Investigation of a Jacobian-free Newton-Krylov solution to multiphase flows

by

Amirali Ashrafizadeh

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2014

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The current study is focused on investigating a Jacobian-Free Newton-Krylov (JFNK) method to obtain a fully-implicit solution for two phase flows. In the JFNK formulation, the Jacobian matrix is not directly determined potentially leading to major computational savings compared to a simple Newton's solver. Prior to the implementation of JFNK to solve two-phase flow problem, it is utilized to solve the governing equations corresponding to single phase flow. The objectives of the present study are (i) Application of the JFNK method to two-fluid models, (ii) Investigation of the advantages and disadvantages of the method compared to commonly used explicit methods, and (iii) Comparison of the numerical predictions with those obtained by the current version of the Network thermal-hydraulics code, CATHENA. The background information required is presented and the numerical setup for each test case is discussed in detail. Three well-known benchmarks are considered, the 1D dam break problem, the water faucet and the oscillating manometer.
For single phase flow simulations, the Shallow Water Wave Equations is selected to model the motion of the fluid and a backward Euler scheme is utilized for the temporal discretization along with a central-upwind Godonuv scheme for the spatial discretization. For the two-phase simulations, an isentropic (four equation) two fluid model is chosen. Time discretization is performed by a backward Euler scheme and the AUSM+ scheme is applied to the convective fluxes. The source terms are discretized using a central differencing scheme. For comparison, one explicit and two implicit formulations, one with Newton's solver with the Jacobian matrix and one with JFNK, are implemented for each set of governing equations. A detailed grid and model parameter sensitivity analysis is performed to identify the advantages and disadvantages of JFNK for each case.
For all three benchmarks, the JFNK predictions are in good agreement with the analytical solutions and explicit profiles. Further, stable results can be achieved using high CFL (CourantFriedrichsLewy) numbers up to 100 with a suitable choice of JFNK parameters. The computational time is significantly reduced by JFNK compared to the calculations requiring the determination of the Jacobian matrix. This reduction is in the order of 80%.

Acknowledgements

First, I would like to thank my supervisor, Professor Cécile Devaud for the opportunity she gave me to work and study here at the University of Waterloo. The knowledge I have gained in CFD and numerical modeling is attributed to her.

I would also like to thank Professor Fue-Sang Lien and Professor Patricia Nieva for reviewing this thesis and providing valuable feedback. The financial support of AECL is gratefully acknowledged. Without their funding, this work would not have been possible.

Special thanks to my friend, Amirhossein (Mammal) for his patience and time in answering all of my programming questions.

My deepest thanks to my family back home in Iran for their love, care and support. Especially my father, Mahmud, for being a great friend and teacher in addition to being my dad.

Finally, I would like to express my sincere gratitude and thanks to my lovely wife, Sarvin. Without her love and support, going through these two years would have been impossible.

Dedication

To the love of my life, Sarvin.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

Nowadays, numerical simulations play a major role in engineering applications and are used in almost every engineering discipline. Particularly, in the nuclear industry, the development of robust and accurate simulation tools is crucial to the design and safe operation of nuclear reactors. In particular, multiphase flow modeling must be incorporated in thermal hydraulics codes. Several strategies exist ranging from the mixture model [76], volume of fluid model [30] and the Euler-Euler two fluid model (also known as Effective Fluid Modelling, EFM) [32]. In current nuclear safety analysis codes, such as CATHENA (Canadian Algorithm for THErmal hydraulic Network Analysis) [28], RELAP (Reactor Excursion and Leak Analysis Program) [15] and CATHARE (Code for Analysis of THermalhydraulics during an Accident of Reactor and safety Evaluation) [5], the two-fluid model is used and usually, for stability purposes, implicit solution techniques are chosen to solve the resulting system of equations.

The CATHENA thermal hydraulic code, is used for the analysis of postulated reactor upset conditions in CANDU (CANada Deuterium Uranium) reactors and is developed by Atomic Energy of Canada Limited (AECL). Currently, it is a one dimensional thermal hydraulics computer code which includes the non-equilibrium two fluid model as its thermal

hydraulics model. This model consists of three Partial Differential Equations (PDEs) for the conservation of mass, momentum and energy for each phase resulting in a total of six coupled equations. A fully implicit discretization of the governing equations is available in the new version, called CATHENA 4 [2, 3], with Newton's method to solve for all of the primitive variables simultaneously. In order to solve the non-linear set of equations using Newton's method, the Jacobian matrix is required to construct a linear system of equations. However, the explicit determination of the Jacobian matrix significantly increases the computational time and prevents further use of more elaborate discretization schemes.

Clearly, increasing the accuracy of the numerical predictions along with decreasing the computational time is a very desirable feature. Due to numerical issues related to the governing equations of two-phase flows (in general the system of equations are non-hyperbolic and cannot be written in conservative form), special care must be taken while trying to obtain and develop numerical schemes for such problems. Many studies have been conducted to finding more accurate numerical schemes for the solution to multiphase problems. Some past investigations have led to the development of the AUSM-family schemes for multiphase simulations which have been shown to give robust and accurate solutions for many different multiphase and thermal hydraulics problems [38]. However, almost all of the solution methods have been focused on explicit schemes and little attention has been placed on finding implicit solution techniques.

The Jacobian-free Newton-Krylov method (JFNK) [42] is an implicit solution technique based on Newton's method to find roots of functions while utilizing a Krylov solver to avoid the explicit formulation of the Jacobian matrix. By removing the necessity of calculating the Jacobian matrix, a large amount of computational time can be saved and more accurate discretization schemes may be used. The method involves multiple tuning parameters which require special care for the system of equations under consideration.

1.2 Objectives

In the present study, the application of the JFNK method to two-phase flows is of interest and is investigated. In order to acquire more insight to the behavior of the method, it is initially applied to a set of equations corresponding to single phase flows and afterwards, extended to two-phase flows. Many studies have been conducted to solve two-phase model problems, however implicit solutions have not been widely explored. In addition, the JFNK has never been used or applied for the solution to the two fluid model. The study is conducted per request of AECL, to help decrease computational time in the future versions of the CATHENA code. To summarize, the objectives of the current study are as follows:

- Application and implementation of JFNK to a system of equations corresponding to single phase flows. Through this, the behavior of the method is investigated and some tuning parameters are set for further use.
- Application and implementation of JFNK to a two-phase model and investigation of the pros and cons of the method.
- Comparison of the results obtained with previous numerical work and the results obtained from CATHENA.

The main expected features are also listed below:

- Implementation of a fully implicit discretization of the governing equations.
- Capability to use high order schemes for the discretization of the governing equations.
- Capability to use large time steps to advance in time through out the simulations.
- Capability to perform the simulation on fine grids.
- Production of stable results through time even when large time steps are used.

1.3 Outline

In Chapter 2 the basic governing equations of fluid dynamics are derived from principle conservation laws. The required background information necessary to follow the present thesis is described and covered. This includes an introduction to explicit and implicit schemes and the main differences between the two methods, Newton's algorithm to find the solution to non-linear functions and a brief description on Krylov solvers (GMRES in particular). The Jacobian free Newton-Krylov is presented in detail followed by a short introduction to CATHENA.

Chapter 3 begins with the derivation of the governing equations of two-phase flows. The local instant formulation of the equations are derived and discussed followed by introducing appropriate jump conditions at the interface between two phases. Averaging techniques utilized for two-phase flows are presented and some of the most commonly used two-phase models are described. The numerical difficulties encountered when trying to solve the system of equations are discussed and some of the previous studies regarding these issues are reviewed. Different numerical schemes used for the solution of multiphase problems are included as well.

As a first step towards the application of the JFNK method to two-phase problems, initially, the algorithm is applied to a simpler set of equations, the 1D shallow water wave equations (SWWE). Chapter 4 includes the full derivation and description of the SWWE system of equations along with the numerical discretization schemes investigated. Initial conditions and boundary conditions corresponding to the 1D dam-break problem are presented and results obtained from multiple simulations are presented and discussed.

Chapter 5 provides a detailed investigation of the application of the JFNK method to an isentropic two-phase system of equations. Full, in-depth detail on the numerical setup, including, the discretization of the numerical fluxes and source terms along with correction models for low Mach number flows and the appearance and disappearance of a phase are provided. The method is applied to two well known test cases, namely the water faucet and the oscillating manometer test cases. Multiple simulations are performed and details on the model parameters are given. The ability of the method to handle larger time-steps for each test case is investigated and the numerical predictions obtained from the algorithm

are compared the predictions obtained from explicit methods.

Finally, Chapter 6 includes the conclusions based on the present study. A summary of the work done and recommendations for future work are also provided.

Chapter 2

Background

Prior to the discussion on multiphase flows it is necessary to introduce some of the concepts and numerical tools used throughout the present thesis. First, the conservation laws of fluid flows are derived followed by a short introduction to explicit and implicit schemes and some of the advantages and disadvantages of each method are reviewed. Newton's method for finding roots to nonlinear equations is summarized followed by a discussion on the JFNK method. The GMRES sparse solver and CATHENA are also briefly introduced.

2.1 Governing Equations of Fluid Flow

In the current section, the governing equations of fluid flow are derived from differential control volumes. The conservation laws obtained are in general form without any simplifying assumptions.

2.1.1 Conservation of Mass

Consider a differential, fixed Control Volume (CV) within a fluid moving with the velocity field $\vec{V}$ (Eulerian CV). The x, y and z component of the velocity field are u, v and w, respectively. The mass of the fluid under consideration can be obtained from $m = \rho V =$

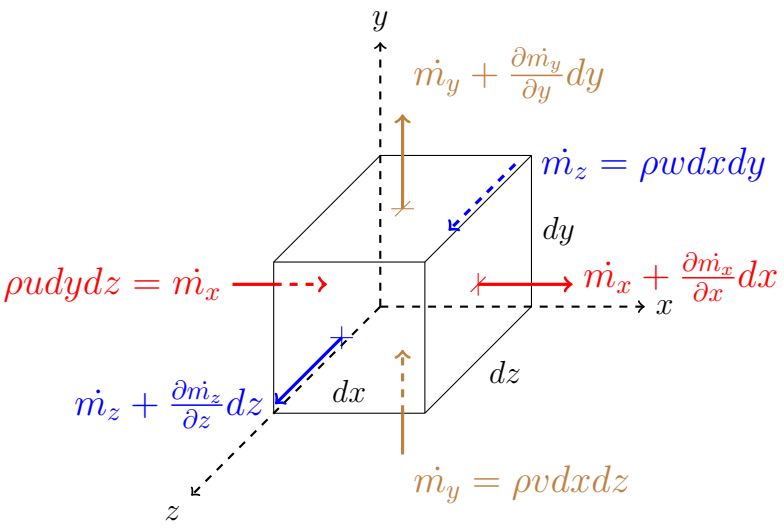


Figure 2.1: Differential control volume used for the derivation of the conservation of mass equation

$\rho(dxdydz)$, where ρ is the fluid density and dx, dy and dz are the lengths of the CV in the x, y and z direction respectively. Conservation of mass for this element simply states

$$\left\{ \begin{array}{c} \text{mass increase} \\ \text{within the CV} \end{array} \right\} = \left\{ \begin{array}{c} \text{net mass} \\ \text{flow in} \end{array} \right\} - \left\{ \begin{array}{c} \text{net mass} \\ \text{flow out} \end{array} \right\}. \tag{2.1}$$

Figure 2.1 illustrates a differential control volume within the fluid and also the mass fluxes passing through each face of the CV. After inserting the appropriate fluxes in Eq. 2.1 one can obtain

$$\frac{\partial m}{\partial t} = -\left[\dot{m}_x + \frac{\partial \dot{m}_x}{\partial x} dx + \dot{m}_y + \frac{\partial \dot{m}_y}{\partial y} dy + \dot{m}_z + \frac{\partial \dot{m}_z}{\partial z} dz\right],$$

which can be simplified using the gradient operator, $\vec{\nabla}$, and written as the continuity equation

$$\boxed{\frac{\partial \rho}{\partial t} + \vec{\nabla}.(\rho \vec{V}) = 0}. \tag{2.2}$$

2.1.2 Conservation of Linear Momentum

In order to derive the Navier-Stokes equation, instead of focusing on a fixed control volume, a moving CV is considered which can deform due to surface and body forces acting on it. The mass contained inside the control volume remains constant (Lagrangian approach). From Newton's second law:

$$\vec{F} = m\vec{a},$$

which for a fluid element can be written as

$$\rho\frac{D\vec{V}}{Dt} = \vec{B} + \vec{S}. \tag{2.3}$$

In Eq. 2.3, $\frac{D}{Dt}$ is the material derivative [20] and $\vec{B}$ and $\vec{S}$ are the body and surface forces per unit volume acting on the CV respectively. Gravity is the only body force that is considered in the present derivation. Therefore, $\vec{B} = \rho\vec{g}$. Figure 2.2 illustrates the surface forces acting on the CV (only the x - direction is shown for clarity). Note that pressure is included in the normal stresses (τ_{xx} in Fig. 2.2). Applying the forces shown into Eq. 2.3, one can obtain (in tensor notation)

$$\rho\frac{Dv_i}{Dt} = \rho g_i + \frac{\partial \tau_{ij}}{\partial x_j}. \tag{2.4}$$

τ_{ij} contains both pressure forces and viscous forces and for a Newtonian fluid, it can be written as

$$\tau_{ij} = -p\delta_{ij} + \underbrace{\left[(\lambda\vec{\nabla}.\vec{v})\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right]}_{\text{viscous part} = \tau^v_{ij}} = -p\delta_{ij} + \tau^v_{ij}. \tag{2.5}$$

In Eq. 2.5, p is the pressure acting on the surface of the CV, λ is the bulk viscosity [60], μ is the viscosity and δ_{ij} is the Kronecker delta. λ can be obtained by Stoke's hypothesis

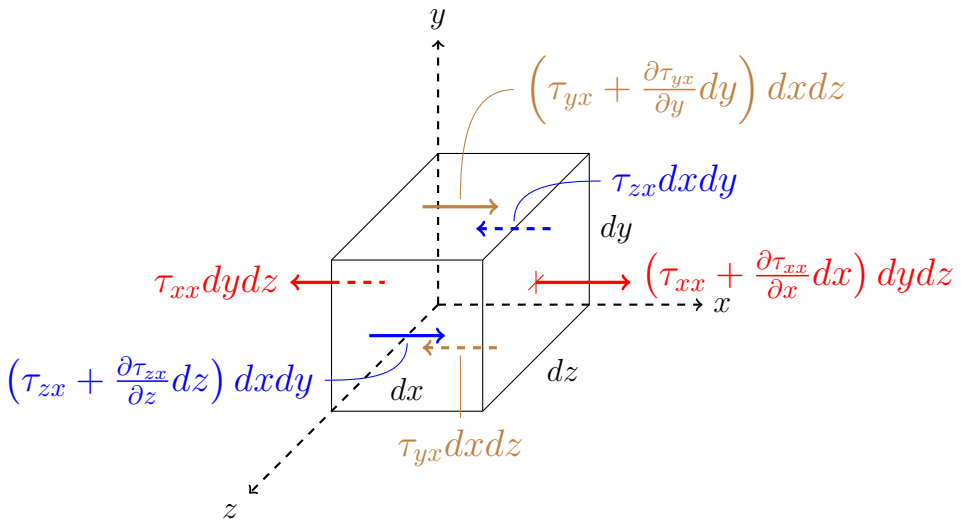


Figure 2.2: Surface forces in the x-direction acting on a CV

which applies for most fluids, $\lambda + \frac{2}{3}\mu = 0$. Inserting Eq. 2.5 into Eq. 2.4 and generalizing for all three directions, the conservation of linear momentum can be written as

$$\boxed{\rho \frac{D\vec{V}}{Dt} = -\vec{\nabla} p + \vec{\nabla}.\tau_{ij}^{v} + \vec{B}}. \tag{2.6}$$

Equation 2.6 is the general form of the conservation of linear momentum law. For many practical conditions, the viscous forces may be neglected (inviscid fluid). By doing so, Euler's conservation of momentum equation is obtained, which states

$$\rho \frac{D\vec{V}}{Dt} = \rho \vec{g} - \vec{\nabla} p. \tag{2.7}$$

2.1.3 Conservation of Energy

As described in Sec. 2.1.2, a CV of fixed mass moving with the fluid is considered. Applying the first law of thermodynamics (conservation of energy) to this control volume gives

$$\frac{DE}{Dt} = \frac{DQ}{Dt} + \frac{DW}{Dt}, \tag{2.8}$$

where the left hand side is the rate of change in total energy. The first term on the right hand side is the rate of heat transfer to the CV and the second term is the rate of work done on the CV by surface and body forces. The total energy of the fluid can be written as the sum of the internal energy and the kinetic energy of the fluid. This is given by

$$\frac{DE}{Dt} = \frac{D}{DT}\left[\underbrace{(\rho V)}_{=m}\left(I + \frac{1}{2}V^2\right)\right] = \rho V \frac{D}{DT}\left[I + \frac{V^2}{2}\right]. \tag{2.9}$$

In Eq. 2.9, I is the internal energy per unit mass and $\frac{V^2}{2}$ is the kinetic energy per unit mass of the fluid. Figure 2.3 shows the heat transfer fluxes applied to the CV under consideration. q_k is the heat flux in the k direction and q''' is the internal heat generation per unit volume. Examining Fig. 2.3 and applying Fourier's law, it can easily be shown that:

$$\frac{DQ'''}{Dt} = -\frac{\partial q_i}{\partial x_i} + q''' = -\vec{\nabla}.\vec{q} + q''' = \vec{\nabla}.\left(k\vec{\nabla}T\right) + q''', \tag{2.10}$$

where k is the thermal conductivity of the fluid. The final term which needs to be calculated is the rate of work done on the CV. The general form is given by $\frac{DW}{Dt} = \left(\vec{\text{Force}}\right).\left(\vec{\text{Velocity}}\right)$. Considering gravity as the only body force acting on the CV, the rate of work done by body forces can easily be calculated. In order to evaluate the rate of work done by the surface forces, consider the surface forces shown in Fig. 2.2. Work done on the CV is considered to be positive if the velocity and the force are in the same

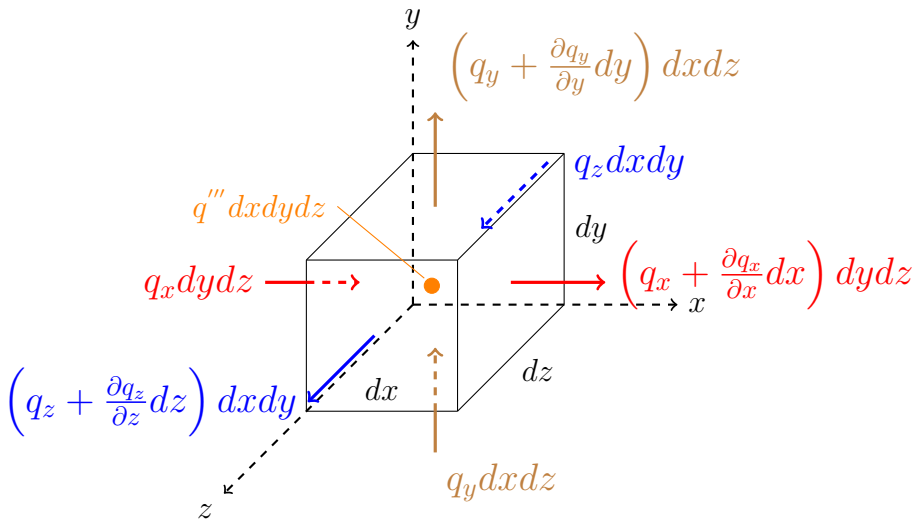


Figure 2.3: Heat fluxes entering and leaving differential control volume

direction and is assumed to be negative if they are in opposite directions. Keeping this in mind, it can shown that

$$\left(\frac{DW'''}{Dt}\right)_{\tau} = \vec{V}.\left(\vec{\nabla}.\tau_{ij}\right) + \tau_{ij}\frac{\partial u_i}{\partial x_j}. \tag{2.11}$$

Combining all terms (Eqs. 2.9, 2.10 and 2.11), the general form of the energy equation is derived

$$\rho\frac{D}{Dt}\left(I + \frac{V^2}{2}\right) = \left[\vec{\nabla}.\left(k\vec{\nabla}T\right) + q'''\right] + \left[\vec{V}.\left(\nabla.\tau_{ij}\right) + \tau_{ij}\frac{\partial u_i}{\partial x_j} + \rho\vec{g}.\vec{V}\right]. \tag{2.12}$$

Eq. 2.12 relates internal energy and kinetic energy to the heat transfer and work done on the CV. It is desirable to eliminate kinetic energy and write the internal energy as a function of the primary variable, temperature. By taking $\vec{V}$.(Eq. 2.12) and using the conservation of momentum and mass (Eqs. 2.2 and 2.6), and also some thermodynamic

relations, the conservation of energy can be written as

$$\boxed{\rho C_p \frac{DT}{Dt} = -\vec{\nabla}.\vec{q} + q''' + \beta T \frac{Dp}{Dt} + \mu\Phi}. \tag{2.13}$$

where, β is the coefficient of thermal expansion and $\mu\Phi$ is called viscous dissipation and is the irreversible work done by viscous forces. More explicitly Eq. 2.13 is

$$\underbrace{\rho C_p \frac{\partial T}{\partial t}}_{\text{Transient term}} + \underbrace{\rho C_p \vec{V}.\vec{\nabla}T}_{\text{convection term}} = \underbrace{\vec{\nabla}.(k\vec{\nabla}T)}_{\text{conduction term}} + \underbrace{\beta T \frac{Dp}{Dt}}_{\text{pressure term}} + \underbrace{\mu\Phi}_{\text{viscous dissipation}} + \underbrace{q'''}_{\text{heat generation}} . \tag{2.14}$$

2.2 Explicit and Implicit Methods

In numerical analysis the terms explicit and implicit are frequently encountered. When an ordinary or partial differential equation is intended to be solved numerically, the equation needs to be discretized on a computational domain. The difference between the two methods, lays in the time level used for the variables in the spatial discretization. If the variables used are from the current time level (known state), the method is explicit and if unknown values from the next time step are used, an implicit discretization is constructed. Suppose the numerical solution to the 1 dimensional (1D) heat equation (Eq. 2.15) is sought,

$$\frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2}. \tag{2.15}$$

Equation 2.16a illustrates an explicit discretization of the governing equation and Eq. 2.16b shows a fully implicit version of the discretization. The subscripts denote the spatial location of the node under consideration and the super scripts indicate the time level of the variable (n being the current time). As can be seen, the only difference is in the time level used for the terms used for the spatial discretization.

