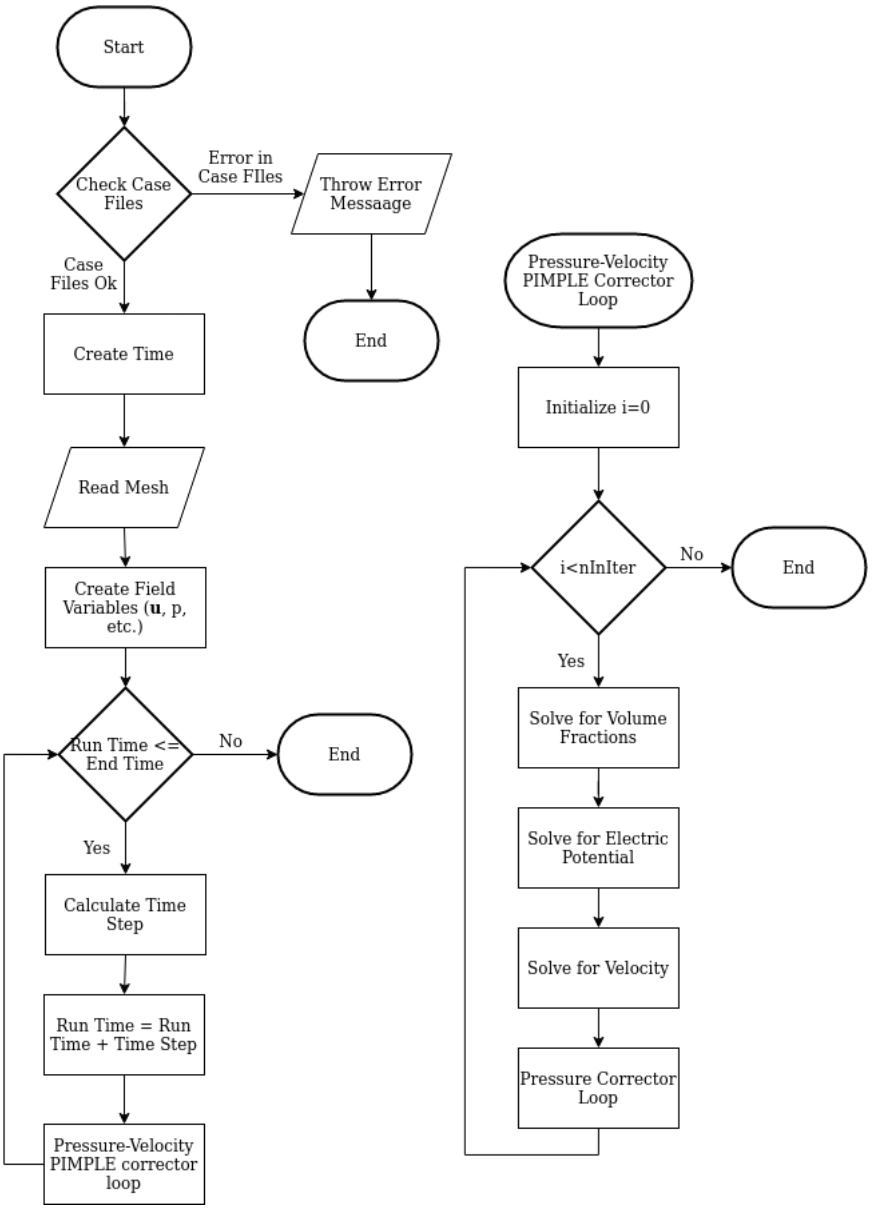


FIGURE 3.1: Flowchart of The Code

Inside the Pressure-Velocity PIMPLE corrector loop, first volume fraction of fluids is calculated followed by electric potential, velocity and finally pressure. The solver can use PIMPLE, SIMPLE, SIMPLEC and PISO algorithms, which is specified in *fvSolutions* file. The volume fraction of fluids is calculated using the VOF method. The electric potential is caculated by solving Eq (2.6) and (2.5). The velocity is calculated using Eq (2.2), (2.7) and (2.8).