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = k \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2}, \tag{2.16a}$$

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = k\frac{T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}}{\Delta x^2}. \tag{2.16b}$$

After rearranging Eq. 2.16a, it is observed that in order to update the temperature at each node, a simple algebraic equation needs to be solved. This is not the case with an implicit scheme. Each equation is involved with multiple unknowns and a system of equations needs to be solved. By solving the resulting system, all unknowns are updated simultaneously.

Using an explicit scheme results in a very efficient and computationally inexpensive update of the different flow parameters. It is also very easy to program and allows the simple implementation of parallel methods. The main drawback to the method is stability issues that occur when large time steps are used. The time step which can be used for explicit methods is highly restricted. Implicit methods do not have this restriction and are capable of handling relatively large time steps compared to explicit schemes and are considered to be unconditionally stable (although in practice implicit schemes are also limited to a certain size of time step chosen). Implicit methods are also much harder to program and require large system of equations to be solved at each time step which leads to a major increase in computational time, per iteration.

2.3 Newton's Method

Newton's method is implemented to solve non-linear equations by finding approximate roots of functions. It can be used for both scalar and vector functions [36]. First, the method is described for scalar functions, then the description will be expanded to vectors. For a scalar function, the solution to satisfy the equation $f(x) = 0$ is required. Newton's method can be summarized in the following steps

(i) Start with an initial guess, x_0

(ii) Calculate $f(x_0)$ and $f'(x_0)$, where f' is the derivative of the function f

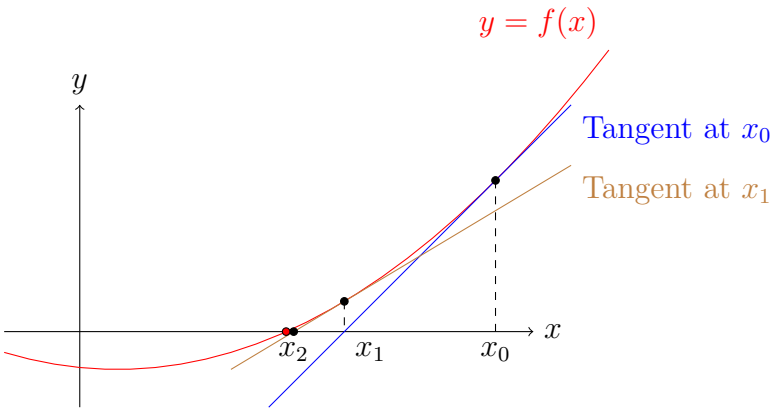


Figure 2.4: Newton's method for a scalar function

(iii) A better estimate of the root, x_1, can be calculated using the following formula

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}. \tag{2.17}$$

Steps i, ii and iii are repeated until the solution is found within a specified tolerance (stop criterion). The iterations start with an initial guess (x_0) and as the initial value is updated ($x_1, x_2, etc. \ldots$) the solution gets closer to the root of the function. It can be shown that the convergence of Newton's method (if the initial guess is close enough to the root) is quadratic [36]. This means that the number of correct decimals approximately doubles at each iteration. These steps are shown in Fig. 2.4.

Newton's method can be extended to vector functions. The vector function may be written as $F(u) = 0$ for which the roots need to be determined. The same methodology is used for vector functions. The state vector, u, can be compared to x and $F(u)$ can be compared to $f(x)$ in Eq. 2.17. In order to compute $F'(u)$ (first derivative), a Taylor series expansion of the function is applied such as

$$F\left(u^{k+1}\right) = F\left(u^{k}\right) + F'\left(u^{k}\right)\left(u^{k+1} - u^{k}\right) + HOT. \tag{2.18}$$

In Eq. 2.18, the superscript k refers to the previous guess of the root and $k+1$ corresponds to the current guess. Neglecting the Higher Order Terms (HOT), and setting $F\left(u^{k+1}\right)$

equal to the zero matrix (this occurs at convergence), a set of linear equations in the form of $Ax = b$ can be written as

$$J\left(u^k\right)\delta u^k = -F\left(u^k\right); \quad \delta u^k = u^{k+1} - u^k. \tag{2.19}$$

By solving Eq. 2.19, the state vector u^{k+1} can be calculated and it serves as the next guess in Newton's method. J is the Jacobian matrix of function F and is defined as

$$J_{(i,j)}\left(u^k\right) = \frac{\partial f_i\left(u^k\right)}{\partial u_j}. \tag{2.20}$$

The subscripts i and j refer to the i^{th} row and the j^{th} column of the Jacobian matrix. This procedure is performed until the desired precision is achieved. Following common practice [44], the stop criterion used in the present study is

$$\frac{\|\partial u^k\|}{\|u^k\|} < \tau_{newt}, \tag{2.21}$$

where τ_{newt} is the tolerance set for Newton's method.

2.4 The Generalised Minimal Residual Method (GMRES)

For linear system of equations, two main classes of iterative methods can be found [70]: i) Stationary methods such as Jacobi, Gauss-Seidel and Successive Over Relaxation (SOR) and ii) Krylov subspace methods. In the present study, a solver from the Krylov subspace techniques is selected to solve large linear systems of the form $Ax = b$ iteratively. These methods use the Krylov subspace, K_j, defined as

$$K_j = span\left(r_0, Ar_0, A^2r_0, \ldots\ldots, A^{j-1}r_0\right). \tag{2.22}$$

In Eq. 2.22, the residual is calculated as $r_0 = b - Ax_0$. A significant advantage of Krylov methods over other iterative methods, is that instead of using individual elements of A, only matrix-vector products are required. This feature is the key to saving computational time in Newton's method [42] and is further discussed in Sec. 2.5. Among Krylov methods, the conjugate gradient, generalized minimum residual (GMRES) and the bi-conjugate gradient methods are widely used [14]. The present study is focused on the GMRES method. The generalized minimum residual method is a projection method based on taking $K = K_m$ and $L = AK_m$, in which K_m is the m-th Krylov subspace with $\nu_1 = r_0/\|r_0\|$. Such a technique minimizes the residual norm over all vectors in $x_0 + K_m$ [71, 70]. Further information can be found in many references, for example [11, 13]. In the current work, GMRES is implemented as a linear solver for a large sparse matrix of the form $Ax = b$ iteratively.

2.4.1 GMRES Performance

The performance of GMRES is proportional to the number of computations performed by the algorithm. There are two main ways to modify the total performance. First, the number of computations performed in a single GMRES iteration may be reduced leading to a reduction of total computations. The second method is to modify the inputs to the GMRES solver (A matrix). This will cause a change in the behavior of the system for convergence. If the system converges more easily, the number of iterations required will decrease, leading to an improvement of the GMRES solver.

Restart Parameter

The computational costs of GMRES increase exponentially at higher GMRES iterations [70, 71]. Therefore the GMRES algorithm becomes impractical when the number of iterations required is large due to the growth of computational and memory requirements. One method to solve this issue is to restart the algorithm periodically. If a restart parameter is defined, the algorithm limits the number of iterations performed to the value of the restart parameter and only saves the last residual. This residual is then used to construct a new Krylov subspace where the algorithm starts again. This is referred to as GMRES(m) or

restarted GMRES.

Pre-conditioning

The main objective of using a pre-conditioner is to input an easier system to solve or a more suited system, to the solver. To this end, matrix multiplications are used to multiply a pre-conditioner (which is a matrix) with the original system

$$M^{-1}Ax = M^{-1}b. \tag{2.23}$$

Matrix M is called the pre-conditioner. A pre-conditioner should have two main properties. It should be cheap to construct and apply, and the pre-conditioned system should be easier to solve [7]. There are two main classes of pre-conditioners, the first class is a case specific pre-conditioner that requires complete knowledge of the physics of the problem. This is referred to as physics based pre-conditioning [42]. The second class is a more general technique that is universally applicable. Although this technique is not the most efficient method for all problems, it is used to achieve reasonable efficiency on a wide range of problems [7].

2.5 Jacobian-free Newton-Krylov Method (JFNK)

As seen in Eq. 2.19, in order for the Newton iteration to advance, the Jacobian matrix, which is composed of the first partial derivatives of the vector function defining the system with respect to all of the state vector unknowns, as shown in Eq. 2.20, is required. In some cases, calculating the Jacobian matrix is difficult or computationally intensive. Further, as the system gets larger, the Jacobian matrix becomes more expensive to calculate and store. In the GMRES algorithm, it can be seen that the Jacobian matrix is only needed in a matrix-vector product form. Therefore, it is possible to approximate the Jacobian-vector product without calculating the Jacobian matrix explicitly. Such a procedure will decrease the memory needed and the number of computations required. A Taylor series expansion

of the zero-sum vector function ($F(u) = 0$) is used to approximate the Jacobian-vector product (any arbitrary vector) by

$$Jv = \frac{F(u + \epsilon v) - F(u)}{\epsilon}, \tag{2.24}$$

where ϵ is a perturbation parameter and v is an arbitrary vector. Eq. 2.24 corresponds to a first-order Taylor expansion approximation to the Jacobian matrix multiplied by an arbitrary vector. Inside the GMRES function, wherever the Jacobian matrix is multiplied by a vector, the right hand side of Eq. 2.24 is kept and the Jacobian matrix is not calculated explicitly. It is clear that the choice of the value for ϵ will affect the precision of the approximation. A large value will lead to a poor approximation of the Jacobian-vector product and a very small value will lead to round-off errors. It has been shown that a value close to the square root of the machine precision error (the smallest distinguishable difference between two numbers on the computer) is able to approximate the Jacobian matrix in many cases [44]. Other methods to determine the perturbation value are also presented in [42, 12]. In the present study, a 64-bit version of MATLAB is used where the machine precision error, $\epsilon_{machine}$, is approximately equal to 2×10^{-16}. The results presented in Chapters 4 and 5 are obtained by using $\epsilon = 1.4 \times 10^{-7}$ unless otherwise stated. Note that this value is an order of magnitude larger than square root of the machine epsilon.

2.5.1 Advantages and disadvantages of the JFNK method

One of the main advantages of the JFNK method is that for a small convergence tolerance value, there is no splitting or linearization error [41]. Another advantage of the method is the simple implementation of different discretization schemes and the addition of governing equations. Different time discretization schemes can also be implemented without major modifications in the code. This is due to the fact that unlike other methods, such as semi-implicit methods, only pure algebraic presentations are required and there is no need to use values of the unknowns at different time levels in the PDEs [43]. Also, due to the implicit nature of the method, the solutions are more stable and the selected time step is not as restricted to the CFL number of the problem considered. This will allow higher

CFL values, resulting in fewer time steps needed to reach the solution.
The major disadvantage of the method is that a nonlinear problem is being solved iteratively. In order to advance in time a large matrix system needs to be solved at each Newton iteration. In general, this is computationally expensive and in order for the method to be efficient, effective pre-conditioners are required. Another approach to decrease the computations required in the JFNK method is the use of restarted GMRES [71]. In the present study, pre-conditioners or restart parameters are not investigated. Instead, special attention is given to the ability of the Jacobian-free Newton-Krylov method to produce stable and accurate results at higher CFL numbers.

2.6 The Canadian Algorithm for Thermal Hydraulic Network Analysis (CATHENA)

CATHENA 4, is a one dimensional two-phase thermal hydraulics code which is currently under development and is used by AECL for reactor design and safety analysis. Currently, CATHENA 4 uses a first order upwind scheme for the discretization of the governing equations (which are modeled by the six equation two-fluid model, as discussed in Chap. 3) and uses Newton's method to solve the system of equations. The code is fully implicit but, the Jacobian matrix is explicitly calculated for each time step. The solution to Eq. 2.19 is then solved by either the PARDISO [73] or the SMPAK solver. CATHENA uses a staggered grid meaning that scalar variables (such as the void fraction and pressure) are stored at node centers and vector variables (fluid velocities) are stored at links connecting the nodes. It is the final goal of the the present project to implement the JFNK method into CATHENA to reduce the computational time required for its simulations.

Chapter 3

Two-Phase Flow Modeling

In the present chapter the fundamentals of two-phase flow are described. The governing equations of two-phase flows are derived along with the appropriate jump conditions. Averaging techniques used to model two phase flow are also discussed. Further, two of the most commonly used two-phase models are reviewed followed by a mathematical analysis of the resulting system of equations. Finally, previous numerical methods used for the solution to two-phase problems are mentioned.

3.1 Governing Equations of Two-Phase Flow

In the current section, the governing equations of two-phase flows are derived. Jump conditions are also necessary at the interfaces and are further discussed. The derived system of equations is then averaged in the following section (Sec. 3.2). Some mathematical tools are required in order to derive the governing equations. Namely, the Reynolds transport theorem

$$\frac{d}{dt}\int_V F_k dV = \int_V \frac{\partial F_k}{\partial t} dV + \oint_A F_k V_k . n dA, \tag{3.1}$$

and Green's theorem

$$\int_V \nabla . F_k dV = \oint_A n . F_k dA. \tag{3.2}$$

In Eqs. 3.2 and 3.1, V is the volume of a control volume and A is the surface of that control volume.

3.1.1 Instantaneous Formulation

In the derivation of the governing equations for two-phase flows, the flow is considered as two single phase regions which are divided by a moving boundary, the interface. The standard governing equations derived in Chapter 2 hold for each one of these regions up to the interface. Appropriate jump conditions need to be considered at each interface to take into account the sudden change in certain flow variables such as fluid density, velocity and energy. The *local instant formulation* is given by considering the local instant variables, $F = F(x, t)$. The derivation begins by considering the general integral balance of a quantity, ψ, over a material volume, V. The general integral balance states

$$\frac{d}{dt} \int_{V_m} \rho_k \psi_k dV = - \oint_{A_m} n_k . \mathcal{J}_k dA + \int_{V_m} \rho_k \phi_k dV, \tag{3.3}$$

where, $\mathcal{J}_k$ is the influx of ψ_k through A and ϕ_k is the body source of ψ_k within V. The subscript k, denotes the phase. Physically, Eq. 3.3 implies that the rate of change of ψ_k is equal to the net amount of ψ_k leaving the control volume and the amount generated within the cell boundaries. Using the Reynolds transport theory (shown in Eq. 3.1) and Green's theory (shown in Eq. 3.2), which relates volume and surface integrals, the general differential balance equation can be obtained as

$$\frac{\partial \rho_k \psi_k}{\partial t} + \nabla . (V_k \rho_k \psi_k) = -\nabla . \mathcal{J}_k + \rho_k \phi_k. \tag{3.4}$$

It can be observed, that with the correct values of ψ_k, ϕ_k and $\mathcal{J}_k$, the resulting equations are in agreement with those obtained in Chapter 2 (Eqs. 2.2, 2.6 and 2.13). These values

Table 3.1: Parameters for general balance equations

Balance	ψ_k	$\mathcal{J}_k$	ϕ_k
Mass	1	0	0
Momentum	V_k	$-\mathbb{T}_k = p_k\mathbb{I} - \tau_k$	g_k
Energy	$u_k + \frac{V^2}{2}$	$q_k - \mathbb{T}_k.V_k$	$g_k.V_k + \frac{\dot{q}_k}{\rho_k}$

are shown in Table 3.1. In Table 3.1, $\mathbb{T}_k$ is the surface stress tensor for phase k and $\mathbb{I}$ is the unit tensor. As explained in Sec. 2.1.2, the stress tensor is divided into a pressure term and a viscous stress term. In addition, u_k, q_k and $\dot{q}_k$ represent the internal energy, heat flux and the heat generation, respectively. To complete the local instant formulation of two phase flows, jump conditions need to be specified for the interfaces. This is described in the following section.

3.1.2 Jump Conditions

The general balance equations (Eq. 3.4 and values from Table 3.1) can be applied to each phase but cannot be used at the interface between the two phases. Instead, a new set of governing of equations should be developed to take into account the singularities which may occur at the interface. These are called *jump conditions* and are unique for multiphase analysis.

In Fig. 3.1 the surface A_I splits the control volume into two regions and is a singular surface with respect to ψ and the fluid velocity, u. The surface may be in motion with a velocity of u_I. The general balance equations shown in Eq. 3.3, hold for this control volume. Now, the transport theorem needs to be modified for a region containing a surface separating a discontinuity. The transport theory is obtained from

$$\frac{d}{dt}\int_V \psi dV = \int_V \frac{\partial \psi}{\partial t} dV + \int_A \psi u_k.ndA - \int_{A_I} [\psi] u_I.ndA, \tag{3.5}$$

where, $[\psi]$ is the defined as $[\psi] = \psi^+ - \psi^-$, which is the difference between the limiting values of ψ on both sides of the singular surface. Again, Eq. 3.3 can be applied to any

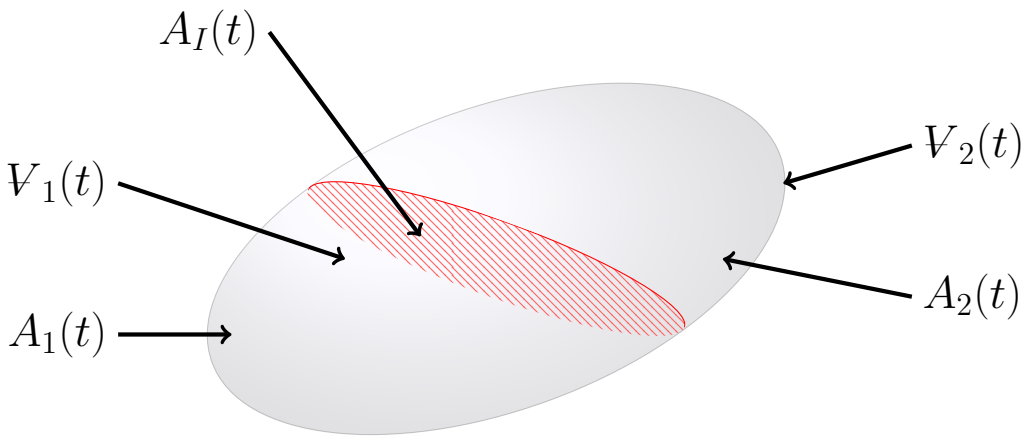


Figure 3.1: Control Volume containing singular surface

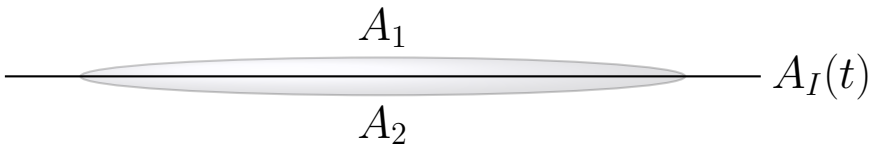


Figure 3.2: Shrunk Control volume for interface

volume regardless of the fact if it contains a singular surface or not. Combining Eqs. 3.3 and 3.5 and shrinking the size of A_1 and A_2 to reach A_I (shown in Fig. 3.2), a new set of governing equations are obtained. Since it is assumed that the the control volume shrinks to the interface surface, all volume integrals may be ignored and the resulting equation is

$$\int_{A_I} ([\rho\psi u.n] - [\rho\psi]u_I.n + [\mathcal{J}.n])dA = 0. \tag{3.6}$$

Due to the fact that Eq. 3.6 must hold for any arbitrary surface (including the interface separating two phases), the integrand must be zero. Therefore, for two-phase flows

$$\sum_{k=1,2} = (\dot{m}_k\psi_k + n_k.\mathcal{J}_k) = 0, \tag{3.7}$$

where, $\dot{m}_k$ is the mass transfer per unit time passing through the interface and can obtained

from $\dot{m}_k = \rho_k \left(u_k - u_I\right).n_k$. The governing equation shown in Eq. 3.7 is the general jump condition for two-phase flows and values shown in Table 3.1 can be used in order to obtain the jump condition for each of the conservation laws. The complete derivation of the conservation laws for multiphase flows can be found in many text books such as [21], [32] and [40]. The reader is encouraged to consult with these references for a more detailed derivation.

3.2 Averaging Techniques

The local instant formulation equations obtained in the Sec. 3.1, are not very useful for most applications due to the fact that the location of the interfaces are unknown at each time. Therefore the governing equations need to be averaged. For most engineering problems, mean values of fluid properties are sufficient, which eliminate the need for the local instant formulation, and instead, averaged equations are used. After averaging, the resulting governing equations are greatly simplified at the cost of loosing information due to the averaging process (similar to what is encountered in single phase turbulent flows). Three main types of averaging techniques can be performed [32], Lagrangian averaging, Eulerian averaging and the Boltzmann statistical averaging.

The most popular class of averaging is the Eulerian averaging method because it is the closest method to experimental observations. Here, time and space are considered as independent variables and changes to all other variables are expressed with respect to these two independent variables. In addition, the Eulerian method basically consists of integral operators and has a smoothing effect within a domain of integration [32]. The Lagrangian method is mostly used when the dynamics of a particular particle is of interest. Furthermore, the Boltzmann statistical method is considered when the behavior of a group of particles is the main focus of investigation. In the following section the Eulerian averaging will be considered. A more detailed description of averaging techniques for multiphase flow problems can be found in [19] and [32].

3.2.1 Instantaneous Volume-Averaged Equations

The local instant formulation can be averaged over area or volume. Here, the volume based averaging method is used due to the fact that singularities may appear in area averaging whereas they do not appear in volume based averaging methods. Consider the fixed tube shown in Fig. 3.3. The arbitrary volume of V^* is intersected by two cross sectional planes. The two planes are located a distance Z apart and the intersection of the two with volume V^* are A_{1k} and A_{2k}. V_k is the volume enclosed by A_{1k}, A_{2k} and the portion of A_I and A_{wk} between the two cross sectional planes. The cross sectional planes limiting the volume V_k are allowed to move and their speeds are $(-u_{A11}.n_z)$ and $(u_{A12}.n_z)$, where, n_k is the unit vector normal to the interface and pointed away from phase k.

Applying Eq. 3.1 to the volume V_k, leads to

$$\begin{aligned}\frac{\partial}{\partial t}\int_{V_{k(z,t)}} f(x,y,z,t)dV &= \int_{V_{k(z,t)}} \frac{\partial f}{\partial t}dV \\ &+ \int_{A_I(z,t)} f u_I.n_k dA - \int_{A_{1k}(z,t)} f u_{A11}.n_z dA \\ &+ \int_{A_{2k}(z,t)} f u_{A_{12}}.n_z dA.\end{aligned} \tag{3.8}$$

Note that the integral over the wall is zero due to the fact that the velocity at the wall is assumed to be zero. $(u_I.n_k)$ represents the speed of displacement of the interface A_I. By considering the general balance equation for V_k (Eq. 3.4), Eq. 3.8 and also Green's

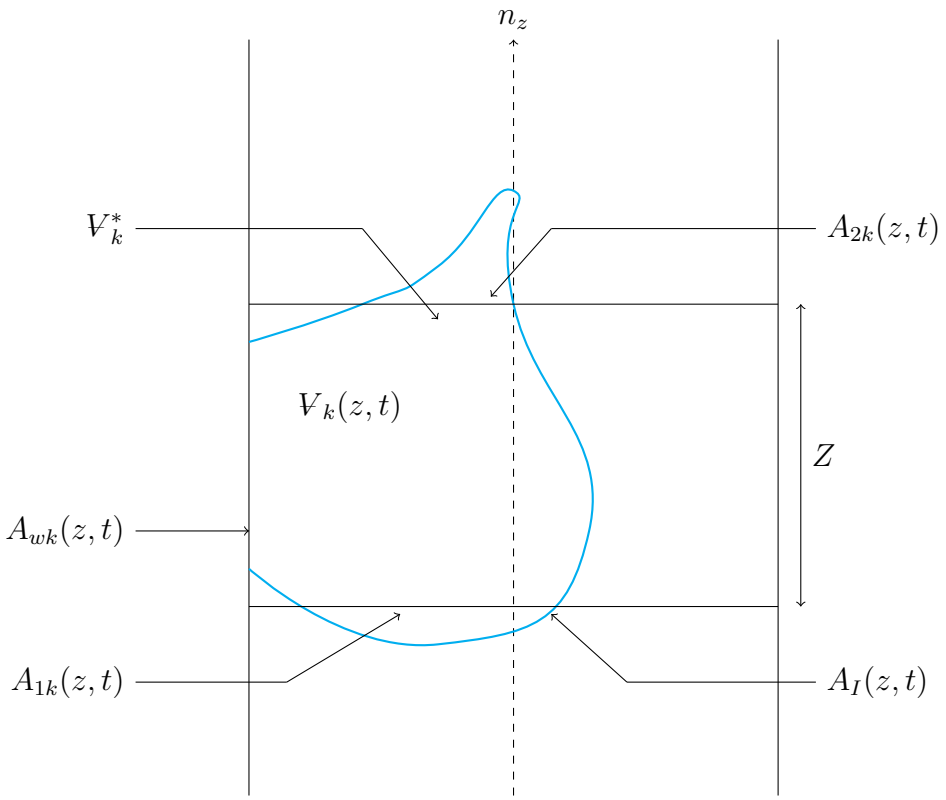


Figure 3.3: Fixed tube containing two fluids separated by an interface and intersected by two cross sectional planes

theory, the following equation is derived

$$\begin{aligned}
&\frac{\partial}{\partial t}\int_{V_k(z,t)} \rho_k\psi_k dV = \int_{A_I(z,t)} \rho_k\psi_k u_I.n_k dA \\
&-\int_{A_{1k}(z,t)} \rho_k\psi_k u_{A_{11}}.n_z dA + \int_{A_{2k}(z,t)} \rho_k\psi_k u_{A_{12}}.n_z dA \\
&-\int_{A_I(z,t)} \left(n_k.\rho_k\psi_k u_k + \mathcal{J}_k\right) dA - \int_{A_{wk}(z,t)} n.\mathcal{J}_k dA \\
&+\int_{A_{1k}(z,t)} n_z.\left(\rho_k\psi_k u_k.\mathcal{J}_k\right) dA - \int_{A_{2k}(z,t)} n_z.\left(\rho_k\psi_k u_k.\mathcal{J}_k\right) dA \\
&+\int_{V_k(z,t)} \rho_k\phi_k dV\,.
\end{aligned} \tag{3.9}$$

By defining the volume averaging operator as

$$\langle f\rangle_3 = \frac{1}{V}\int_V f dV, \tag{3.10}$$

and also using the mass transfer at the interface, $\dot{m}_k$, Eq. 3.9 is written as

$$\begin{aligned}
&\frac{\partial}{\partial t} V_k\langle\rho_k\psi_k\rangle_3 - V_k\langle\rho_k\phi_k\rangle_3 = \\
&\int_{A_{1k}} n_z.\left[\rho_k\psi_k(u_k - u_{A_{11}}) + \mathcal{J}_k\right] dA \\
&-\int_{A_{2k}} n_z.[\rho_k\psi_k(u_k - u_{A_{12}}) + \mathcal{J}_k] dA \\
&-\int_{A_I} (\dot{m}_k\psi_k + u_k.\mathcal{J}_k) dA - \int_{A_{wk}} u_k.\mathcal{J}_k dA.
\end{aligned} \tag{3.11}$$

Equation 3.11 corresponds to the instantaneous volume averaged equations. By using the values shown in Table 3.1, each volume averaged conservation law is obtained.

3.2.2 Double Volume-Time Averaged Equations

Most practical situations, require the governing equations to be averaged over both space and time. Here, the volume averaged equations (Eq. 3.11) are averaged over time to obtain the *Double Volume-Time Averaged Equations*. Consider the time averaging operator over a period of $[T]$ defined as

$$\overline{f_k} = \frac{1}{T}\int_{[T]} f_k dt. \tag{3.12}$$

Since no jump discontinuities occur in f_k (due to the fact that a volume average was used instead of a surface average in the derivation of Eq. 3.11), the time average of a partial derivative is equal to the partial differentiation of a time averaged function [32]. Therefore,

$$\frac{\partial \overline{f_k}}{\partial t} = \overline{\frac{\partial f_k}{\partial t}}. \tag{3.13}$$

Integrating Eq. 3.11 over a time interval of $[T]$ (and assuming the two cross sectional planes to remain stationary) gives

$$\begin{aligned}
&\int_{[T]} \frac{\partial}{\partial t} \forall_k \langle \rho_k \psi_k \rangle_3 dt - \int_{[T]} \forall_k \langle \rho_k \phi_k \rangle_3 dt = \\
&\int_{[T]} \int_{A_{1k}} n_z . \left[\rho_k \psi_k (u_k - u_{A_{11}}) + \mathcal{J}_k\right] dA dt \\
-&\int_{[T]} \int_{A_{2k}} n_z . [\rho_k \psi_k (u_k - u_{A_{12}}) + \mathcal{J}_k] dA dt \\
-&\int_{[T]} \int_{A_I} (\dot{m}_k \psi_k + u_k . \mathcal{J}_k) dA dt - \int_{[T]} \int_{A_{wk}} u_k . \mathcal{J}_k dA dt.
\end{aligned} \tag{3.14}$$

Using the definition of the time averaging operator from Eq. 3.12, Eq. 3.14 may is rewritten

as

$$\begin{aligned}
&\frac{\partial}{\partial t}\overline{V_k\langle\rho_k\psi_k\rangle_3} - \overline{V_k\langle\rho_k\phi_k\rangle_3} = \\
&\int_{A_{1k}} \overline{n_z.(\rho_k\psi_k u_k + \mathcal{J}_k)}dA \\
&-\int_{A_{2k}} \overline{n_z.(\rho_k\psi_k u_k + \mathcal{J}_k)}dA \\
&-\overline{\int_{A_I}(\dot{m}_k\psi_k + u_k.\mathcal{J}_k)dA} - \overline{\int_{A_{wk}} u_k.\mathcal{J}_k dA}.
\end{aligned} \tag{3.15}$$

The first two terms on the RHS of Eq. 3.15 are cast in the form of a differential term because the distance separating the two cross sectional planes is arbitrarily small, hence

$$\begin{aligned}
\frac{\partial}{\partial t}\overline{V_k\langle\rho_k\psi_k\rangle_3} - \overline{V_k\langle\rho_k\phi_k\rangle_3} = -\frac{\partial}{\partial z}V_k\left\langle\overline{n_z.(\rho_k\psi_k u_k + \mathcal{J}_k)}\right\rangle \\
-\overline{\int_{A_I}(\dot{m}_k\psi_k + u_k.\mathcal{J}_k)dA} - \overline{\int_{A_{wk}} u_k.\mathcal{J}_k dA}.
\end{aligned} \tag{3.16}$$

Eq. 3.16 is the double volume-time averaged equation and again, values from Table 3.1 are used for each conservation law.

3.2.3 Averaged Interface Balance Equation

In addition to the averaged governing equations for each phase, the interface jump balance also needs to be averaged. Using a similar procedure, averaging Eq. 3.7 leads to

$$\sum_{k=1,2}\left(\overline{\int_{A_I}\dot{m}_k\psi_k da} + \overline{\int_{A_I} n_k.\mathcal{J}_k dA}\right) = 0 \tag{3.17}$$

3.3 Two-Phase Flow Models

In the present section, two of the most commonly used two phase models are introduced. In each model, certain assumptions are made in order to simplify the governing equations and make them easier to solve. The most general model used for 1D simulations is the two pressure six equation model where each phase has a different pressure. The 1D system of equations obtained from the volume-time averaged balance equations for each conservation law are as follows (the overbar has been dropped for simplicity).

Conservation of mass:

$$\frac{\partial}{\partial t}(\alpha_k \rho_k) + \frac{\partial}{\partial z}(\alpha_k \rho_k u_k) = \phi'''_{ck}. \tag{3.18}$$

Conservation of momentum:

$$\frac{\partial}{\partial t}(\alpha_k \rho_k u_k) + \frac{\partial}{\partial z}(\alpha_k \rho_k u_k^2) + \int_{A_I} n_z.n_k.p_k dA + \int_{A_{kw}} n_k.n_z p_k.dA = \phi'''_{mk}. \tag{3.19}$$

Conservation of energy:

$$\frac{\partial}{\partial t}(\alpha_k \rho_k E_k) + \frac{\partial}{\partial z}(\alpha_k \rho_k E_k u_k) + \int_{A_I} n_z.n_k.p_k.u_I dA + \int_{A_{kw}} n_k.n_z p_k.u_k.dA = \phi'''_{ek}. \tag{3.20}$$

In Eqs. 3.18, 3.19 and 3.20, α_k is the volume fraction of phase k (arising from the division of each conservation law by the total volume, V). The ϕ''' terms, contain all the source terms for each equation. The $\int_{A_{kw}} n_k.n_z p_k.dA$ term in the momentum equation is usually neglected when a constant cross section is used [32] and the $\int_{A_{kw}} n_k.n_z p_k.u_k.dA$ term in the energy equation is equal to zero due to the velocity being zero at the wall. For the pressure term seen in the momentum equation, Eq. 3.19 (and also the energy equation), two main approaches are taken in the literature. First is the case where each phase has a

constant pressure of p_k in order to derive a two pressure model. Such an approach can be found in [10]. The integral is then written as

$$\int_{A_I} n_z.n_k.p_k dA = -p_k \frac{\partial \alpha_k A}{\partial z}. \tag{3.21}$$

The second approach, which is selected for the current study, is to assume an equal pressure for both phases and introduce an interface pressure, p^{int}. Therefore, the integral becomes

$$\int_{A_I} n_z.n_k.p_k dA = -p^{int} \frac{\partial \alpha_k A}{\partial z}. \tag{3.22}$$

Further information on p^{int} is given in Section 3.4. Following this approach, the general form of the single pressure two-fluid model can be written as follows [63]

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} = C^{nv} + C^{v} + S. \tag{3.23}$$

In Eq. 3.23, U is the vector of conserved variables and E is the corresponding flux vector. The source terms, have been separated into three terms C^{nv}, C^{v} and S. The first two terms contain differential terms and the last term carries non-differential terms. C^{nv} represents all the non-viscous differential terms including the interfacial pressure correction term used as a regularization term. C^{v} includes all the viscous differential terms and S is the source term containing all of the non-differential terms such as gravity and interfacial drag. In the present study, the viscous terms are neglected, meaning $C^{v} = 0$. It is also assumed that no phase change occurs between the liquid and gas phases. These assumptions simplify Eq. 3.23 to the following form

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} = C^{nv} + S. \tag{3.24}$$

3.3.1 Six Equation Model

The six equation model consists of a mass, momentum and energy balance for each phase. In order to cast the model in the general form shown in Eq. 3.24 the following definitions

are used.

$$U = \begin{pmatrix} \alpha_g\rho_g \\ \alpha_l\rho_l \\ \alpha_g\rho_g u_g \\ \alpha_l\rho_l u_l \\ \alpha_g\rho_g E_g \\ \alpha_l\rho_l E_l \end{pmatrix}, \qquad E = \begin{pmatrix} \alpha_g\rho_g u_g \\ \alpha_l\rho_l u_l \\ \alpha_g\rho_g u_g^2 + \alpha_g p \\ \alpha_l\rho_l u_l^2 + \alpha_l p \\ \alpha_g\rho_g u_g H_g \\ \alpha_l\rho_l u_l H_l \end{pmatrix},$$
$$C^{nv} = \begin{pmatrix} 0 \\ 0 \\ p\frac{\partial\alpha_g}{\partial x} + F_g^{nv} \\ p\frac{\partial\alpha_l}{\partial x} + F_l^{nv} \\ -p\frac{\partial\alpha_g}{\partial t} + u^{int}F_g^{nv} \\ -p\frac{\partial\alpha_l}{\partial t} + u^{int}F_l^{nv} \end{pmatrix}, \qquad S = \begin{pmatrix} 0 \\ 0 \\ \alpha_g\rho_g g_x + F_g^D \\ \alpha_l\rho_l g_x + F_l^D \\ \alpha_g\rho_g u_g g_x + u^{int}F_g^D \\ \alpha_l\rho_l u_l g_x + u^{int}F_l^D \end{pmatrix}. \tag{3.25}$$

In Eq. 3.25, the subscripts g and l represent the gas and liquid phase respectively. α_k is the volume fraction of phase k and $\alpha_g + \alpha_l = 1$ must hold at all times. ρ_k and u_k show the density and velocity of phase k respectively. p is the common pressure between both phases (single pressure model). E_k is the specific total energy of phase k and H_k is the specific total enthalpy of phase k. They are related by $H_k = E_k + p/\rho_k$. A pressure correction term is added in order to make the system of equations hyperbolic. F_k^{nv} is the pressure correction term for phase k (more about this in Sec. 3.4). In Eq. 3.25 the gravity vector can be seen as g_x and corresponds to the projection of the gravity term in the x-direction of the flow. The interfacial drag term of phase k is represented by F_g^D.

System 3.25 is closed by defining the Equations Of State (EOS) for each phase, e.g. the ideal gas law for the gas phase and the stiffened gas law (Tait's EOS) for the liquid phase [63]. For the gas phase, the perfect gas law states,

$$p(\rho_g, e_g) = (\gamma_g - 1)\rho_g e_g, \tag{3.26a}$$

$$p(\rho_g, T_g) = \rho_g R_g T_g, \tag{3.26b}$$

$$e_g(\rho_g, T_g) = \frac{R_g}{\gamma_g - 1} T_g. \tag{3.26c}$$

For air, the following values are considered [63]

$$\gamma_g = 1.4, \qquad R_g = 288.2 \frac{J}{kg^\circ K}.$$

For the liquid phase, Tait's EOS assumes

$$p(\rho_l, e_l) = (\gamma_l - 1)\rho_l e_l - \gamma_l p_\infty, \tag{3.27a}$$

$$p(\rho_l, T_l) = \frac{\gamma_l - 1}{\gamma_l} \rho_l C_p T_l - p_\infty, \tag{3.27b}$$

$$e_l(\rho_l, T_l = \frac{C_p}{\gamma_l} T_l + \frac{p_\infty}{\rho_l}. \tag{3.27c}$$

Assuming, water is selected as the fluid under consideration for the liquid phase, the following values may be used [63]

$$\gamma_l = 2.8, \qquad p_\infty = 8.5 \times 10^8 \text{Pa}, \qquad C_p \approx 4186 \frac{J}{kg^\circ K}$$

The speed of sound for each phase can also be calculated by

$$a_g(T_g) = \sqrt{\frac{\gamma_g p}{\rho_g}} = \sqrt{\gamma_g R_g T_g}, \tag{3.28a}$$

$$a_l(T_l) = \sqrt{\gamma_l \frac{p + p_\infty}{\rho_l}} = \sqrt{(\gamma_l - 1) C_p T_l}. \tag{3.28b}$$

3.3.2 Four Equation Model (Isentropic)

The four equation model assumes both phases to be in thermal equilibrium. Therefore, the energy equation is not necessary and only the mass and momentum balances are written and solved. In order for the model to be consistent with the general form of Eq. 3.23, the following definitions need to be posed

$$U = \begin{pmatrix} \alpha_g \rho_g \\ \alpha_l \rho_l \\ \alpha_g \rho_g u_g \\ \alpha_l \rho_l u_l \end{pmatrix}, \qquad E = \begin{pmatrix} \alpha_g \rho_g u_g \\ \alpha_l \rho_l u_l \\ \alpha_g \rho_g u_g^2 + \alpha_g p \\ \alpha_l \rho_l u_l^2 + \alpha_l p \end{pmatrix},$$

$$C^{nv} = \begin{pmatrix} 0 \\ 0 \\ p\frac{\partial \alpha_g}{\partial x} + F_g^{nv} \\ p\frac{\partial \alpha_l}{\partial x} + F_l^{nv} \end{pmatrix}, \qquad S = \begin{pmatrix} 0 \\ 0 \\ \alpha_g \rho_g g_x + F_g^D \\ \alpha_l \rho_l g_x + F_l^D \end{pmatrix}. \tag{3.29}$$

The same definitions used for Eq. 3.25 are used here. The equations of state slightly differ due to the fact that the temperature does not change in this model. Here, for the gas phase, the perfect gas law states

$$p = p(\rho_g) = C\left(\frac{\rho - g}{\rho_g^\circ}\right)^\gamma, \tag{3.30}$$

with $\gamma = 1.4$, $C = 10^5$ Pa, $\rho_g^\circ = 1$ kg/m^3 as appropriate values for air [63]. The speed of sound in the gas phase can be evaluated from

$$a_g = \sqrt{\frac{\gamma p}{\rho_g}}. \tag{3.31}$$

For the liquid phase, Tait's law becomes

$$p = p(\rho_l) = B\left[\left(\frac{\rho_l}{\rho_l^\circ}\right)^n - 1\right], \tag{3.32}$$

with the values $n = 7.15$, $B = 3.3 \times 10^8$ Pa and $\rho_l^\circ = 1000$ kg/m^3 being representatives of water [25, 34]. The speed of sound for the liquid phase is also evaluated by

$$a_l = \sqrt{\frac{n}{\rho_l}(p + B)} \tag{3.33}$$

3.4 Analysis of Two-Phase Flow System of Equations

Some simplifying assumptions were made to the general two fluid model as shown in Eq. 3.24. Even with these simplifying assumptions, the system of equations still poses some mathematical difficulties. Due to the presence of the source term (C^{nv}) in Eq. 3.24, the system of equations is not in conservative form. Further, a characteristic analysis on the system of Eq. 3.24 shows that in general, two-fluid models are non-hyperbolic and have complex characteristic values associated with them [79]. As a result, numerical oscillations might be noticed due to the ill-posedness of the problem. In order to solve this issue, regularization terms in the form of virtual mass or pressure correction terms, are included which can make the system conditionally or unconditionally hyperbolic. In the current study, an interfacial pressure correction term is added for each phase (F_k^{nv}) and is given by [63]

$$F_k^{nv} = (p^{int} - p)\frac{\partial \alpha_k}{\partial x}, \tag{3.34}$$

where p^{int} is the interfacial pressure term which can be evaluated by [9]

$$p^{int} = p - \sigma \frac{\alpha_g \rho_g \alpha_l \rho_l}{\alpha_g \rho_l + \alpha_l \rho_g}(u_g - u_l)^2. \tag{3.35}$$

In Eq. 3.35 σ is a positive constant and a choice of $\sigma \geq 1$ makes the system of equations hyperbolic [63]. The interfacial drag force, F_g^D in Eq. 3.29, still remains to be defined. In the present study, the drag force for the gas phase is defined as

$$F_g^D = -C_f \alpha_g (1 - \alpha_g)\rho_g (u_g - u_l). \tag{3.36}$$

Using Newton's second law, the drag force for the liquid phase is found by

$$F_l^D = -F_g^D. \tag{3.37}$$

In Eqs. 3.36 and 3.37, C_f is the interfacial drag coefficient and is a positive constant. The value of C_f depends on the test case used. The final term which needs to be defined is the interfacial velocity, u^{int}. In general the interfacial velocity is modelled as a weighted average of the velocity of the two phases (liquid and gas). Therefore,

$$u^{int} = \beta u_g + (1-\beta)u_l, \quad \beta \in [0,1]. \tag{3.38}$$

Two common choices of β are $\beta = 0.5$ or $\beta = \alpha_g$ [63]. However, the term $u^{int}F_k^{nv}$ is usually neglected in the energy equation.