3.3 Numerical Schemes

Various numerical schemes are available to solve time derivative, gradient, divergence, laplacian, interpolation and surface normal gradient. All the options available can be looked up in

OpenFOAM® manual [9]. The numerical schemes have to be specified in *fvSchemes* file which is read at the run-time.

3.4 Linear Solvers for Discretised Equations

OpenFOAM® offers a wide range of linear solvers for discretised equations. The user can choose which solver to use for each variable and also set its tolerance in the *fvSolution* file. The list of all solvers is available in Ref [9].

3.5 Defining Fluid Properties

Most *OpenFOAM*® solvers use *transportProperties* file for specifying fluid properties, however, all solvers which are a part of *rheoTool* toolkit use *constitutiveProperties* file for specifying fluid properties. Since *rheoInterEHDFoam* solves EHD flows, it needs additional parameters- electric permitivitty ϵ and electric conductivity K which are specified in the *parameters* sub-dictionary inside the fluid's dictionary in the *constitutiveProperties* file. The conductivity is specified by keyword *sig* and permitivitty is defined by keyword *eps*.

3.6 Defining Boundary Conditions

The solvers takes pressure, velocity, electric potential, volume fraction of fluid and other parameters which vary depending upon which viscosity model is used inside the 0 directory in the case files directory. The electric potential file is named *Ep*, which follows standard *OpenFOAM*® convention. All other files are used in other *rehoTool* solvers too and they are to be written in the default *OpenFOAM*® convention only.

3.7 Writing Results

The frequency of writing results can be controlled from *controlDict* file like in noraml *OpenFOAM*® cases. The solver writes pressure, velocity, electric potential, electric charge density (file named *rhoE*), maxwell stress tensor (file named *MST*), electric field (file named *Ef*), electric conductivity (file named *sig*), electric permitivitty (file named *eps*) and other fluid parameters based on the viscosity model selected.

Chapter 4

Test Case

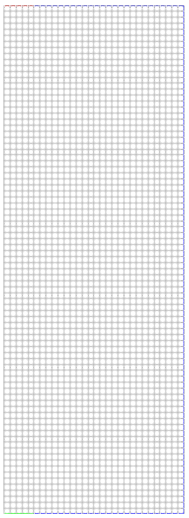
The solver *rheoInterEHDFoam* is a derivative of *rheoInterFoam* and only extra electric equations have been added, with the base solver still being the same. Thus, the solver will be only tested for working of electric equations as the viscosity models and two-phase solving capability is already tested and demonstrated for *rheoInterFoam* solver in Ref [4].

4.1 Geometry and Mesh

For testing the solver, an axis symmetric simulation of two water drops suspended in dielectric Oldroyd-B fluid subjected to electric potential is done. *OpenFOAM*® can work only with 3D geometries and as a result axis symmetric simulations use wedge shaped 3D geometry as seen in Fig 4.1. The grey coloured left edge is the axis, the red edge is upper electrode, the green edge is lower electrode and the blue edges are walls. The front and back face of the wedge is defined as a special *wedge* boundary condition in *OpenFOAM*® for axis symmetric simulations. The geometric parameters of the case can be viewed in Table 4.1.

Parameter	**Value**
Drop Radius	1mm
Electrode Radius	0.5mm
Outer Radius	3mm
Height	8.5mm

TABLE 4.1: Geometric Parameters



(A) Front View

(B) Top View

FIGURE 4.1: Mesh

4.2 Fluid Parameters

The water drops are assumed to be Newtonian. The properties of water and Oldroyd-B fluid are given in Table 4.2. The surface tension between the two fluids is 0.01 kg/s^2.