3.5 Previously Used Solution Methods

Due to the importance of simulating multiphase flows in engineering fields such as the nuclear industry and the oil and gas industry, much research has been focused on obtaining good results for the solution to different multiphase models. The majority of the literature available, focuses on explicit schemes for the solution to multiphase problems and has tried to develop accurate methods to resolve the discontinuity observed in different parameters. A brief review of some of the relevant work on two-phase flows are mentioned in the present section along with appropriate references.

The system of equations governing two-phase flows are non hyperbolic due to complex eigenvalues, which introduce extra difficulty to finding a numerical solution. Numerous studies are focused on the hyperbolicity of the governing equations. Some of the important contributions are listed here. In [64] a two pressure model is considered with the addition of a virtual mass term using the isentropic case. In [57] a single pressure model is used and results obtained by including and not including a virtual mass term, are compared. Banarjee et al. [4] perform a more complete study on the hyperbolicity of single pressure,

two pressure and the homogeneous model. Stewart [78] suggests the usage of semi-implicit methods to avoid the problems caused by the non-hyperbolicity of the equations. Ransom [67] utilizes a pressure correction term on a two pressure model to make the system of equations hyperbolic. A good review paper by Stewart et al. [79] gathers such approaches and discusses them thoroughly. In addition, [85, 87] gathers some of these aspects with more detail.

The main issue with the non-hyperbolicity of the governing equations is the fact that the initial value problem no longer continuously depends on the initial values and becomes ill-possed. As mentioned in Sec. 3.4, the addition of a virtual mass term or a pressure correction term can remedy the issue to a certain extent.

Due to the non conservative nature of the momentum and energy equations, researchers mostly utilize non conservative schemes to numerically solve the system of equations. Examples of using conservative schemes for the solution to the two fluid model are [8, 84, 63]. However, despite issues with the non conservative form of the equations, conservative schemes have been extended for the solution to the four and six equations models to achieve higher resolutions at capturing discontinuities seen in the solution to two-phase problems. In the class of conservative schemes, two main approaches exist, a centered approach and an upwind approach. Upwind approaches take into consideration the direction of propagation of information and are more physical. Upwind methods can be further divided into two main classes, Flux Difference Splitting schemes (FDS) and Flux Vector Splitting scheme (FVS). Godunov-type methods [72] and the Roe and Osher's method [86, 62] fall in the FDS category, while Steger-Warming [77] and the van Leer [48] scheme fall into the FVS category. The main difference between the two is the way the numerical fluxes are defined. A complete description of these methods can be found in books such as [49, 83]. The main advantage of FVS compared to FDS is that in general the schemes are much simpler and more efficient. However, this comes at the cost of having a lower resolution near discontinuities.

A relatively newer approach arises from the combination of FVS and FDS, called hybrid flux splitting schemes. Some examples of these methods would be LDFSS [22], HUS [18] and AUSM [55]. The AUSM class of schemes is chosen for the present study due to their proven accuracy and robustness. Many different variations of the AUSM have been

developed such as AUSM$^+$ [52], AUSM$^+$-up [53], SLAU [74], AUSM$^+$-up2 [39], SLAU2 [39], AUSM$^+$-up (Niu) [61] and AUSMPW$^+$ [37]. Each method has some advantages and disadvantages. A complete comparison of these methods is performed in [38].

All of the mentioned methods are explicit schemes. A small fraction of studies have also been dedicated to implicit schemes for the solution of multiphase problem modeled by the two fluid equations [59, 24]. Implicit methods have the main advantage of not being restricted to the Courant-Friedrichs-Lewy (CFL) condition and are able to produce stable results while using larger time steps to advance in time. In nuclear reactor safety codes such as CATHENA, RELAP and TRAC, semi-implicit and implicit schemes are used on staggered grids to solve the system of equations. Simple upwind schemes are used for the convective flux terms. The main issue with the current methods used in such codes is that they do not produce enough numerical damping when fine grids are used and only coarse grids produce stable results. One of the goals of the present work is to utilize more effective shock capturing schemes (AUSM$^+$) and be able to perform the simulations on fine grids, while solving the equations using a fully implicit discretization. In order to do so, the JFNK method is used. To the authors knowledge, the JFNK method has not been applied to multiphase flow problem before.

Chapter 4

Application of JFNK to the Shallow Water Wave Equations (SWWE)

In order to advance towards the implementation of the JFNK method to two-phase flows, as a first step, the method is applied to a set of equations applicable to single phase flows. The equations are related to the dam break problem and correspond to the one dimensional Shallow Water Wave Equations (SWWE). The SWWE are a set of coupled, non-linear, hyperbolic PDEs for a single phase fluid. In the present chapter, the SWWE are derived and discussed. Different discretization schemes used to solve the governing equations are investigated. Throughout the numerical simulations, all terms are calculated implicitly (all the unknowns are calculated at the next time step simultaneously) and the JFNK method is applied to solve the system of equations. In addition, the numerical predictions are compared with analytical solutions.

4.1 Shallow Water Wave Equations

In the present section, the Shallow Water Wave Equations (SWWE) are derived from Euler's conservation laws. The SWWE are a set of PDEs used to model the propagation of disturbances in incompressible fluids, with the main assumption that the depth of the

fluid is much smaller than the wavelength of the disturbance. For example, oceans have a depth of two to three kilometers, however, waves created in the ocean can be hundreds of kilometers long. Therefore, the SWWE can be utilized to model this situation [58]. There are many different approaches which can be adapted in order to derive the depth-averaged shallow water equations. Here, the method presented is similar to that of Ligget [51]. Three main assumptions regarding the fluid are made when deriving the SWWE. The fluid is assumed to be incompressible, irrotational and inviscid. The latter two are made when deriving Euler's equations. Also, the construction of the depth-averaged SWWE require three main assumptions about the spatial domain: the vertical extent of the fluid should be much smaller than the horizontal length, pressure is assumed to increase linearly with depth and it is also assumed that the vertical acceleration within the fluid is negligible.

4.1.1 Boundary Conditions

Prior to the derivation of the SWWE, boundary conditions need to be specified for the domain under consideration. These boundary conditions are defined at the free surface of the fluid and the bottom of the domain. The location of the free surface is identified by

$$z = f(x, y, t). \tag{4.1}$$

Therefore, the velocity of the free surface is given by $\vec{V} = (0, 0, \partial f/\partial t)$ and the velocity of a fluid particle is given by $\vec{u} = (u, v, w)$. It is assumed that a fluid particle cannot cross the interface, therefore,

$$\vec{V}.\vec{n} = \vec{u}.\vec{n}, \tag{4.2}$$

where $\vec{n}$ is the normal vector to the free surface and is given by

$$\vec{n} = \left(-\frac{\partial f}{\partial x}, -\frac{\partial f}{\partial y}, 1\right). \tag{4.3}$$

Calculation of Eq. 4.2 gives

$$\frac{\partial f}{\partial t} = -u\frac{\partial f}{\partial x} - v\frac{\partial f}{\partial y} + w. \tag{4.4}$$

The free surface height can be considered as the sum of the bottom topography and the depth of water at a certain location. This implies, $f(x, y, t) = h(x, y, t) + b(x, y)$ where b is the elevation of the topography of the domain and h is the depth of water calculated from the topography to the surface. Denoting the values of surface variables with $\hat{}$, and the evaluation of Eq. 4.4 at the free surface gives the top surface boundary condition

$$\frac{\partial h}{\partial t} + \hat{u}\frac{\partial}{\partial x}(b + h) + \hat{v}\frac{\partial}{\partial y}(b + h) - \hat{w} = 0. \tag{4.5}$$

By assuming that water cannot cross the bottom topography, the same methodology may be used again. Denoting values of bottom topography variables with $\breve{}$, the bottom surface boundary condition (where $h = 0$) can be obtained from

$$\breve{u}\frac{\partial b}{\partial x} + \breve{v}\frac{\partial b}{\partial y} - \breve{w} = 0. \tag{4.6}$$

The no slip boundary condition can also be applied at the bottom of the domain. This means

$$\breve{u} = 0 \quad , \quad \breve{v} = 0 \quad , \quad \breve{w} = 0. \tag{4.7}$$

4.1.2 Conservation of Mass

Euler's continuity equation is given by

$$\frac{1}{\rho}\frac{\partial \rho}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \tag{4.8}$$

where as defined in Sec. 4.1.1, u, v and w are the x, y and z components of the fluid velocity, respectively. Equation 4.8 implies that the rate of change of density considered

inside a control volume is equal to the expansion of the fluid. Since the fluid is assumed to be incompressible, the change in density is equal to zero. Therefore, Eq. 4.8 is reduced to

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0. \tag{4.9}$$

Integration of Eq. 4.9 across the vertical length of the fluid leads to

$$\int_b^{b+h} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) dz = 0. \tag{4.10}$$

By applying Leibniz' theorem to Eq. 4.10

$$\begin{aligned} &\int_b^{b+h} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) dz = \\ &\frac{\partial}{\partial x} \int_b^{b+h} u dz - \hat{u} \frac{\partial}{\partial x}(b+h) + \breve{u} \frac{\partial b}{\partial x} + \\ &\frac{\partial}{\partial y} \int_b^{b+h} v dz - \hat{v} \frac{\partial}{\partial y}(b+h) + \breve{v} \frac{\partial b}{\partial y} + \\ &\int_b^{b+h} \frac{\partial w}{\partial z} dz = 0. \end{aligned} \tag{4.11}$$

An average velocity over the depth of the domain in the x and y directions is defined such that

$$\bar{u} = \frac{1}{h} \int_b^{b+h} u dz, \tag{4.12a}$$

$$\bar{v} = \frac{1}{h} \int_b^{b+h} v dz. \tag{4.12b}$$

Substitution of Eqs. 4.12a and 4.12b into Eq. 4.11 and applying boundary conditions shown in Eqs. 4.5 and 4.6 results in the conservation of mass for the two dimensional depth-averaged shallow water equations,

$$\frac{\partial}{\partial t} h + \frac{\partial}{\partial x} h\bar{u} + \frac{\partial}{\partial y} h\bar{v} = 0. \tag{4.13}$$

4.1.3 Conservation of Momentum

Euler's equations of motion in differentiable form are,

in the x-direction,

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial x}, \tag{4.14}$$

and in the y-direction

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial y}. \tag{4.15}$$

In Eqs. 4.14 and 4.15, the pressure within the fluid is assumed to increase linearly with depth, meaning

$$\Delta p = \rho g \left(z_1 - z_0\right), \tag{4.16}$$

where $(z_1 - z_0)$ is the distance between any two arbitrary points and Δp corresponds to the pressure difference between the two points. Using Eq. 4.8 and integrating Eq. 4.14 over the vertical length of the fluid yields

$$\int_b^{b+h} \left(\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z} + \frac{1}{\rho}\frac{\partial p}{\partial x}\right) dz = 0. \tag{4.17}$$

Applying Leibniz theorem to each term of Eq. 4.17 and considering the depth averaged velocities (Eqs. 4.12a and 4.12b) and also the free surface and bottom boundary conditions (Eqs. 4.5 and 4.6) leads to the second of the three SWWE (the derivation is rather long and is not included in this thesis, for further information, the reader can consult [51]). A similar approach is taken for the momentum equation in the y - direction. After integration of Eqs. 4.14 and 4.15 , the other two equations can be derived

$$\frac{\partial}{\partial t}(\bar{u}h) + \frac{\partial}{\partial x}(\bar{u}^2 h) + \frac{\partial}{\partial y}(\bar{u}\bar{v}h) + g\frac{1}{2}\frac{\partial h^2}{\partial x} = -gh\frac{\partial b}{\partial x}, \tag{4.18}$$

$$\frac{\partial}{\partial t}(\bar{v}h) + \frac{\partial}{\partial y}(\bar{v}^2 h) + \frac{\partial}{\partial x}(\bar{u}\bar{v}h) + g\frac{1}{2}\frac{\partial h^2}{\partial y} = -gh\frac{\partial b}{\partial y}. \tag{4.19}$$

Equations 4.13, 4.18 and 4.19 are the SWWE in two dimensions. Considering the depth averaged velocities for the remainder of the current chapter, the average sign may be dropped and the general form of the two dimensional SWWE can be written as

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ uh \\ vh \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} uh \\ u^2h + gh^2/2 \\ uvh \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} vh \\ uvh \\ v^2h + gh^2/2 \end{bmatrix} = \begin{bmatrix} 0 \\ -gh\frac{\partial b}{\partial x} \\ -gh\frac{\partial b}{\partial y} \end{bmatrix}. \tag{4.20}$$

4.1.4 General Form of the One Dimensional SWWE

In the current study, only the one dimensional case is investigated and energy transfer is not considered, corresponding to an isothermal case. The general conservative form of the one dimensional SWWE is given by [26]

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ uh \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} uh \\ u^2h + gh^2/2 \end{bmatrix} = \begin{bmatrix} 0 \\ -gh\frac{\partial b}{\partial x} \end{bmatrix}. \tag{4.21}$$

Equation 4.21 can be cast into a more compact form as follows

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} = S, \tag{4.22}$$

where Q is the vector of conserved variables and F is the flux vector and S is the source term vector with the following definitions

$$Q = \begin{bmatrix} h \\ uh \end{bmatrix}, \quad F = \begin{bmatrix} uh \\ u^2h + gh^2/2 \end{bmatrix}, \quad S = \begin{bmatrix} 0 \\ -ghb_x \end{bmatrix}. \tag{4.23}$$

In Eq. 4.23, b_x is the first derivative of the bottom topography elevation, but in the current study, the homogeneous system is analyzed, therefore, $S = 0$ (bottom surface is flat).

4.1.5 Mathematical Properties

In the present study, the homogeneous form of the SWWE is considered. This form of the system, does not include any source terms ($S = 0$ in Eq. 4.22). Writing Eq. 4.22 in quasilinear form [51] leads to

$$Q_t + F^{'}(Q)Q_x = 0, \tag{4.24}$$

where $F^{'}(Q)$ is the Jacobian matrix of the flux vector, with respect to the conserved variables. In order to be able to calculate the Jacobian matrix, the following steps are taken: first, the flux vector is written as a function of the conserved variables,

$$Q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} h \\ uh \end{bmatrix} \quad \text{and}$$

$$F = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} uh \\ u^2h + gh^2/2 \end{bmatrix} = \begin{bmatrix} (uh) \\ (uh)^2/h + gh^2/2 \end{bmatrix} = \begin{bmatrix} q_2 \\ (q_2)^2/q_1 + g(q_1)^2/2 \end{bmatrix}. \tag{4.25}$$

The Jacobian matrix can easily be calculated using its definition

$$F^{'}(Q) = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} \\ \\ \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \\ -u^2 + gh & 2u \end{bmatrix}. \tag{4.26}$$

The homogeneous shallow water system of equations is hyperbolic due to the fact that the Jacobian matrix calculated in Eq. 4.26 has real eigenvalues. These eigenvalues and their corresponding eigenvectors are

$$\lambda_1 = u - \sqrt{gh} \quad \& \quad \lambda_2 = u + \sqrt{gh}, \tag{4.27a}$$

$$r_1 = \begin{bmatrix} 1 \\ u - \sqrt{gh} \end{bmatrix} \quad \& \quad r_2 = \begin{bmatrix} 1 \\ u + \sqrt{gh} \end{bmatrix}. \tag{4.27b}$$

Due to the hyperbolic nature of the SWWE, the eigenvalues in Eq. 4.27a indicate the characteristic speeds which determine how fast certain properties are propagated through time. In the case of the SWWE, the fluid velocity u, and the water depth h are the propagated variables. It can be shown that the characteristic curves obtained from the eigenvalues are [51]

$$C_1 : \frac{dx}{dt} = u - \sqrt{gh} \quad \& \quad C_2 : \frac{dx}{dt} = u + \sqrt{gh}. \tag{4.28}$$

The curvature of these curves are dependent on the point where they pass at $t = 0$ (s) and the Riemann invariants are constant along these curves. The Riemann invariants for the 1D shallow water equations are

$$u - 2\sqrt{gh} = k_1 \quad \& \quad u + 2\sqrt{gh} = k_2. \tag{4.29}$$

Using the initial profile of Q, values for u and h can be approximated at some time in the near future, Δt. This is done by plotting the curves C_1 and C_2 at each point on the x - axis and finding the intersection of the characteristics at $t = \Delta t$. Because the values of k_1 and k_2 remain constant on the curves C_1 and C_2, respectively, u and h can be approximated at the intersection (keep in mind the values of k_1 and k_2 are calculated using the initial profile).

This approach is valid only if two or more characteristics of the same family do not intersect. If they do, a shock will appear [50]. The speed of this shock, $s(t)$, can be calculated using the state vectors immediately to the left and right of the shock. These vectors are shown with Q_l and Q_r, respectively. Applying the conservation laws to a small region containing the shock yields

$$s = \frac{F(Q_r) - F(Q_l)}{Q_r - Q_l}. \tag{4.30}$$

Equation 4.30 is known as the Rankine-Hugoniot condition and any solution satisfying it across a shock is called a weak solution to the Riemann problem [50]. The Rankine-Hugoniot condition does not guarantee a unique, physical solution. Therefore, additional constraints are necessary. The Lax entropy condition is a commonly used constraint for the SWWE [50]. It states

$$\lambda_1(Q_l) > s > \lambda_1(Q_r), \tag{4.31a}$$

$$\lambda_2(Q_l) > s > \lambda_2(Q_r). \tag{4.31b}$$

If Eqs. 4.30 and 4.31 are both satisfied, it ensures that the energy decreases across a shock, and a physical solution has been found. A more detailed description can be found in [50].

4.1.6 CFL Number

For the one-dimensional shallow water wave equations, the Courant-Friedrichs-Lewy (CFL) number can be defined by [6]

$$CFL = \frac{1}{2}\frac{\Delta t}{\Delta x} max\left|\lambda_p\right|, \tag{4.32}$$

where λ_p are the eigenvalues of the system from Eq. 4.27a. This CFL condition is calculated at each computational node. The time step selected for the simulation is obtained from the minimum value within all nodes. Different numerical methods require a certain limit for the CFL number for stability reasons. For explicit schemes, in order to maintain stability, the CFL number used for the simulations should be smaller than 1. The reader should keep in mind that although this is a necessary condition for stability, it is not sufficient.

4.2 The Riemann Problem

A Riemann problem is described as a problem with two uniform sets of initial conditions at each side of a single discontinuity. The main reason why the Riemann problem is of interest, is the fact that it can be solved analytically. Consider a Riemann problem with the initial conditions

$$Q(x, t_0) = \begin{cases} q_l & x < x_0 \\ q_r & x > x_0 \end{cases} . \tag{4.33}$$

In Eq. 4.33, q_l and q_r are constant vectors corresponding to the left and right states of the Riemann problem centered at $x = x_0$ and $t = t_0 = 0$. In Eq. 4.33, as time increases, the solution stretches in space so that it only depends on a single variable x/t. By reducing the number of independent variables from 2 (x and t) to 1 (x/t) analytical solutions may be found more easily.

In the present section, only the qualitative structure of the solution is described and the analytical solution for a simplified Riemann problem (the 1D dam break problem) is included in Sec. 4.3.1. It is also assumed that initially, the water level on the left is higher than the water level on the right.

As time increases, the fluid to the left is instantly accelerated causing it to collapse on the slower moving fluid to the right of the discontinuity, which leads to the formation of a shock. Therefore, at some time $t > t_0$, the solution domain can be split into three regions: i) an undisturbed region on the left, ii) an undisturbed region on the right and iii) and intermediate region connecting the other two. Assuming the shock is moving at a constant speed, the intermediate region can be further decomposed into two regions: i) a constant region immediately behind the shock and ii) a depression wave connecting it to the undisturbed region on the left (a rarefaction fan). An alternative to the rarefaction fan connecting the two regions would be a shock. However, this would violate the entropy condition shown in Eq. 4.31. Further information can be found in [50].

4.3 Numerical Benchmark

To investigate the capabilities of the JFNK method for the shallow water wave equations, the one dimensional dam break problem is considered. This problem simulates the instantaneous collapse of a virtual dam within a long channel. The dam break problem can be categorized into two main classes, wet and dry bed. For the wet bed case, water exists on both sides of the dam and a shock is created in the solution. The dry bed case, however, only has water on one side of the dam at $t = 0\ (s)$, therefore the solution does not contain a shock or discontinuity.