Fluid	**Property**	**Value**
Water	Density	1000 kg/m^3
	Viscosity	1×10^{-3} $Pa \cdot s$
	Electric Permitivitty	7.17189×10^{-10} F/m
	Electric Conductivity	1×10^{-4} $W/m \cdot K$
Oldroyd-B Fluid	Density	972 kg/m^3
	Solvent Viscosity	0.01 $Pa \cdot s$
	Polymeric Viscosity	0.03 $Pa \cdot s$
	Relaxation Time	0.02 s
	Electric Permitivitty	2.656×10^{-11} F/m
	Electric Conductivity	1×10^{-12} $W/m \cdot K$

TABLE 4.2: Properties of Fluids

4.3 Simulation Parameters

The fastest process occurring in this setup is charge displacement which is generally of the order of $10^{-6}s$ to $10^{-5}s$. Thus, the minimum time-step for the simulation was chosen to be $10^{-6}s$ with maximum allowable time-step set as $10^{-5}s$ to capture the charge displacement. The Table 4.3 enlists all the simulation parameters.

Parameter	Value
Minimum Time Step	1×10^{-6} s
Maximum Time Step	1×10^{-5} s
Solution Write Interval	0.001 s
Maximum Courant Number	0.2
Maximum Alpha Courant Number	0.2
Simulation Run Time	0.16 s

TABLE 4.3: Simulation Parameters

4.4 Boundary and Initial Conditions

4.4.1 Boundary Conditions

The boundary conditions applied in the test case are given in Table 4.4, where α_{water} is the volume fraction of water, ϕ is the electric potential, $\mathbf{u}$ is flow velocity and p is the pressure. The front and back faces are specified as *wedge* boundaries and the axis is specified as an *empty* boundary. For more information on these boundary types please refer the *OpenFOAM*® user manual [9].

4.4.2 Initial Conditions

The initial conditions applied in the whole domain are given in Table 4.5. The pressure can be initialized to any value of choice as it is only the relative pressure that matters for flow in an enclosed space. Adding or removing same amount of pressure in whole domain will not affect the other flow properties. Hence, here pressure has been initialized as 0 uniformly. The water drops are initialized by specifying $\alpha_{water} = 1$ in two spheres centred at the axis and seperated by a distance of $2.5mm$ symmetric about the axis. The initialized α_{water} field can be seen in Fig 4.2. The two red regions represent the two water droplets and the blue region is the Oldroyd-B fluid.

Boundary	Variable	Condition/Value
Upper Electrode	ϕ	1500 V
	u	No Slip
	p	Zero Gradient
	α_{water}	0
Lower Electrode	ϕ	-1500 V
	u	No Slip
	p	Zero Gradient
	α_{water}	0
Walls	ϕ	Zero Gradient
	u	No Slip
	p	Zero Gradient
	α_{water}	0