4.3.1 Analytical Solution

The dam break problem can be cast in the form of a Riemann problem (Eq. 4.33) with the following definitions

$$q_l = \begin{bmatrix} h_l \\ 0 \end{bmatrix} \quad \& \quad q_r = \begin{bmatrix} h_r \\ 0 \end{bmatrix}, \tag{4.34}$$

where x_0 corresponds to the location of the dam. In order to find an analytical solution, an additional assumption is made which assumes water is initially at rest throughout the domain. Two cases are considered, a wet and a dry bed case. In the present section, the procedure for finding the analytical solution for each case is briefly described.

Wet bed

The wet bed case is considered, when water exists on both sides of the virtual dam. It is assumed that the water level on the left hand side is higher than the water level on the right hand side, hence $h_l > h_r$. Water is assumed to be at rest throughout the domain at $t = 0\ (s)$. An analytical solution to this Riemann problem has been developed by Stoker [80] in which at any future time t, the solution domain is divided into four main regions. These regions can be seen in Fig. 4.1. Figure 4.1.a illustrates the general

solution associated with the dam break Riemann problem. The thick solid lines show the location of the shock and the edges of the rarefaction fan, and the dashed lines indicate the characteristic lines. At any time, $t > 0$, it can be determined what region the solution falls in, at each spatial point. This is shown in Fig. 4.1.b. For example, the intersection between the horizontal line, $t = t_1$ and the shock line easily predicts the spatial location of the shock at $t = t_1$. Once the dam is instantly removed, a shock (hydraulic jump) is formed moving to the right caused by the fast water collapsing on the stationary water downstream from the dam. Upstream to the shock is a constant region which is connected to the upstream stationary water by a rarefaction fan. The meeting point of these two regions (undisturbed left and the rarefaction wave) is determined by the characteristic given by $x = -c_l t$, where $c_l = \sqrt{gh_l}$. Similarly, the characteristic connecting the rarefaction wave (region 2) and the constant region behind the shock (region 3), is given by $x = (u_3 - c_3)t$ where u_3 is the constant velocity of fluid particles in region 3 and $c_3 = \sqrt{gh_3}$. The position of the shock is also given by $x = S_3 t$ where S_3 is the shock speed.

After applying the conservation of mass and momentum across the shock (not a trivial process), the following expressions are obtained respectively

$$h_r(S_3 - u_r) = h_3(S_3 - u_3), \tag{4.35a}$$

$$\frac{1}{2}(c_3^2 - c_r^2) = h_3(S_3 - u_3)(u_3 - u_r). \tag{4.35b}$$

After rearrangement and considering $u_r = 0$, Eqs. 4.35a and 4.35b can be written as

$$\frac{c_r^2}{c_3^2} = \frac{h_r}{h_3} = \frac{S_3 - u_3}{S_3}, \tag{4.36a}$$

$$c_3^2 + c_r^2 = 2S_3(S_3 - u_3). \tag{4.36b}$$

Elimination of c_3 by using Eq. 4.36a in Eq. 4.36b and solving for u_3 (u_3 and S_3 are positive,

therefore the positive sign of the square root is used) yields

$$u_3 = S_3 - \frac{c_r}{4S_3}\left(1 + \sqrt{1 + \frac{8S_3^2}{c_r^2}}\right). \tag{4.37}$$

Again, eliminating u_3 from Eq. 4.37 by using Eq. 4.35a gives

$$\frac{c_3^2}{c_r^2} = \frac{1}{2}\sqrt{1 + \frac{8S_3^2}{c_r^2}} - \frac{1}{2}. \tag{4.38}$$

Equations 4.37 and 4.38 give analytical solutions for the velocity u_3 and c_3 (which can be used to evaluate h_3) for the constant region behind the shock. Proceeding to region 2, the Riemann invariant $u + 2c$ (shown in Eq. 4.29) remains constant along the characteristic lines. Therefore, on the left edge of the rarefaction fans $u + 2c = 2c_l$ and on the right edge of the rarefaction, $u + 2c = u_3 + 2c_3$. Combining the two expressions and using values of u_3 and c_3 from Eqs. 4.37 and 4.38 gives a nonlinear equation to determine the shock speed

$$S_3 = 2c_l + \frac{c_r^2}{4S_3}\left(1 + \sqrt{1 + \frac{8S_3^2}{c_r^2}}\right) - \left(2c_r^2\sqrt{1 + \frac{8S_3^2}{c_r^2}} - 2c_r^2\right)^{1/2}. \tag{4.39}$$

After the determination of the shock speed from Eq. 4.39, u_3 and c_3 can be evaluated. In region 2, the characteristic curves are determined by

$$\frac{dx}{dt} = \frac{x}{t} = u_2 - c_2. \tag{4.40}$$

Again, keeping in mind that the Riemann invariants remain constant along the characteristic curves, on the left edge of the rarefaction fan, $u + 2c = 2c_l$ and along the rarefaction fan, $u + 2c = u_2 + 2c_2$. Combining the expressions and solving them yields

$$u_2 = \frac{2}{3}\left(c_l + \frac{x}{t}\right), \tag{4.41a}$$

$$c_2 = \frac{2}{3}c_l - \frac{x}{3t}. \tag{4.41b}$$

By finding the solution for region 2, all regions are accounted for and an analytical solution is obtained. A full and more complete derivation of the solution can be found in [80]. For a certain set of initial values, the water level and its velocity at any time after the dam breaks, can be calculated by

$$h(x) = \begin{cases} h_l & x \leq -t\sqrt{gh_l} \\ h_2 = \frac{4}{9g}\left(\sqrt{gh_l} - \frac{x}{2t}\right)^2 & -t\sqrt{gh_l} \leq x \leq t(u_3 - \sqrt{gh_3}) \\ h_3 = \frac{h_r}{2}\sqrt{1 + 8\left(\frac{2h_3}{h_3 - h_r}\frac{\sqrt{h_l} - \sqrt{h_3}}{\sqrt{h_r}}\right)^2} - \frac{h_r}{2} & t(u_3 - \sqrt{gh_3}) < x < tS_3 \\ h_r & tS_3 \leq x \end{cases}, \tag{4.42a}$$

$$u(x) = \begin{cases} 0 & x \leq -t\sqrt{gh_l} \\ u_2 = \frac{2}{3}\left(\sqrt{gh_l} + \frac{x}{t}\right) & -t\sqrt{gh_l} \leq x \leq t(u_3 - \sqrt{gh_3}) \\ u_3 = S_3 - \frac{gh_r}{4S_3}\left(1 + \sqrt{1 + \frac{8S_3^2}{gh_r}}\right) & t(u_3 - \sqrt{gh_3}) < x < tS_3 \\ 0 & tS_3 \leq x \end{cases}. \tag{4.42b}$$

Here, S_3 is the shock velocity and it can be evaluated by (Eq. 4.39)

$$S_3 = 2\sqrt{gh_l} + \frac{gh_r}{4S_3}\left(1 + \sqrt{1 + \frac{8S_3^2}{gh_r}}\right) - \left(2gh_r\sqrt{1 + \frac{8S_3^2}{gh_r}} - 2gh_r\right)^{1/2}.$$

It is seen that in order to find the analytical solution, one must solve two nonlinear equations. A simpler solution was developed by Wu [90] which only requires the solution of

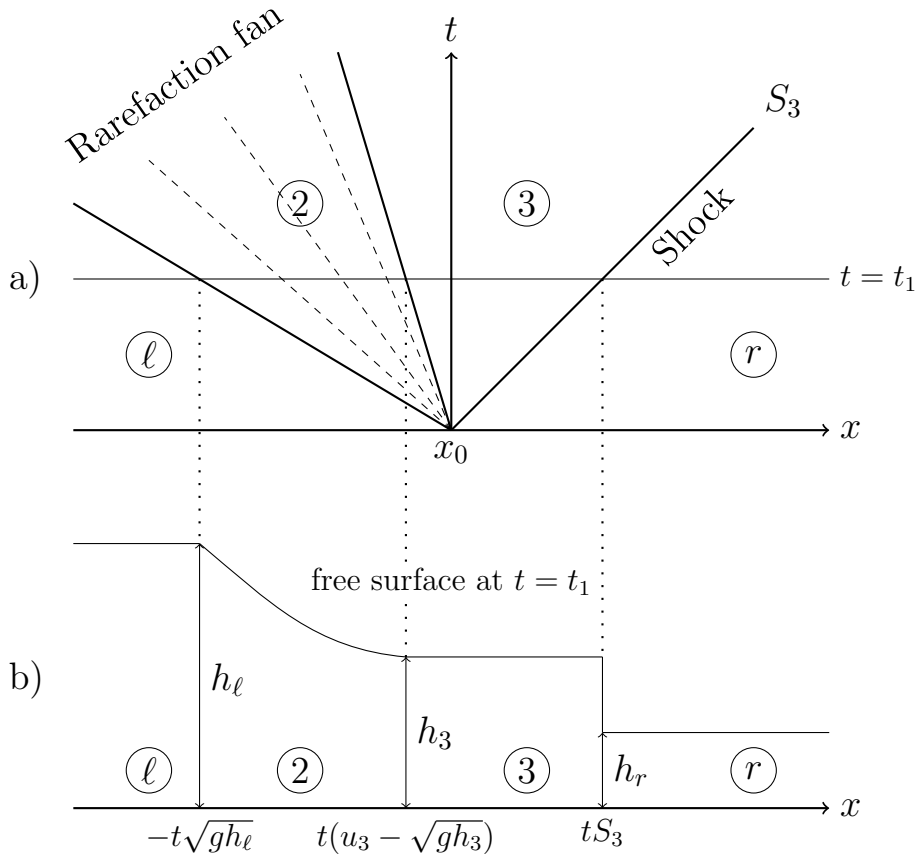


Figure 4.1: General solution to the dam break problem with a wet bed at any time $t > 0$; a) Indicates the location of the shock and the edges of the rarefaction fan and also, characteristic lines inside the rarefaction fan, b) Free surface (water level) within the domain at $t = t_1$. The correspondence of the characteristic profiles and the free surface at $t = t_1$ is also shown.

a single nonlinear equation. Therefore, first a nonlinear equation is solved to find h_3 as follows

$$h_3 = \frac{h_r}{2}\left[\sqrt{1+8\left(\frac{2h_3}{h_3-h_r}\frac{\sqrt{gh_l}-\sqrt{gh_3}}{\sqrt{gh_r}}\right)^2}-1\right]^{1/2}, \tag{4.43}$$

and the resulting value is used to determine S_3 and u_3 by

$$S_3 = \frac{2h_3}{h_3-h_r}\left(\sqrt{gh_l}-\sqrt{gh_3}\right), \tag{4.44a}$$

$$u_3 = 2\left(\sqrt{gh_l}-\sqrt{gh_3}\right). \tag{4.44b}$$

All other values may be found using Eqs. 4.42.

Dry bed

The dry bed case considers a situation where water only exists on one side of the dam. In the current configuration, water is assumed to be present on the left side of the dam. In this case, a shock is not created and the solution domain is divided into three sections as shown in Fig. 4.2: i) an undisturbed region on the left, ii) an undisturbed region to the right and iii) a rarefaction fan connecting these two regions. The rarefaction connects to the undisturbed region of the left at the characteristic line $x = -c_l t$, and to the undisturbed region on the right at the characteristic line $x = 2c_l t$. The analytical solution for this situation was derived by Ritter [68] and the derivation is similar to that of the wet bed case. The analytical expression for the water level and the fluid velocity is given respectively, by

$$h(x) = \begin{cases} h_l & x \le -t\sqrt{gh_l} \\ h_2 = \frac{4}{9g}\left(\sqrt{gh_l}-\frac{x}{2t}\right)^2 & -t\sqrt{gh_l} \le x \le 2t\sqrt{gh_l} \\ h_r & 2t\sqrt{gh_l} \le x \end{cases}, \tag{4.45a}$$

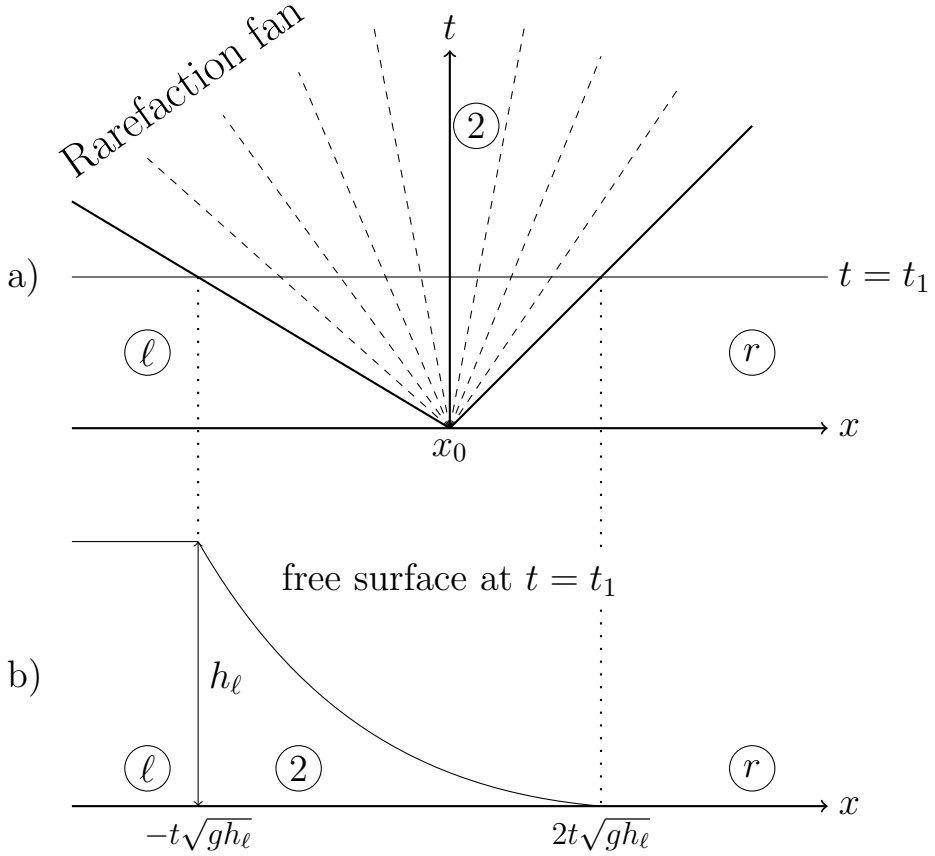


Figure 4.2: General solution to the dam break problem with a dry bed at any time $t > 0$: a) Indicates the edges of the rarefaction fan and also, the characteristic lines inside the rarefaction fan, b) Free surface (water level) within the domain at $t = t_1$. The correspondence of the characteristic profiles and the free surface at $t = t_1$ is also shown.

$$u(x) = \begin{cases} 0 & x \leq -t\sqrt{gh_l} \\ u_2 = \frac{2}{3}\left(\sqrt{gh_l} - \frac{x}{t}\right) & -t\sqrt{gh_l} \leq x \leq 2t\sqrt{gh_l} \\ 0 & 2t\sqrt{gh_l} \leq x \end{cases} . \tag{4.45b}$$

4.4 Solution methods

Due to the nonlinear nature of the SWWE, finding analytical solutions are extremely difficult and are limited to a few cases. Therefore, numerical techniques must be used to solve these equations. In the present study, a finite volume approach is selected to solve the SWWE.

Finite volume methods use the integral form of the conservation equations which is

$$\frac{d}{dt}\int_{x_1}^{x_2} q(x,t)dx + f(q(x_2,t)) - f(q(x_1,t)) = 0. \tag{4.46}$$

Using the integral form of the conservation laws ensures that the approximated solution is conservative. When the solution contains discontinuities, this feature becomes more important. In order to solve Eq. 4.46, the domain is divided into cells known as control volumes, and fluxes passing through the control volume interfaces are calculated at each time step. By integrating Eq. 4.46 from t^n to t^{n+1}, an explicit formulation for updating the solution can be obtained by

$$Q_j^{n+1} = Q_j^n - \frac{\Delta t}{\Delta x}\left(F_{j+1/2}^n - F_{j-1/2}^n\right), \tag{4.47}$$

where Q_j^n is an approximate of the quantity q at $t^n = n\Delta t$ and $F_{j+1/2}$ is the average flux crossing at $x = x_{j+1/2}$. Mathematically, this means

$$Q_j^n = \frac{1}{\Delta x}\int_{x_1}^{x_2} q(x,t^n)dx, \tag{4.48a}$$

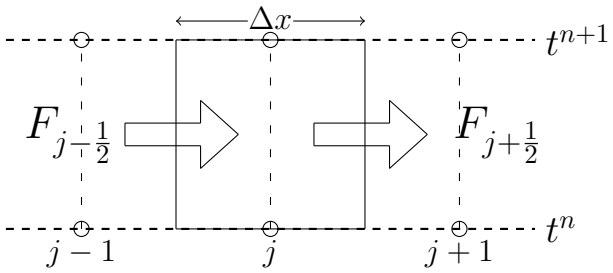


Figure 4.3: Control volume j and the corresponding fluxes

$$F^n_{j+1/2} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(q(x,t^n))dt. \tag{4.48b}$$

It can also be seen that Eq. 4.47 can be considered to be a finite difference approximation to Eq. 4.22. Different finite volume methods differ in the way the numerical flux is calculated at the interface. Figure 4.3 shows a control volume and the corresponding fluxes required to update the quantity Q_j. However, in the current investigation, implicit solutions to the governing equations are sought. Therefore, unlike Eq. 4.47 where the numerical fluxes are evaluated using values at a previous time step, the new time step is used to construct the fluxes (and potentially, the source terms). Therefore, the solution will be in the form

$$Q^{n+1}_j = Q^n_j - \frac{\Delta t}{\Delta x}\left(F^{n+1}_{j+1/2} - F^{n+1}_{j-1/2}\right). \tag{4.49}$$

4.5 Numerical Discretization

In order to numerically solve the governing equation (Eq. 4.22), the partial differential terms must be discretized. A temporal and spatial discretization scheme is required. The present section describes the selected numerical schemes used for the current investigation.

4.5.1 Spatial Discretization

In order to update the solution at each time step, the fluxes passing through at the control volume interfaces must be evaluated and used in Eq. 4.49. The central-upwind Godunov scheme is chosen to obtain the fluxes passing through the interfaces due to its accuracy at resolving the shocks and rarefaction fans seen in the dam break problem [35]. However, for comparison and to investigate the capabilities of the JFNK method, the Lax-Friedrich scheme is also considered.

Central-Upwind Godunov

This method is based on the work of [17, 45]. Although this method was originally used as an explicit method, in the present study it is converted to an implicit scheme. Some of the advantages of the method are its ability to model wet and dry bed dam-breaks and the ability to accurately resolve the shocks [35]. In addition, the underlying principle is novel and simple. The derivation is quite long and beyond the scope of this thesis. Only the main formulae are reported. For a more detailed derivation, the reader can consult [35].

The central-upwind scheme is a finite volume Godunov-type scheme [27] that does not require exact or approximate Riemann solvers or characteristic analysis which is in contrast with an upwind Godunov-type method, where an exact or approximate Riemann solver is required. This method gives higher resolution accuracy due to less numerical dissipation. To derive the numerical flux, first an average of the conservative variable at each cell is computed by

$$Q_j^{n+1}(t) = \frac{1}{\Delta x} \int_{x_j - \frac{1}{2}}^{x_j + \frac{1}{2}} q(x, t^{n+1}) dx. \tag{4.50}$$

Using these averages, a piecewise polynomial reconstruction of the solution in each cell is made as

$$\tilde{q}(x, t^{n+1}) = p_j^{n+1}(x) \quad ; \quad x_{j-\frac{1}{2}} < x < x_{j+\frac{1}{2}}. \tag{4.51}$$

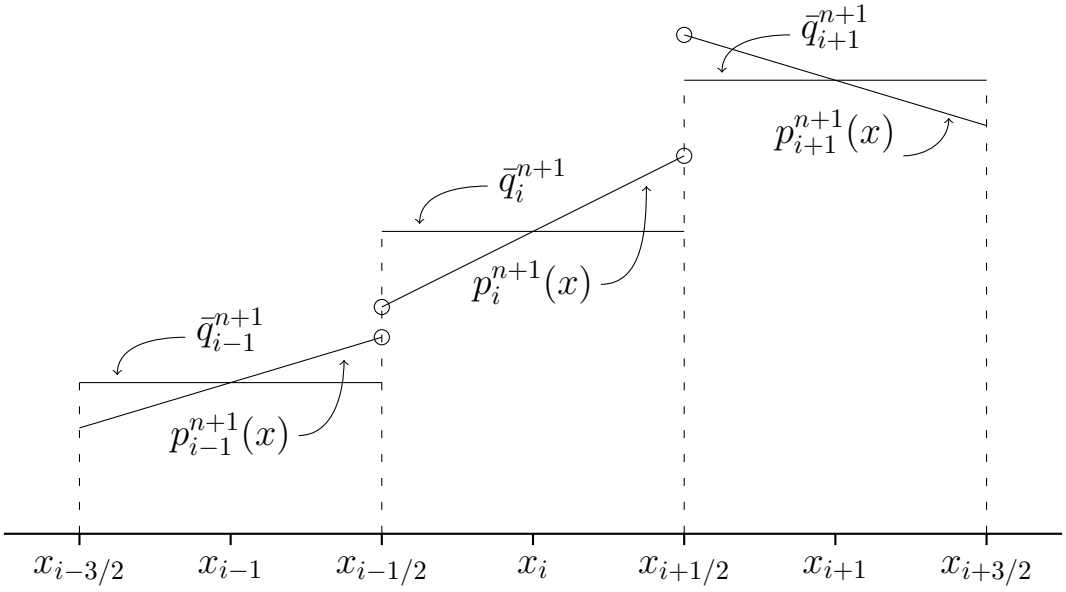


Figure 4.4: Polynomial reconstruction of the central-upwind method

The conserved variables remain continuous within a specific control volume but discontinuities of the conserved variables are allowed at cell interfaces. Figure 4.4 illustrates the mentioned concept. In Fig. 4.4, the dashed lines represent the control volume boundaries. q is the conserved variable under consideration and p is the polynomial reconstruction of q within the cell. Instead of considering q to be constant throughout the entire cell, $\bar{q}$, the profile shown by p is used. As seen in Fig. 4.4, the conserved variable is continuous within each cell but at the interfaces, the value is allowed to jump. If the conservative variables are considered to be constant along each cell, a first-order approximation to the inter-cell fluxes is obtained. But if a linear estimate is considered, a second-order accurate scheme is obtained.