TABLE 4.4: Boundary Conditions

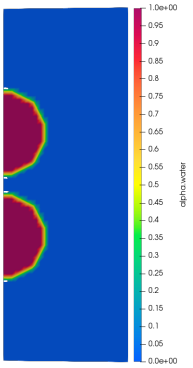


FIGURE 4.2: Initial α_{water} field

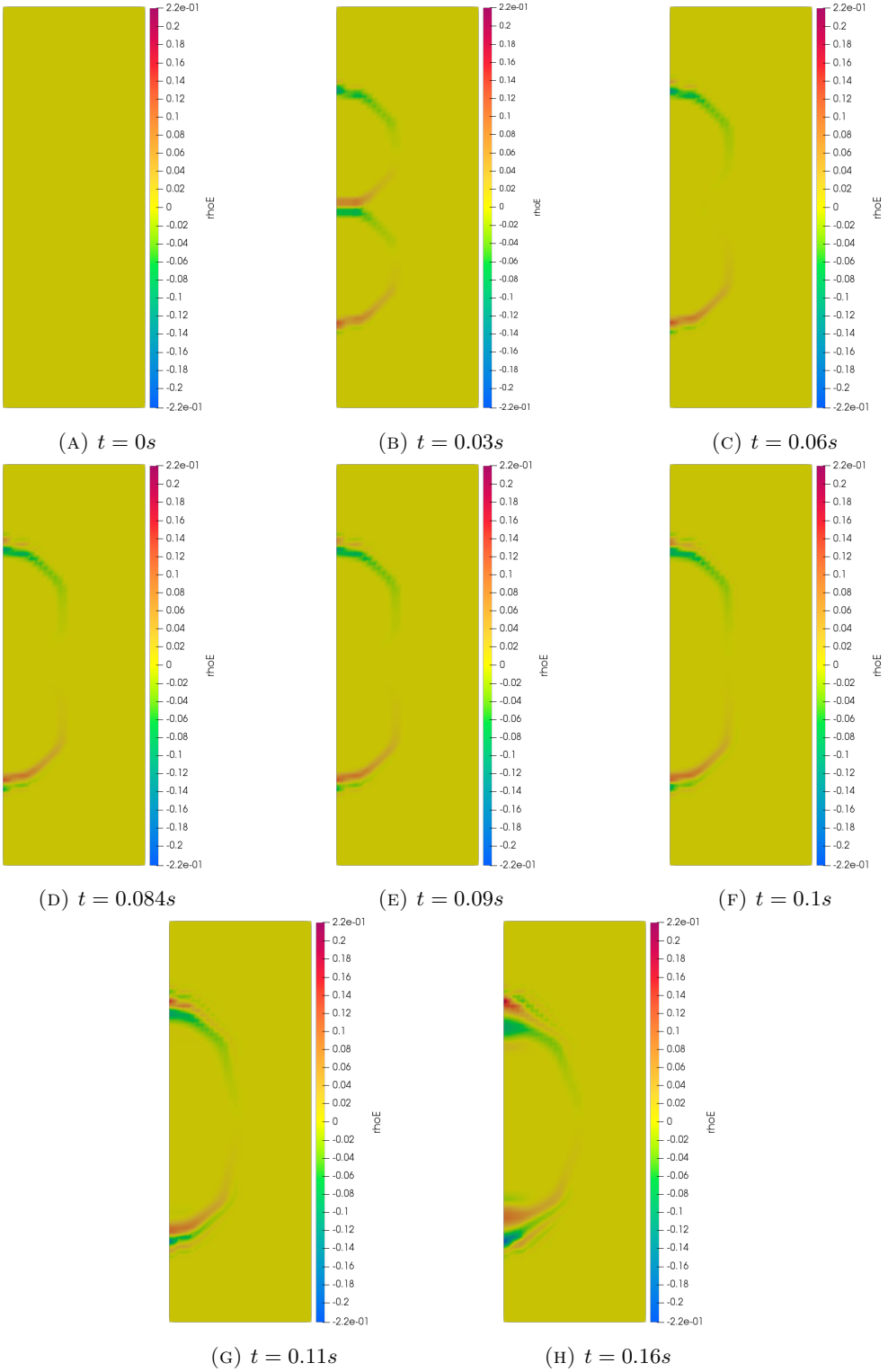
Variable	Initial Value
ϕ	0
u	(0,0,0)
p	0
α_{water}	0

TABLE 4.5: Initial Conditions

4.5 Simulation Results

As expected, the drops got polarized electrically, with positive charge accumulating towards negative electrode and negative charge accumulating towards positive electrode in both drops as seen in Fig 4.3. Both drops start moving towards each other because of the attractive force from oppositely charged surfaces and finally coalesce. The coalescing of drops can be scene in Fig 4.4. The coalescing starts at $0.084s$ as visible in Fig 4.4d. The fluid is almost stationary as visible in Fig 4.6h at $0.16s$ and thus, the process of coalescing can be assumed to be complete. Hence, a run time of $0.16s$ was used for the simulation.