Using Eq. 4.51 and Fig. 4.4, the left and right states of a cell interface are approximated, respectively by

$$q^{-}_{j+\frac{1}{2}} = Q_j^{n+1} + \frac{\Delta x}{2}\sigma_j^{n+1}, \tag{4.52}$$

$$q^{+}_{j+\frac{1}{2}} = Q^{n+1}_{j+1} + \frac{\Delta x}{2}\sigma^{n+1}_{j+1}. \tag{4.53}$$

In Eqs. 4.52 and 4.53, σ^{n+1}_{j} is the slope of the jth cell. If the slope is equal to zero, the method is first-order and if it is not zero it is second-order accurate. The slope can be calculated using many limiters. For the current investigation, the "*minmod*" limiter is selected. Therefore, at each cell the slope is calculated by

$$\sigma^{n+1}_{j} = minmod\left(\frac{Q^{n+1}_{j} - Q^{n+1}_{j-1}}{\Delta x}, \frac{Q^{n+1}_{j+1} - Q^{n+1}_{j}}{\Delta x}\right), \tag{4.54}$$

where the *minmod* function is defined by

$$minmod(a,b) = \begin{cases} a & if\ |a| < |b| \quad and \quad ab > 0 \\ b & if\ |b| < |a| \quad and \quad ab > 0 \\ 0 & if\ ab \leqslant 0 \end{cases}. \tag{4.55}$$

Using the limiter from Eq. 4.55, the slope of a cell is reduced by a great amount near discontinuities. At each discontinuity, a Rarefaction fan exists. In order to deal with the Rarefaction fans, unlike upwind Godunov methods where a Riemann problem needs to be solved exactly or approximately, here, the left and right speeds of the Riemann fans are specified to split the domain into smaller regions with varying size which will contain the entire Riemann fan. Using this method, a Reimann solver is not necessary and a fully discretized Godunov type central-upwind scheme is obtained. The numerical flux for the central-upwind Godunov method can be calculated by

$$\begin{aligned} F_{j+\frac{1}{2}} = & \frac{a^{+}_{j+\frac{1}{2}} f(q^{-}_{j+\frac{1}{2}}) - a^{-}_{j+\frac{1}{2}} f(q^{+}_{j+\frac{1}{2}})}{a^{+}_{j+\frac{1}{2}} - a^{-}_{j+\frac{1}{2}}} \\ & + \frac{a^{+}_{j+\frac{1}{2}} a^{-}_{j+\frac{1}{2}}}{a^{+}_{j+\frac{1}{2}} - a^{-}_{j+\frac{1}{2}}} \left[q^{+}_{j+\frac{1}{2}} - q^{-}_{j+\frac{1}{2}}\right], \end{aligned} \tag{4.56}$$

where $a^+_{j+\frac{1}{2}}$ and $a^-_{j+\frac{1}{2}}$ are local speeds of the Riemann fan evaluated at each interface by

$$a^+_{j+\frac{1}{2}} = max\left\{\lambda_2(q^-_{j+\frac{1}{2}}), \lambda_2(q^+_{j+\frac{1}{2}}), 0)\right\}, \tag{4.57}$$

$$a^-_{j+\frac{1}{2}} = min\left\{\lambda_1(q^-_{j+\frac{1}{2}}), \lambda_1(q^+_{j+\frac{1}{2}}), 0)\right\}. \tag{4.58}$$

Using Eq. 4.56 for the numerical flux, a second-order scheme is obtained. This method also satisfies the TVD condition [29].

Lax-Friedrich

The Lax-Friedrich scheme is a classical first order scheme used for the evaluation of the interface fluxes in the finite volume method. As mentioned in the Section 4.4, finite volume methods differ in the way the numerical flux is calculated. In the Lax-Friedrich method the numerical flux is calculated by

$$F_{j+\frac{1}{2}} = \frac{1}{2}\left(F_{j+1} - F_j\right) - \frac{\Delta x}{2\Delta t}(Q_{j+1} - Q_j). \tag{4.59}$$

A common approach is to calculate the terms in Eq. 4.59 using values from the previous time step (explicit method). However, in the present study, the terms are calculated using values from the next time step which correspond to an implicit formulation of the Lax-Friedrich scheme.

4.5.2 Temporal Discretization

The time derivative in Eq. 4.22 needs to be discretized in order to numerically solve the governing equations. As the main goal of the present study is to investigate the capabilities of the JFNK method, a simple backward Euler scheme is used for the temporal discretization scheme. However, the accuracy of the results can be increased with the use of a higher order temporal discretization scheme. For comparison, the Crank-Nicolson scheme

is also investigated. In order to demonstrate the formulation of the backward Euler and Crank-Nicolson schemes, a simple example is given below. Consider the following PDE

$$\frac{\partial u}{\partial t} = F\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right). \tag{4.60}$$

The discretized form of the equation can be written as

$$\frac{u^{n+1} - u^n}{\Delta t} = (1-\theta)F^n\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right) + \theta F^{n+1}\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right). \tag{4.61}$$

Now, with a choice of $\theta = 0$, the explicit forward Euler scheme is obtained. With a choice of $\theta = 1$ the fully implicit backward Euler scheme is obtained and with a choice of $\theta = \frac{1}{2}$ the implicit Crank-Nicolson scheme is obtained. As can be seen in Eq. 4.61, the Crank-Nicolson scheme requires information from both the previous time step and the current time step, leading to a second-order accurate scheme in time, while the backward Euler scheme only requires information for the next time step and is a first-order scheme.

4.6 Numerical Details

4.6.1 Computational Domain

The computational domain consists of discrete points (nodes) or finite volumes (control volumes). Both variables, h and uh, are stored at the center of these control volumes or at the nodes. Thus, the computational grid is collocated [89]. In the present study, the finite volume method will be used and therefore, the computational domain will be split into control volumes. Consequently, N_x represents the number of control volumes in use and $2N_x$ variables involved. Figure 4.5 shows the computational domain used. The highlighted control volumes in Fig. 4.5 (labeled as GC_k) are "*ghost cells*" used to implement the boundary conditions. Further information will be provided in Sec. 4.6.2.

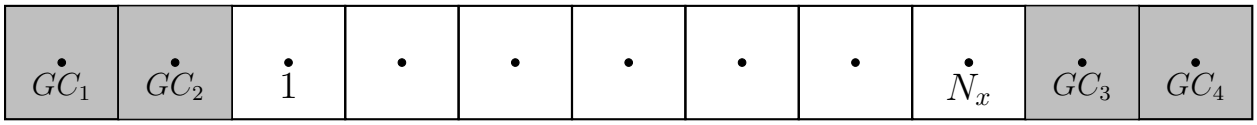


Figure 4.5: Schematic of computational domain

4.6.2 Boundary and Initial Conditions

Boundary conditions need to be defined to close the system of equations. The boundary conditions are as follows. At the left wall,

$$\left(\frac{\partial h}{\partial x}\right)_{left\ wall} = 0 \tag{4.62}$$

$$(uh)_{left\ wall} = 0 \tag{4.63}$$

Equations 4.62 and 4.63 imply that when the wave reaches the left wall, it will bounce back off the wall (zero velocity at the wall and a zero gradient condition for the water level at the wall). They are also referred to as reflective boundary conditions. The same boundary conditions are applied to the right wall. Thus,

$$\left(\frac{\partial h}{\partial x}\right)_{right\ wall} = 0 \tag{4.64}$$

$$(uh)_{right\ wall} = 0 \tag{4.65}$$

In order to implement the boundary conditions, ghost cells are placed at the boundaries. Due to the second order nature of the central-upwind scheme, each node requires information from the two neighboring cells on each side. Therefore, two ghost cells are put on each side of the computational domain. This means that at the beginning of each Newton loop, the values at the boundaries are set to the following values

$$(uh)_{GC_1} = (uh)_{GC_2} = (uh)_1 \tag{4.66}$$

$$(uh)_{GC_4} = (uh)_{GC_3} = (uh)_{N_x} \tag{4.67}$$

$$h_{GC_1} = h_{GC_2} = h_1 \tag{4.68}$$

$$h_{GC_4} = h_{GC_3} = h_{N_x} \tag{4.69}$$

Equations 4.66 and 4.67 are used to define that the velocity at the wall is zero and Eqs. 4.68 and 4.69 are used to satisfy Eqs. 4.62 and 4.64, respectively. When the Lax-Friedrich scheme is used, only one ghost cell is required at each boundary due to its first-order nature.

In order to be able to compare the results to previous numerical work, three different flow regimes are considered: sub critical, supercritical and dry bed flows. All three flow regimes have an initial velocity of $u = 0$ (m/s) throughout the entire domain and the water level on the left hand side of the dam is set to $h_l = 10$ (m) and they only differ in the initial water height at the right hand side of the dam. For the sub critical flow regime, the water to the right is set to $h_r = 5$ (m) and the Froude number of the flow remains beneath 1 at all times during the simulation. If the water level is set to $h_r = 0.1$ (m), the flow goes through a transition to supercritical flow after a while due to the fact the Froude number goes higher than 1. For the dry bed regime, there is no water on the right hand side of the dam. In order to prevent numerical errors, the water level is set to a very small value. Here, $h = 0.0001$ (m) is used.

4.7 Results

The results obtained from several simulations are presented. Three different methods are used:

- An explicit scheme used to solve the governing equations which will be referred to as the explicit method.

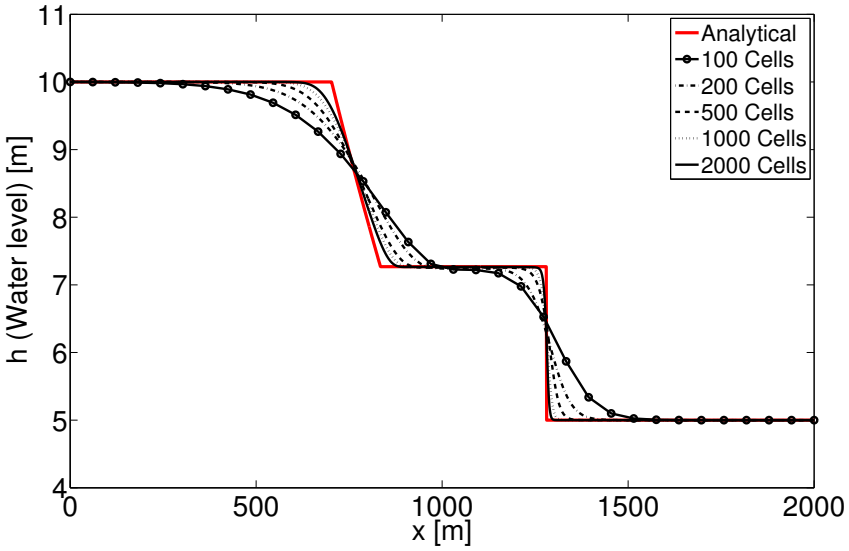


Figure 4.6: Grid convergence study for the 1D dam break problem

- A fully implicit discretization which leads to a system of equations and is then solved by Newton's method. The Jacobian is not calculated explicitly and is instead approximated. This method will be referred to as the JFNK method.
- The same discretization scheme used in the JFNK method, but with the full calculation of the Jacobian matrix. This will be referred to as the NK method.

First a grid analysis study is performed to obtain a suitable grid size for the simulations. Results obtained from the JFNK and NK method are compared to those from the explicit scheme. The capability to handle high CFL numbers with the JFNK method is investigated and the results obtained from different numerical discretization schemes are also compared.

4.7.1 Grid Sensitivity Analysis

Figure 4.6 shows the numerical predictions obtained for the 1D dam break problem using multiple grid sizes. The results are obtained by using the JFNK method and a CFL value equal to 1. The predictions are shown at $t = 30$ (s) and by considering the sub critical

Table 4.1: Computational time required for different grid sizes

Grid Size	100	200	500	1000	2000
CPU Time (s)	3.7	12	63	264	1150

flow regime. The central-upwind discretization scheme is used for the numerical fluxes and a backward Euler scheme is used for the temporal discretization ($\theta = 1$ in Eq. 4.61). It can be seen that by increasing the number of cells, the predictions converge towards the analytical solution and both the hydraulic jump and the rarefaction fan are captured more accurately.

Table 4.1 shows the computational time required for each simulation in seconds. It is seen that as the number of cells is doubled, the computational time approximately increases by a factor of 4. As can be seen in Fig. 4.6, only small differences, in the order of 0.5 - 1 percent, can be noticed between the results obtained using 1000 cells and 2000 cells. Therefore, for the remainder of the current section, 1000 grid points is selected.

4.7.2 Comparison Between JFNK, NK and Explicit Methods

In order to compare the results obtained from the JFNK and NK method with those from the explicit scheme, all methods are run using the central-upwind scheme and a backward Euler temporal discretization with a constant CFL number equal to 0.1. This value is selected to prevent the explicit scheme from diverging. All solutions are shown at $t = 30$ (s). Similar trends are obtained at any other time.

Figures 4.7, 4.8 and 4.9 present the numerical predictions obtained for the sub critical, supercritical and dry bed flow regimes, respectively. No significant difference can be noticed between the results and the solutions almost exactly overlap. Table 4.2 shows the CPU time required for each simulation. As expected, the explicit scheme is much faster when the same CFL numbers are used for both methods. This is due to the large system of equations which needs to be solved in the JFNK and NK methods compared to algebraic equations which are used to update the unknowns in the explicit scheme. It can also be seen that by solving the equations using the JFNK method, almost 90% computational

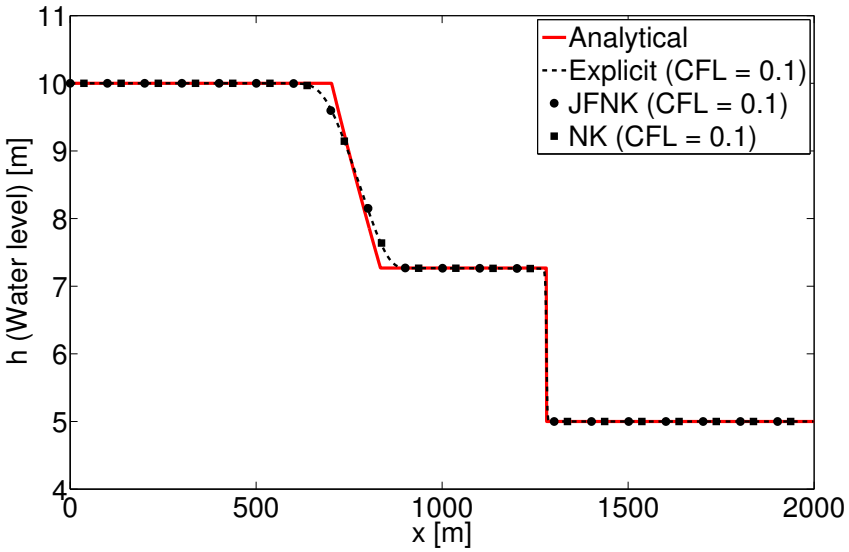


Figure 4.7: Comparison of numerical predictions obtained from JFNK, NK and explicit schemes for the sub critical flow regime

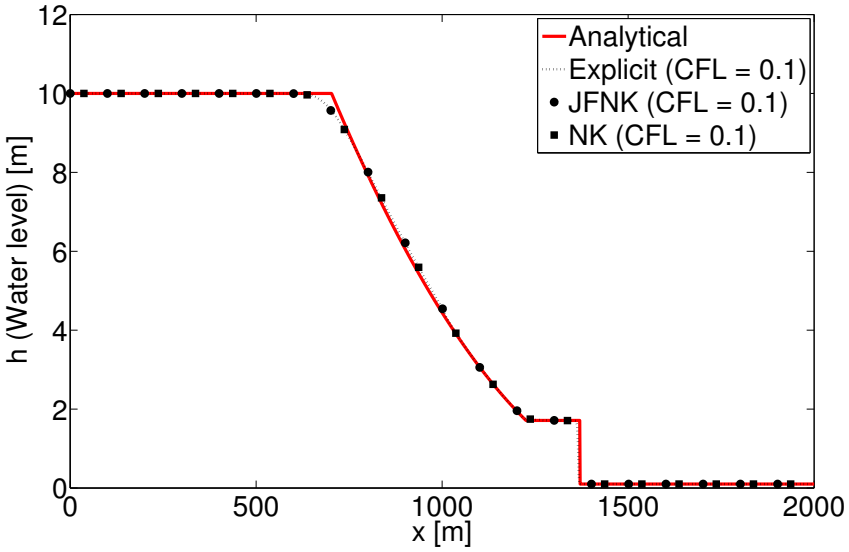


Figure 4.8: Comparison of numerical predictions obtained from JFNK, NK and explicit schemes for the supercritical flow regime

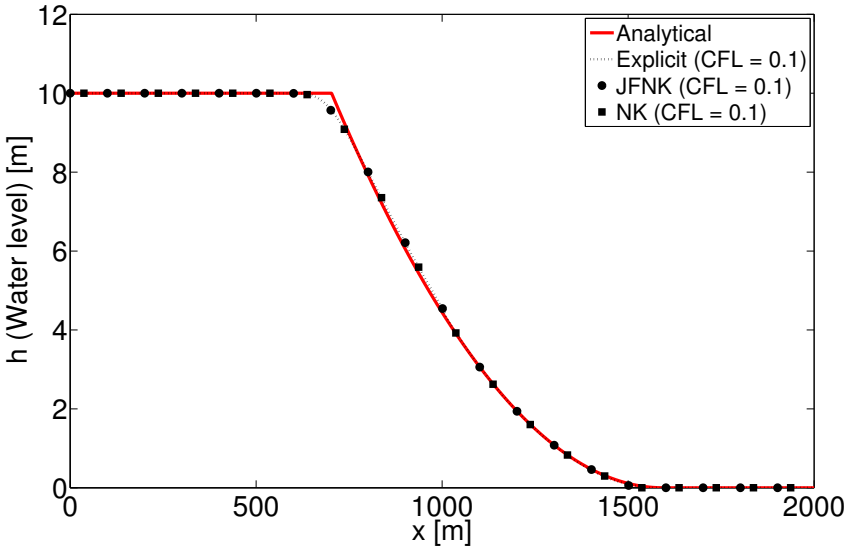


Figure 4.9: Comparison of numerical predictions obtained from JFNK, NK and explicit schemes for the dry bed flow regime

time can be saved. This is due to the large Jacobian matrix which needs to be calculated in the NK method.

4.7.3 CFL Range

Due to the implicit nature of the JFNK method, it is expected that the method can handle higher CFL values compared to the explicit method. The predictions should still remain stable to be acceptable. The simulations used in this section are obtained by using the central-upwind scheme with a backward Euler discretization and are shown at $t = 30\ (s)$. First the CFL numbers which can be used by the explicit scheme are investigated. It is a well known fact that the time step permitted in explicit schemes are limited. The CFL number should be smaller than 1, but this condition is not sufficient. For the 1D dam break problem at hand, using the current discretization scheme, it is found that for all flow regimes (sub critical, supercritical and dry bed) a CFL number higher than 0.1 either lead to divergence or produce instabilities in the results. For the sub critical regime, a

Table 4.2: Comparison of computational time in seconds of the JFNK, NK and explicit methods with CFL=0.1 in different flow regimes using the central-upwind Godunov scheme

	Computational Time		
	Sub critical	Supercritical	Dry bed
Explicit	23	29	34
NK	4654	6229	5811
JFNK	558	703	778

CFL value of 0.5 results to a converged solution but is highly oscillatory. This prediction is shown in Fig. 4.10. It can be seen that the oscillation is caused before the shock and in the constant level region.

Figure 4.11 shows the range of CFL numbers used for the simulations under the sub critical regime. As the time step is increased (higher CFL number), the accuracy of the results decreases which is an expected outcome due to the fact that a transient problem is being solved. For the sub critical regime, high CFL numbers can be used to reach a stable converged solution. In comparison to the explicit method, CFL values up to 400 times larger may be used. Figures 4.12 and 4.13 present the CFL values used for the supercritical and dry bed flow regimes, respectively. Higher CFL values could not be used and would cause the outer Newton iteration to diverge and enter an infinite loop. Many attempts were made in order to remedy this issue such as changing the initial guess at each Newton iteration and changing the perturbation value used in the Jacobian matrix approximation, but no solution was found. Even with the current state of the simulations, larger time steps (compared to the explicit method) would lead to a stable converged solution using JFNK. Another interesting observation is that for the sub critical simulation (Fig. 4.11), changing the CFL number from 1 to 4 results to a relatively larger decrease in accuracy (almost 3%) whereas this is not the case with the supercritical flow regime. A change in the CFL number from 0.5 to 3 does not produce a significant difference in the results (less than 1%). This is most likely due to the size of the hydraulic jump (shock) which is larger in the sub critical flow allowing more tolerance for both the GMRES and Newton iterations.