The pressure plot given in Fig 4.5 gives the relative pressure values as explained in Sec 4.4.2.

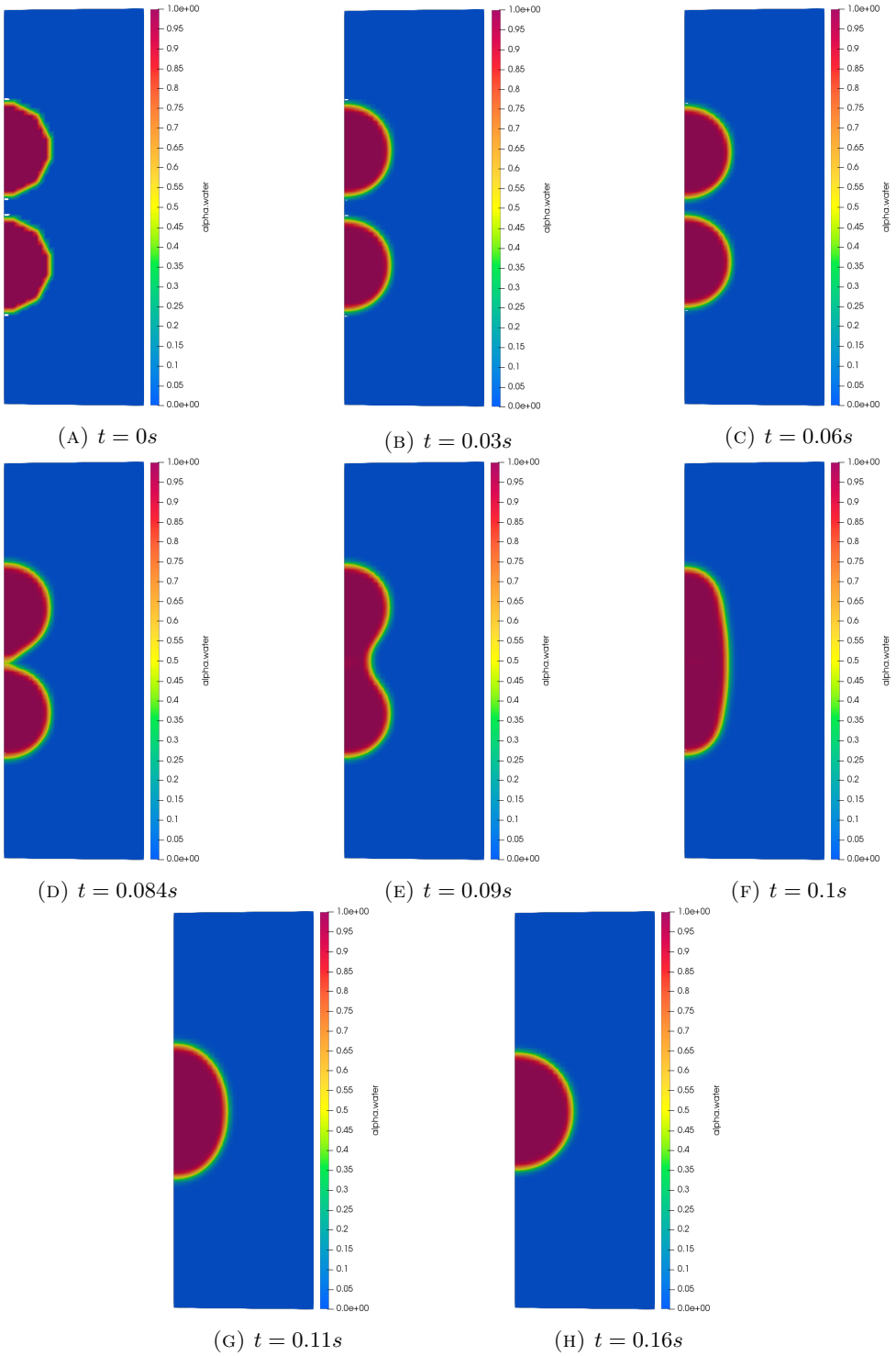


(A) $t = 0s$ (B) $t = 0.03s$ (C) $t = 0.06s$

(D) $t = 0.084s$ (E) $t = 0.09s$ (F) $t = 0.1s$

(G) $t = 0.11s$ (H) $t = 0.16s$

FIGURE 4.3: Electric Charge Density

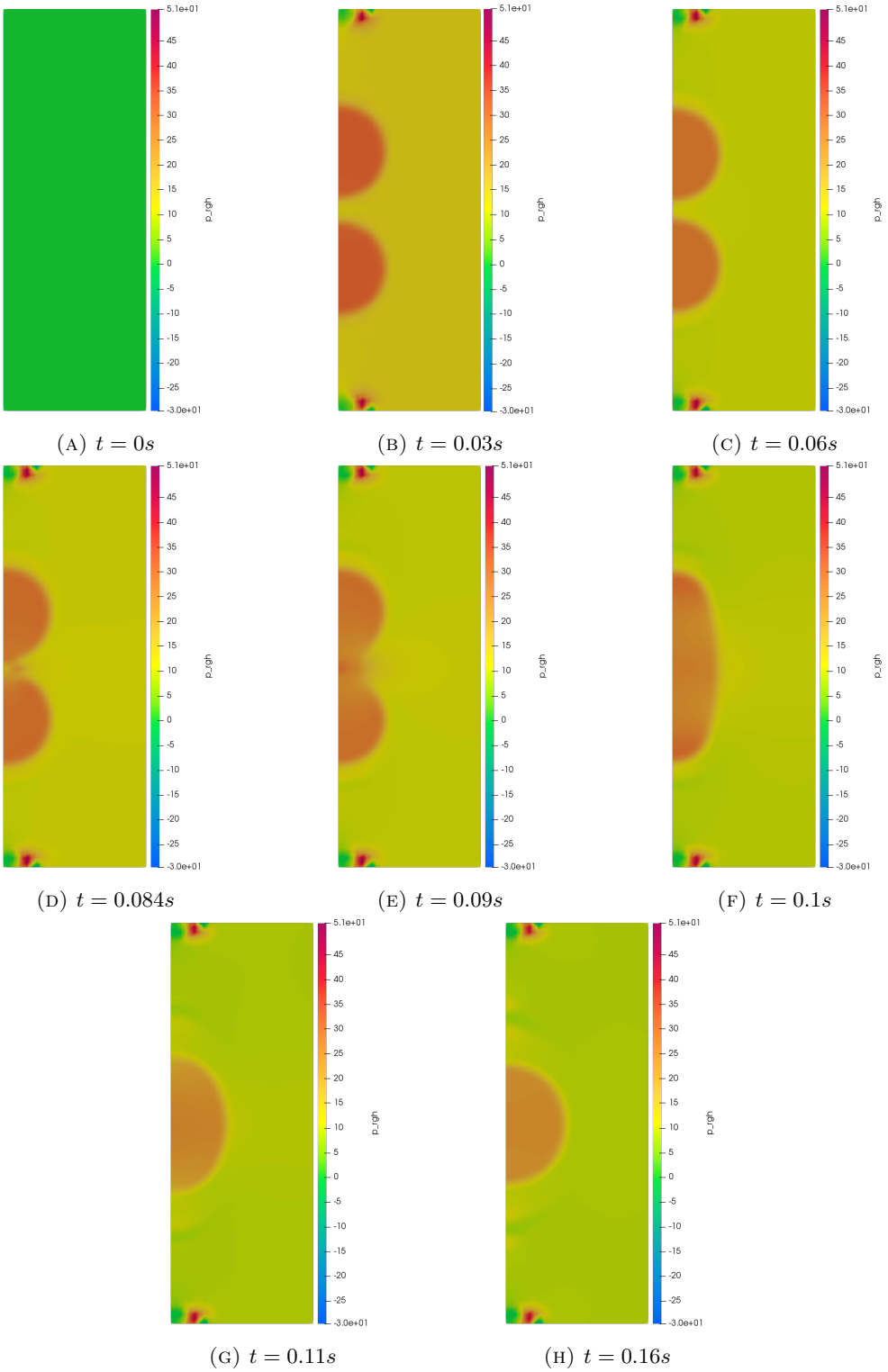


(A) $t = 0s$ (B) $t = 0.03s$ (C) $t = 0.06s$

(D) $t = 0.084s$ (E) $t = 0.09s$ (F) $t = 0.1s$

(G) $t = 0.11s$ (H) $t = 0.16s$

FIGURE 4.4: Coalescing Drops

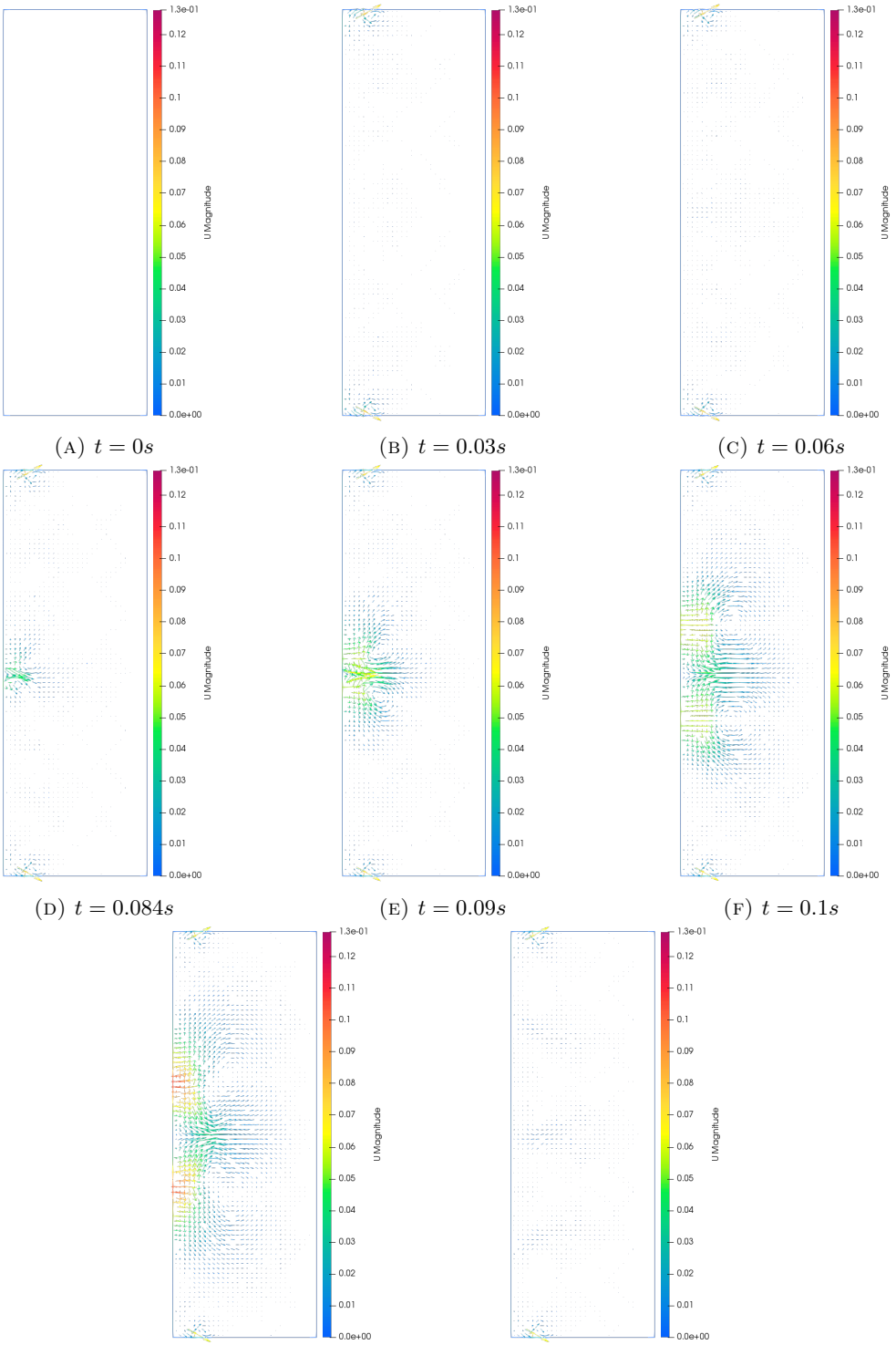


(A) $t = 0s$ (B) $t = 0.03s$ (C) $t = 0.06s$

(D) $t = 0.084s$ (E) $t = 0.09s$ (F) $t = 0.1s$

(G) $t = 0.11s$ (H) $t = 0.16s$

FIGURE 4.5: Pressure Field



(A) $t = 0s$ (B) $t = 0.03s$ (C) $t = 0.06s$

(D) $t = 0.084s$ (E) $t = 0.09s$ (F) $t = 0.1s$

(G) $t = 0.11s$ (H) $t = 0.16s$

FIGURE 4.6: Velocity Vector Plot

Chapter 5

Conclusion and Recommendations

The solver is working for the basic drop coalescing case as expected. The accumulation of charge is coherent with the electrostatic laws and the drops move towards each other an coalesce as they should. However, there is a lot of scope for future work. The solver's stability and accuracy have not been tested. The solver has been developed for *OpenFOAM v7* and the backward compatibility has not been checked. As of now, the solver can only solve two-phase equations, however a truly multi-phase capability can be added to the solver in future. The solver currently uses a complete electric model without much simplifications, which can take up a lot of computational power. For few cases, the electric equations can be simplified leading to computationally less expensive solver. Implementation of Ohmic (Leaky-Dielectric) Model, Slip Model, Debye-Hückel Model and others can be looked into in this regard.

Bibliography

[1] J. B. Fenn, M. Mann, C. K. Meng, S. F. Wong, and C. M. Whitehouse, "Electrospray ionization for mass spectrometry of large biomolecules," *Science*, vol. 246, no. 4926, pp. 64–71, 1989.

[2] J. S. Shrimpton and A. J. Yule, "Characterization of charged hydrocarbon sprays for application in combustion systems," *Experiments in Fluids*, vol. 26, no. 5, pp. 460–469, 1999.

[3] J. López-Herrera, S. Popinet, and M. Herrada, "A charge-conservative approach for simulating electrohydrodynamic two-phase flows using volume-of-fluid," *Journal of Computational Physics*, vol. 230, no. 5, pp. 1939–1955, 2011.

[4] F. Pimenta and M. Alves, "rheoTool." `https://github.com/fppimenta/rheoTool`, 2016.

[5] N. Tofighi, "interEHDFoam." `https://github.com/ntofighi/interEHDFoam`, 2017.

[6] F. M. et al, "openFoamEHD." `https://github.com/iroghair/openFoamEHD`, 2020.

[7] W. Aniszewski, T. Ménard, and M. Marek, "Volume of fluid (vof) type advection methods in two-phase flow: A comparative study," *Computers & Fluids*, vol. 97, pp. 52–73, 2014.

[8] D. A. Saville, "ELECTROHYDRODYNAMICS: The Taylor-Melcher Leaky Dielectric Model," *Annual Review of Fluid Mechanics*, vol. 29, no. 1, pp. 27–64, 1997.

[9] The OpenFOAM Foundation, "Openfoam v7 user guide." `https://cfd.direct/openfoam/user-guide-v7`.