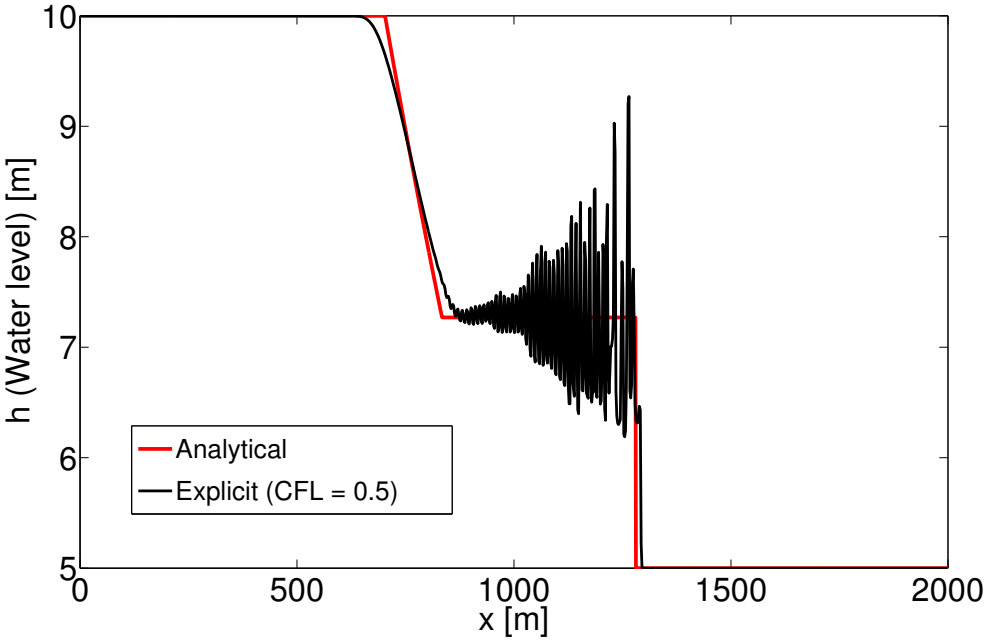


Figure 4.10: Instabilites when using the explicit method and $CFL = 0.5$

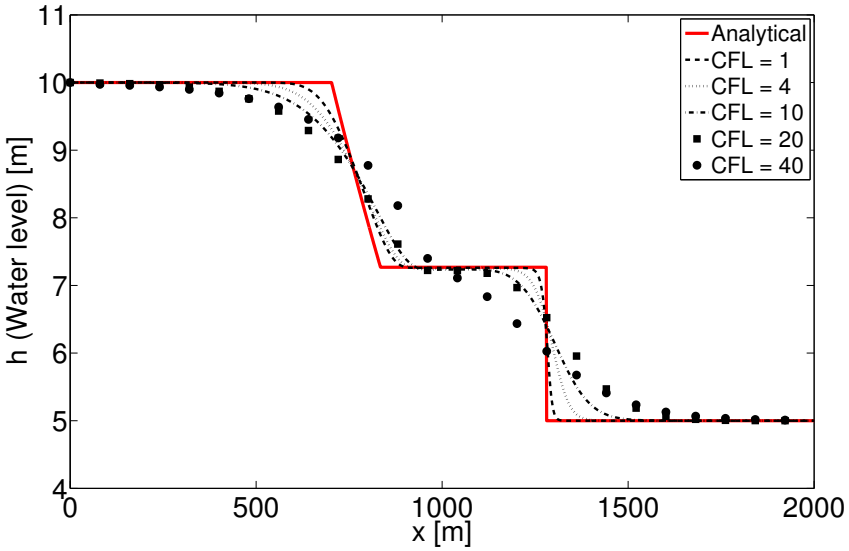


Figure 4.11: CFL range used for 1D dam break problem with JFNK, sub critical regime

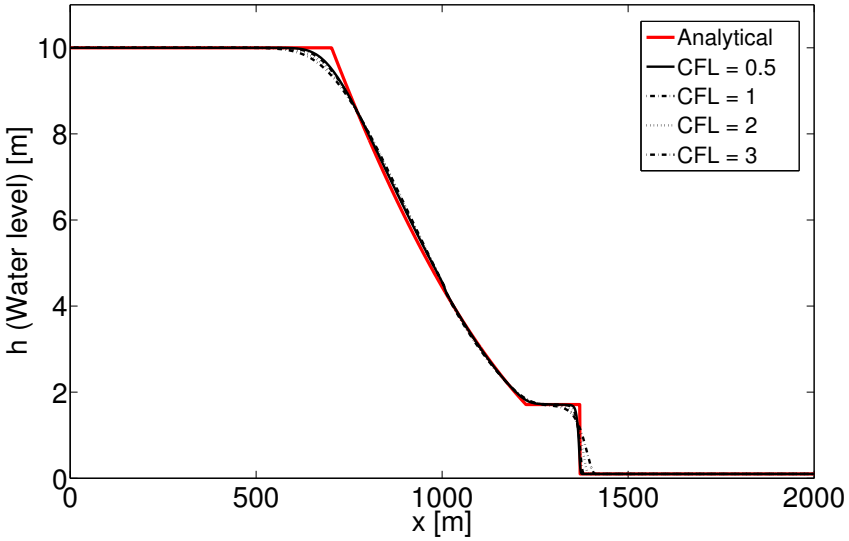


Figure 4.12: CFL range used for 1D dam break problem with JFNK, supercritical regime

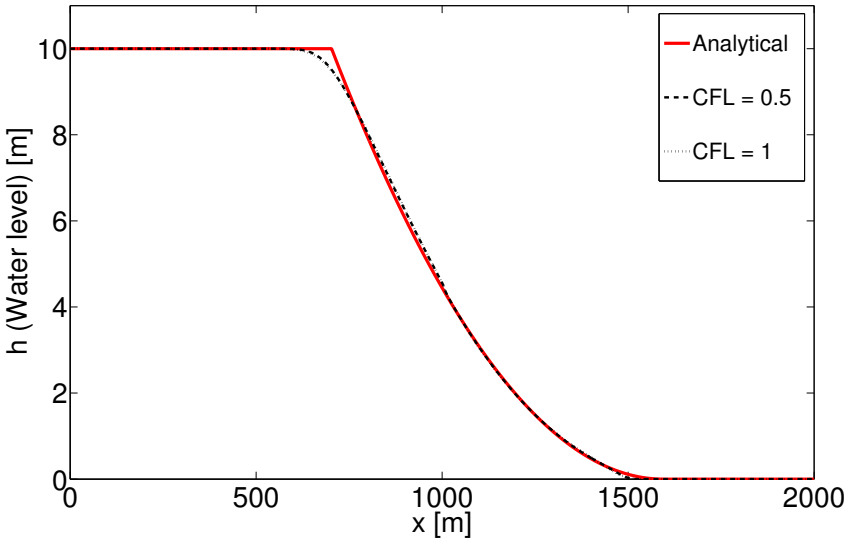


Figure 4.13: CFL range used for 1D dam break problem with JFNK, dry bed regime

4.7.4 Effect of Numerical Discretization

Different discretization schemes are tested to identify their impact on the results. Two different discretization schemes for the numerical fluxes are selected, the first order Lax-Friedrich scheme (Eq. 4.59) and the second order central-upwind scheme (Eq. 4.56). Two temporal schemes are also implemented, a first order backward Euler scheme ($\theta = 1$ in Eq. 4.61) and a second order Crank-Nicolson scheme ($\theta = 1/2$ in Eq 4.61). Multiple combinations of these schemes are investigated. The results are considered at $t = 30$ (s) and are labeled as follows: Central-Upwind Godunov (CUG), Lax-Friedrich (LF), Backward Euler (BE) and Crank-Nicolson (CN).

Figure 4.14 presents the numerical predictions obtained from the JFNK method using different discretization schemes. The main difference between the predictions is around the shock (hydraulic jump). As expected, with the same temporal scheme, the central-upwind Godunov scheme captures the shock more accurately. However, when the second order Crank-Nicolson scheme is used, oscillations occur near the shock which is also an expected outcome, due to the higher order of accuracy. Oscillations near the shock, have a higher peak value with CUG. When a smaller CFL value is used, CFL = 0.5, oscillations do not appear near the shock for any of the discretization schemes (results are not shown for brevity). Figure 4.15 shows the numerical predictions obtained for the supercritical flow regime. Similar trends are observed, except that the CUG-CN combination does not produce numerical oscillations and resolves the shock accurately. The numerical results obtained for the dry bed test case are presented in Fig. 4.16. No significant difference can be observed by the different methods and they all converge to the same solution.

4.8 Summary

In the present chapter the JFNK is successfully applied to a nonlinear system of equations, the SWWE, which describe the motion of a single phase flow. The 1D dam break problem is chosen as the numerical benchmark for the simulations. Numerical results obtained from the JFNK method are compared to results from a fully explicit scheme and also a fully implicit scheme with the explicit calculation of a numerical Jacobian matrix (the

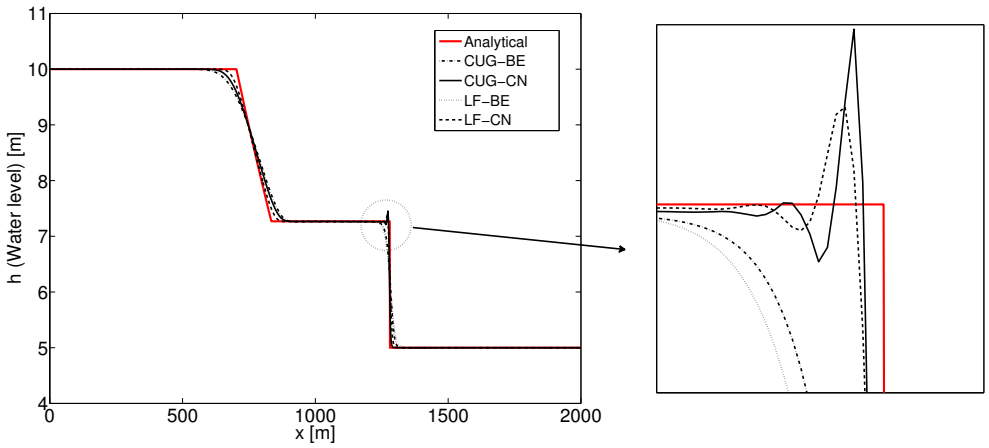


Figure 4.14: Comparison of different numerical discretization schemes in sub critical flow regime, CFl = 1

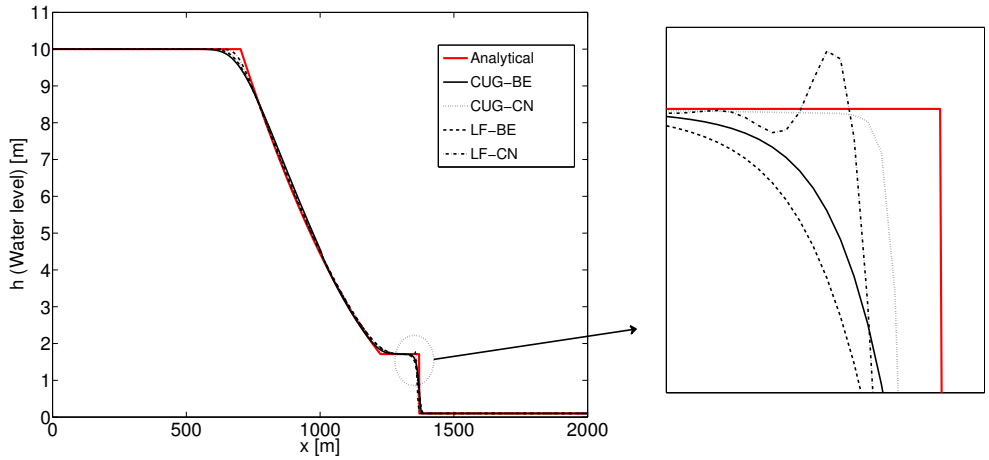


Figure 4.15: Comparison of different numerical discretization schemes in supercritical flow regime, CFL = 1

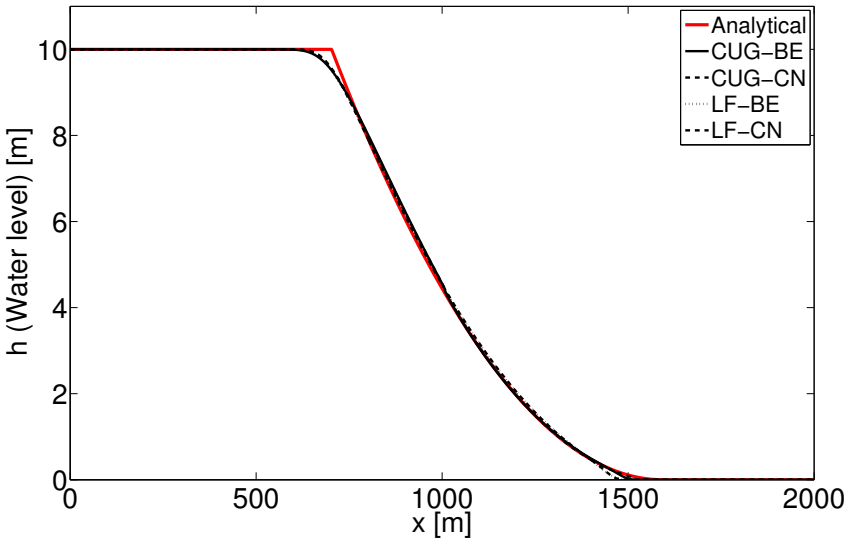


Figure 4.16: Comparison of different numerical discretization schemes in dry bed flow regime, CFL = 0.5

NK method). The results are shown to be consistent throughout all methods and are in agreement with previous numerical results (obtained by using explicit schemes) and as expected, the implicit methods (NK and JFNK) are capable of handling relatively higher CFL values resulting to larger time-steps. The computational time required for the simulations are decreased by almost 90% when the JFNK method is used instead of the NK method. Moving forward, in the following chapter, the JFNK method is applied to another set of governing equations describing the motion of two-phase flows, namely the two-fluid model, in order to perform a similar investigation and identify the capabilities of JFNK and in hope to obtain similar gains in CPU time.

Chapter 5

Application of JFNK to Isentropic Two-Fluid Model

In this section, the JFNK method is applied to the isentropic two-fluid model where the conservation of mass and momentum equations are solved for each phase. A backward Euler scheme is used for the temporal discretization and the AUSM+ numerical fluxes are applied for the convective fluxes. The source terms are discretized using a central scheme. For comparison, one explicit and two implicit formulations, one with Newton's solver and with the Jacobian matrix and one with JFNK, are implemented. A detailed grid and model parameter sensitivity analysis is performed.

5.1 Isentropic Two-Fluid Model

For this part part of the study, the four equation two fluid model described in Sec. 3.3.2 is considered. The source terms used are also given from Sec. 3.4. The initial conditions and the boundary conditions are not explicitly expressed for the conserved variables. Instead, they are given in terms of primitive variables such as pressure, velocity and the void fraction. Further, throughout the discretization steps shown in Section 5.2.1, these primitive variables are needed. However, after each time step the conserved variables are calculated.

Therefore, there is a need to convert the conserved variables to primitive variables. The vector of primitive variables is defined as

$$V = \begin{pmatrix} \alpha_g \\ p \\ u_g \\ u_l \end{pmatrix}. \tag{5.1}$$

The velocities of both phases are directly calculated from the conserved variables using

$$u_g = \frac{U_3}{U_1}, \tag{5.2a}$$

$$u_l = \frac{U_4}{U_2}. \tag{5.2b}$$

U_i is the i^{th} component of the conserved vector shown in Eq. 3.29. The two other primitive variables cannot be evaluated directly and require further calculations. Using the definition of U_1 and U_2, and also keeping in mind that $\alpha_g + \alpha_l = 1$, one obtains

$$\alpha_g \rho_g(p) = U_1, \tag{5.3a}$$

$$(1 - \alpha_g)\rho_l(p) = U_2. \tag{5.3b}$$

Eliminating α_g in Eq. 5.3b by using Eq. 5.3a, a non-linear function of p is obtained. Due to the single pressure assumption, solving $F(p) = 0$ leads to the value of the pressure. Here,

$$F(p) = \left[1 - \frac{\alpha_g \rho_g(p)}{\rho_g(p)}\right] \rho_l(p) - (1 - \alpha_g)\rho_l(p). \tag{5.4}$$

This non-linear equation can be solved using the Newton-Raphson method. This procedure has to be performed at each computational node at every time step.

5.2 Numerical Setup

Here, the complete discretization used for the four equation two-fluid model is reviewed. A fully-implicit discretization scheme is applied to the four equation two-fluid model presented in Section **??**. A simple first order backward Euler scheme is selected for the time discretization and the AUSM+ scheme [52] is implemented for the discretization of the convective flux terms. Thus, the resulting discretized form of Eq. 3.24 is

$$U_i^{n+1} + \frac{\Delta t}{\Delta x}\left[E^\star(U_i^{n+1}, U_{i+1}^{n+1}) - E^\star(U_{i-1}^{n+1}, U_i^{n+1})\right] - \Delta t\left[(C^{nv})_i^{n+1} + S_i^{n+1}\right] = U_i^n. \quad (5.5)$$

The subscript i denotes the spatial location of the computational node and the superscript denotes the time step, with $n+1$ being the current time level. Δx is the spatial grid spacing and is constant throughout the computational grid (uniform meshes are used). Δt is the time step chosen and may vary throughout the simulations. It is calculated as a function of the CFL number, following [63]

$$\Delta t = CFL \times \min_i \left[\frac{(1-\alpha_g)\Delta x}{|(u_l)_i| + (a_l)_i} + \frac{\alpha_g \Delta x}{|(u_g)_i| + (a_g)_i}\right]. \quad (5.6)$$

5.2.1 Numerical Fluxes: AUSM+

In Eq. 5.5, $E^\star(U_L, U_R)$ is the numerical flux which is calculated at the cell faces. In order to calculate the flux at the faces, the values of the flux function at the two adjacent nodes are used. L and R are the left and right nodes to the cell face, respectively.
The AUSM family numerical flux has been successfully applied to multiphase flow [38]. The AUSM+ scheme was originally developed by Liou [52] and has been extended to two-fluid models by Paillere et al. [63]. Different accuracy order schemes have been developed within the AUSM family. As a first step to the current implementation of a fully implicit scheme for multiphase flows using the JFNK method, the AUSM+ scheme is selected which is first-order accurate. However, AUSM+ has been shown to perform well in previous studies [63, 69] for the same two benchmarks selected in the current investigation that will be used for comparison. The main idea behind the scheme is to divide the numerical flux into two terms, a convective and a pressure term. Therefore, the flux for each phase can be written

as

$$E_k = \dot{m}_k \underbrace{\begin{pmatrix} 1 \\ u_k \end{pmatrix}}_{\psi_k} + \underbrace{\begin{pmatrix} 0 \\ \alpha_k p \end{pmatrix}}_{P_k}, \tag{5.7}$$

where $\dot{m}_k$ is the mass flux. The convected vector is denoted by ψ_k and the pressure vector by P_k. Left and right Mach numbers need to be calculated at each interface using a numerical speed of sound defined as

$$a_k^\star = \sqrt{(a_k)_L (a_k)_R}, \tag{5.8}$$

where a_k is the speed of sound for phase k and $a_k^\star$ is the numerical speed of sound at the interface between two adjacent nodes. Numerical Mach numbers are evaluated by

$$(M_k)_L = \frac{(u_k)_L}{a_k^\star} \quad , \quad (M_k)_R = \frac{(u_k)_R}{a_k^\star}. \tag{5.9}$$

The numerical Mach number at the interface is calculated using values obtained from Eq. 5.9 and a polynomial function proposed in [52] following

$$M_k^\star = \mathscr{M}^+((M_k)_L) + \mathscr{M}^-((M_k)_R). \tag{5.10}$$

The polynomial function splitting the numerical Mach numbers must satisfy numerous properties [52] such as consistency, meaning $\mathscr{M}^+(M) + \mathscr{M}^-(M) = M$ (M being the mach number used), symmetry ($\mathscr{M}^+(M) = -\mathscr{M}^-(-M)$) and it should also be continuously differentiable. In addition, $\mathscr{M}^\pm$ need to be monotone increasing functions of M. This leads to the following split functions (with a suitable choice of B):

$$\mathscr{M}^\pm(M) = \begin{cases} \frac{1}{2}(M \pm |M|) & \text{if } |M| \geq 1 \\ \mathscr{M}_B^\pm(M) & \text{otherwise} \end{cases}, \tag{5.11a}$$

$$\mathscr{M}_B^\pm(M) = \pm\frac{1}{2}(M \pm 1)^2 \pm B(M^2 - 1)^2, \tag{5.11b}$$

where B is a constant. Using the previously calculated numerical Mach number (Eq. 5.10) and the speed of sound at the interface, Eq. 5.8, the mass flux at the interfaces are calculated by

$$\dot{m}_k{}^\star = a_k^\star \left((\alpha_k \rho_k)_L \frac{M_k^\star + |M_k^\star|}{2} + (\alpha_k \rho_k)_R \frac{M_k^\star - |M_k^\star|}{2} \right). \tag{5.12}$$

In order to evaluate the pressure terms at the interfaces, another split function is required. In addition to consistency, symmetry and differentiability, the splitting function must be selected such that negative a pressure does not appear in the solution leading to $\mathscr{P}^{\pm}(M) \geq 0$. The original splitting function proposed by [52] is applied:

$$\mathscr{P}^{\pm}(M) = \begin{cases} \frac{1}{2}(1 \pm sign(M)) & \text{if } |M| \geq 1 \\ \mathscr{P}^{\pm}_{A}(M) & \text{otherwise} \end{cases}, \tag{5.13a}$$

$$\mathscr{P}^{\pm}_{A}(M) = \frac{1}{4}(M \pm 1)^2(2 \mp M) \pm AM(M^2 - 1)^2. \tag{5.13b}$$

To satisfy all of the properties mentioned above for both split functions, the choice of $A = 3/16$ and $B = 1/8$ is taken. The fluxes of the AUSM+ scheme shown in Eq. 5.7 are calculated at cell centers, but in Eq. 5.5 the numerical fluxes at the interfaces are required and are calculated by [63]

$$E^{\star}_{k}(U_L, U, R) = \dot{m}_k{}^{\star}\psi^{\star}_{k} + P^{\star}_{k}. \tag{5.14}$$

The convected terms at the cell faces are calculated using an upwind approach depending on the direction of the flow. Therefore,

$$\dot{m}_k{}^{\star}\psi^{\star}_{k} = \frac{1}{2}\dot{m}_k{}^{\star}[\psi_k(U_L) + \psi_k(U_R)] + \frac{1}{2}|\dot{m}_k{}^{\star}|[\psi_k(U_L) - \psi_k(U_R)]|. \tag{5.15}$$

The pressure at the interface must also be evaluated using the the pressure at adjacent nodes and the split function, eq.5.13 as follows

$$P^{\star}_{k} = \mathscr{P}^{+}((M_k)_L)(P_k)_L + \mathscr{P}^{-}((M_k)_R)(P_k)_R \tag{5.16}$$

Due to the fact that only values at adjacent nodes to each interface are used to calculate the numerical fluxes at that interface, the AUSM+ scheme used is first-order accurate.

5.2.2 Discretization of Source Terms

In Eq. 5.5, the two source term vectors need to be discretized. In C^{nv} as shown in Eq. 3.29, the spatial first derivatives of the void fraction are determined using a second-order

central scheme and all other terms are calculated at the time level $n+1$ due to the a fully implicit scheme leading to the following discretized terms,

$$\left(p\frac{\partial \alpha_k}{\partial x}\right)_i^{n+1} = p_i^{n+1}\frac{(\alpha_k)_{i+1}^{n+1} - (\alpha_k)_{i-1}^{n+1}}{2\Delta x}, \tag{5.17a}$$

$$(F_k^{nv})_i^{n+1} = \left(-\sigma\frac{(\alpha_g\rho_g\alpha_l\rho_l)_i^{n+1}}{(\alpha_g\rho_l)_i^{n+1} + (\alpha_l\rho_g)_i^{n+1}}\left((u_g)_i^{n+1} - (u_l)_i^{n+1}\right)^2\right)\frac{(\alpha_k)_{i+1}^{n+1} - (\alpha_k)_{i-1}^{n+1}}{2\Delta x}, \tag{5.17b}$$

$$(\alpha_k\rho_k g_x)_i^{n+1} = (\alpha_k)_i^{n+1}(\rho_k)_i^{n+1}(g_x)_i^{n+1}, \tag{5.17c}$$

$$\left(F_g^D\right)_i^{n+1} = -C_f(\alpha_g)_i^{n+1}\left(1 - (\alpha_g)_i^{n+1}\right)(\rho_g)_i^{n+1}\left((u_g)_i^{n+1} - (u_l)_i^{n+1}\right), \tag{5.17d}$$

$$\left(F_l^D\right)_i^{n+1} = -\left(F_g^D\right)_i^{n+1}. \tag{5.17e}$$

5.2.3 Treatment for Low Mach Number Flows

In the low Mach number limit, the AUSM+ scheme might cause oscillations due to odd-even decoupling. This issue is related to the liquid phase where high speeds of sound lead to very small Mach numbers. To resolve this issue, modifications can be made to the AUSM+ scheme, as described in [16, 54]. In the present study, the method developed by [23] and used by [63, 69] is implemented. Thus, the pressure and velocity fields are coupled by modifying the liquid mass flux. The addition of a term known as the pressure diffusion term to the original liquid mass flux (Eq. 5.12) will add enough dissipation when necessary and hence, control the oscillations. First, the numerical speed of sound defined by Eq. 5.8 is modified by

$$(a_l^\star)_m = f^\star\left(0.5\left[\frac{(u_l)_L}{(a_l)_L} + \frac{(u_l)_R}{(a_l)_R}\right]\right)a_k^\star, \tag{5.18}$$

where the additional subscript m denotes 'modified' and $f^\star$ is a scaling function defined as [63]

$$f^\star(M) = \frac{\sqrt{(1-M_o^2)^2 M^2 + 4M_o^2}}{1+M_o^2}, \tag{5.19}$$

with M_o being a "cut-off" Mach number and taking a value between 0 and 1. This parameter requires some tuning for each test case and is therefore, one of the drawbacks of the current method. After the modification of the speed of sound, left and right numerical Mach numbers, initially defined by Eq. 5.9, also need to be corrected with

$$(M_k)_{L,m} = \frac{(u_k)_L}{(a_l^\star)_m} \quad , \quad (M_k)_{R,m} = \frac{(u_l)_R}{(a_l^\star)_m} \tag{5.20}$$

Next, similar to Eq. 5.10, a numerical Mach number is calculated at the interface using

$$(M_l^\star)_m = \mathscr{M}^+((M_k)_{L,m}) - \mathscr{M}_1^+((M_k)_{L,m}) - \mathscr{M}^-((M_k)_{R,m}) + \mathscr{M}_1^-((M_k)_{R,m}). \tag{5.21}$$

The pressure diffusion term is then calculated by [54]

$$\dot{m}_p = 0.5\,(a_l^\star)_m \left(\frac{1}{M_o^2} - 1\right)(M_l^\star)_m \frac{(\alpha_l)_L p_L - (\alpha_l)_R p_R}{(a_l^\star)^2}, \tag{5.22}$$

and is added to the original liquid mass flux

$$(\dot{m_l^\star})_m = \dot{m_l^\star} + \dot{m}_p. \tag{5.23}$$

The mass flux obtained by Eq. 5.23 is then used in Eq. 5.15 for the conservation of mass and momentum of the liquid phase. The pressure diffusion term scales the amount of dissipation when the liquid Mach number is small. It can be seen from Eq. 5.22, a value of $M_o = 1$ is equivalent to not modifying the liquid mass flux.

5.2.4 Phase appearance and disappearance

In cases where the the void fraction tends to 0 or 1, numerical difficulties might arise. This can be seen when one converts the conserved variables to primitive variables in Eqs. 5.2a and 5.2b where the velocities are obtained from the division of two very small numbers.

This issue shows itself in the form of pressure spikes at the interface between the two phases. The velocity of the vanishing phase should be very close to the velocity of the remaining phase. Therefore, instead of calculating the velocities using Eqs. 5.2a and 5.2b, a correction function is applied in the following form

$$u_g = G(\alpha_g)\frac{U_3}{U_1} + (1 - G(\alpha_g))\frac{U_4}{U_2}, \tag{5.24a}$$

$$u_l = G(\alpha_l)\frac{U_4}{U_2} + (1 - G(\alpha_l))\frac{U_3}{U_1}. \tag{5.24b}$$

$G(\alpha)$ should be a function in which: $G(0) = 0$ and $G(1) = 1$ and should tend smoothly from 0 to 1 very close to $\alpha = 0$. An illustration of $G(\alpha)$ is given in [63]. In the present work, the "tanh" function is found to have the mentioned properties by testing and selecting the suitable values for a and b such that there is a smooth and short transition between zero and unity, resulting in

$$G(\alpha) = 0.5 + 0.5\tanh(\frac{\alpha - a}{b}), \tag{5.25}$$

with $a = 0.0015$ and $b = 0.0001$. When using an explicit discretization, the velocities can easily be corrected with Eqs. 5.24a and 5.24b after calculating the conserved variables. However, when using the implicit method, this approach cannot be used due to the fact that the unknowns are guessed in the Newton loop. Therefore, in order to correct the variables, after each successful Newton iteration, the void fraction values are checked and the velocity of the vanishing phase is set to be equal to the remaining phase. Also, in order to avoid numerical errors, void fractions smaller than 0.001 are set to be equal to 0.001.

5.3 Application of the JFNK Method to the Four Equation Model

The present section describes how the four equation model is cast into the JFNK structure. For this purpose, the JFNK method is applied to the system of Eq. 3.23 using the vectors defined in Eq. 3.29. A vector function (F) must be defined along with a state vector (u) containing the unknowns. Note that in order to use Newton's method, the vector function

must be written in a form so that the solution to $F(u) = 0$ is sought. In the current study the following vector function is created

$$F(u) = \begin{pmatrix} \text{mass balance for gas phase at node 1} \\ \text{mass balance for liquid phase at node 1} \\ \text{momentum balance for gas phase at node 1} \\ \text{momentum balance for liquid phase at node 1} \\ \vdots \\ \vdots \\ \text{mass balance for gas phase at node } N_x \\ \text{mass balance for liquid phase at node } N_x \\ \text{momentum balance for gas phase at node } N_x \\ \text{momentum balance for liquid phase at node } N_x \end{pmatrix}. \tag{5.26}$$

Here, N_x is the number of nodes used for the simulation. For the state vector, u, two main choices are available, either the conserved variables or the set of primitive variables. All results presented in Section 5.6 are obtained by using the primitive variables as the vector of unknowns in the following form:

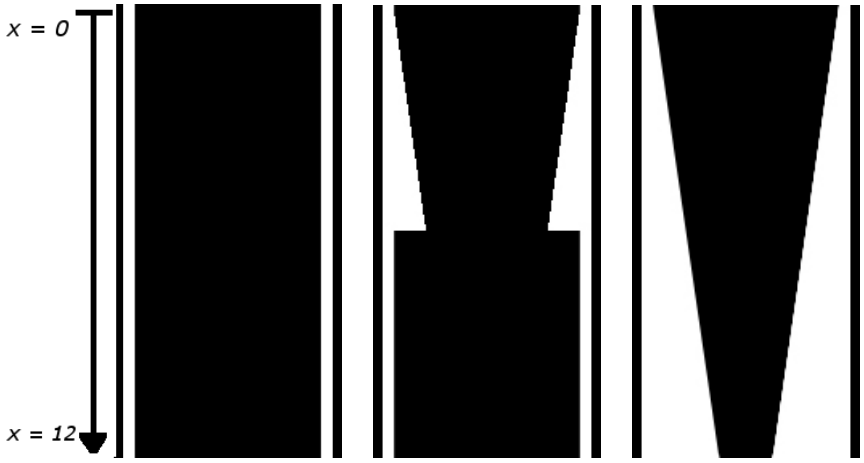
$$u = \begin{pmatrix} \alpha_g \text{ for node 1} \\ p \text{ for node 1} \\ u_g \text{ for node 1} \\ u_l \text{ for node 1} \\ \vdots \\ \vdots \\ \alpha_g \text{ for node } N_x \\ p \text{ for node } N_x \\ u_g \text{ for node } N_x \\ u_l \text{ for node } N_x \end{pmatrix} \tag{5.27}$$

One of the advantages of the JFNK method is the easy modification of any discretization scheme used. In order to change the time discretization, flux function and the discretization scheme for the source terms used in Eq. 5.5, only modifications to the vector function defined in Eq. 5.26 are required and all other parts of the code will remain the same.

5.4 Water Faucet

5.4.1 CFD Domain

Figure 5.1 illustrates the configuration of the water faucet problem. It consists of a 12 m vertical tube with a diameter of 1 m filled with a two-phase mixture of air and liquid water and the outlet of the tube is open to ambient pressure. Due to gravity, the water is driven downwards through the tube and the cross section is decreased: the flow narrows down, which is shown in Fig. 5.1(b). This causes the void fraction of the mixture to increase in time in the positive direction of the flow until steady state conditions are reached. The



(a) Initial state (b) Transition (c) Steady state

Figure 5.1: Ransom's water faucet configuration (x in meters)

steady state condition is shown in Fig. 5.1(c).Energy transfer plays no role in this test case, therefore the four equation model is valid.
The computational grid is constructed by N_x physical control volumes, and two ghost cells in order to implement the boundary conditions. All the variables are stored at node centers resulting in a collocated grid [1]. Wall friction and interfacial drag forces are neglected in the present test case.

5.4.2 Boundary Conditions & Initial Conditions

Initially the void fraction, α_g, throughout the entire tube is set to a value of 0.2, the velocity of the gas phase is set to 0, such that $u_g = 0$ m/s and that of the water to 10 m/s yielding $u_l = 10$ m/s. The pressure, p, inside the column is uniformly set to 10^5 Pa. In order to implement the boundary conditions, values for the void fraction and the velocities are set to their initial value at the inlet, and pressure is extrapolated from the interior domain. These values are set at the ghost cell adjacent to the inlet node. As for the outlet, only pressure is set to atmospheric conditions ($p = 10^5$ Pa) and the void fraction and fluid velocities are extrapolated from within the computational domain.

5.4.3 Analytical Solution

An analytical solution is available for the evaluation of the void fraction throughout the domain in time. In the derivation of this solution, the liquid phase is considered to be incompressible and the variation of pressure in the liquid phase is also neglected. Eq. 5.28 gives the expression for the analytical solution [63, 72]

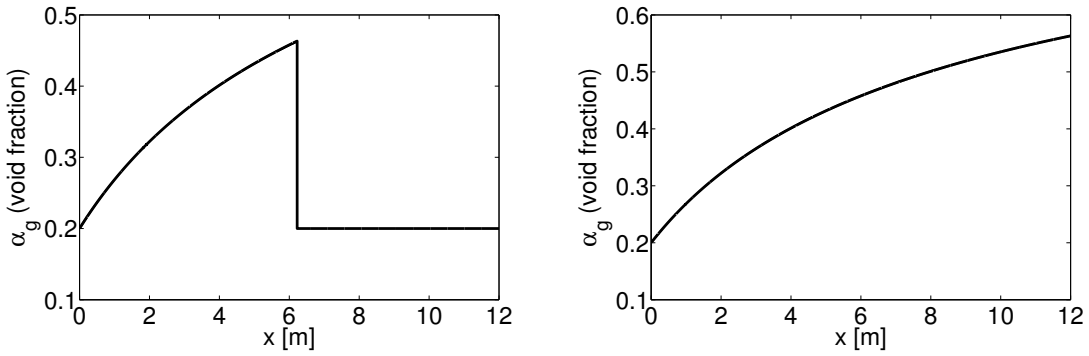
$$\alpha(x,t) = \begin{cases} 1 - \frac{(1-\alpha_o)(u_l)_o}{\sqrt{(u_l)_o^2 + 2gx}} & \text{if } x < \frac{gt^2}{2} + (u_l)_o t \\ 0.2 & \text{otherwise} \end{cases} . \tag{5.28}$$

The analytical solution for the water faucet test case at a transition time and at steady state are shown in Figure 5.2. This solution is used in order to verify the numerical predictions.

5.5 Oscillating Manometer

5.5.1 CFD Domain

For the oscillating manometer case, a U-tubed manometer with a length of 20 m, which is half filled with water, is considered, as shown Fig. 5.3. Again, energy transfer can be



(a) t = 0.5 s

(b) Steady state

Figure 5.2: Analytical solution of the void fraction inside the domain at t =0.5 s (a) and steady-state (b) for the water faucet test case

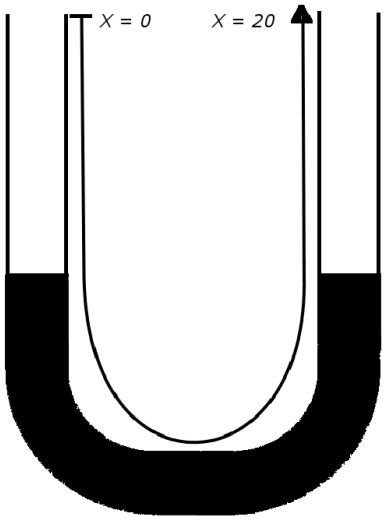


Figure 5.3: Ransom's oscillating manometer configuration (x in meters)

ignored, allowing the use of the isentropic model or 4-equation model. The original test case treats the manometer as a closed loop [66]. In the present investigation, the approach of [63] is applied where both ends of the tube are open to the atmosphere ($p = 10^5$ Pa). In contrast to the water faucet problem, the oscillating manometer case will test the ability of the numerical simulations to handle a change in the flow direction of the liquid column. In order to properly implement the geometry of the problem, the x-component of the gravity field must be modified including the effects of flow turning such as

$$g_x(x) = \begin{cases} g & \text{if } 0 \leq x \leq 5\text{m} \\ g\cos\left(\frac{(x-5)}{10}\pi\right) & \text{if } 5 < x \leq 15\text{m} \\ -g & \text{if } 15 < x \leq 20\text{m} \end{cases} . \tag{5.29}$$

5.5.2 Boundary Conditions & Initial Conditions

Initially, the water level on both legs of the manometer is equal. The column of water is assumed to have an initial velocity. Thus, the water level oscillates within the manometer. The velocity of both phases is equal to 2.1 m/s throughout the domain and the void fraction is set such that half the domain is filled with pure liquid ($\alpha_g = 0$) and the other half is filled with pure vapour ($\alpha_g = 1$). The pressure is also assumed to be hydrostatic at $t = 0$ s. Therefore, the following initial profiles are imposed,

$$\alpha_g(x) = \begin{cases} 0.999 & \text{if } 0 \leq x \leq 5\text{m} \\ 0.001 & \text{if } 5 < x \leq 15\text{m} \\ 0.999 & \text{if } 15 < x \leq 20\text{m} \end{cases} , \tag{5.30}$$

$$p(x) = \begin{cases} 10^5 & \text{if } 0 \leq x \leq 5\text{m} \\ (\rho_l)_o g\frac{L_w}{\pi}\sin\left(\frac{\pi(x-5)}{L_w}\right) & \text{if } 5 < x \leq 15\text{m} \\ 10^5 & \text{if } 15 < x \leq 20\text{m} \end{cases} , \tag{5.31}$$

where $(\rho_l)_o = 1000$ kg/m^3 and L_w is the length of the liquid column equal to 10 m.
In order to set the boundary conditions, since both ends are open to the atmosphere, the pressure is set to be equal to atmospheric pressure at the ghost cells and all other values are extrapolated from the interior domain. Similar to the water faucet case, wall friction is ignored but interfacial drag is included.

5.5.3 Analytical Solution

With the assumption that the liquid moves with uniform velocity due to gravity, the following analytical expression may be used for the solution of the liquid velocity at the bottom of the manometer through time [63]

$$u_l(t) = V_o \cos(\omega t), \tag{5.32}$$

where V_o is the initial velocity and

$$\omega = \sqrt{\frac{2g}{L_w}}. \tag{5.33}$$

5.5.4 Setup in CATHENA

For the oscillating manometer, the configuration is slightly different from what is presented in Section 5.5. Two vertical tubes are considered which are connected at the top and bottom. The gravitational force varies in magnitude for every node as a function of the elevation and changes sign for each leg. This configuration is shown in Fig. 5.4.

5.6 Results

In this section, the numerical results are presented for each of the two benchmarks. Three simulation codes corresponding to three different implementations to solve the governing equations, are written in MATLAB. The first code uses an explicit scheme for the discretization (the method described in [63]), in which all the fluxes and source terms are evaluated at the previous time step. Results obtained from the explicit discretization will be referred to as the explicit results. The second program includes a fully implicit scheme to create the system of equations solved by Newton's method. Here, the Jacobian matrix is explicitly calculated and the linear update equation is solved using the GMRES solver. All of the fluxes and source terms will be evaluated at the current time level (as shown in Eq. 5.5). Results from this method will be referred to as Newton-Krylov (NK) results.

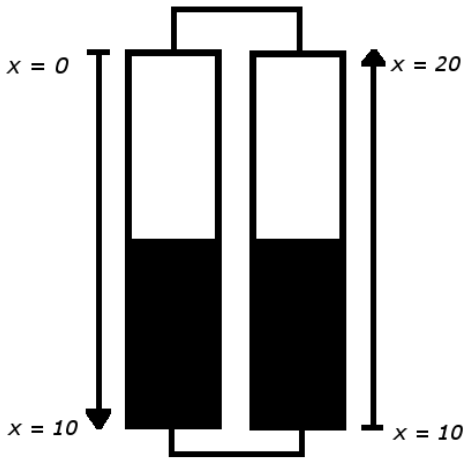


Figure 5.4: Oscillating manometer configuration in CATHENA (x in meters)

The final code will be similar to the NK method, but avoids the explicit calculation of the Jacobian matrix. Instead, the Jacobian-vector product is determined inside the GMRES solver as shown in Section 2.5. This method will be referred to as the JFNK method. For the JFNK method, the value used for ϵ in Eq. 2.24 is equal to 1.4×10^{-7} unless otherwise stated.

The effect of σ, parameter which makes the system of equations hyperbolic, initially shown in Eq. 3.35, is investigated in the present study. Also, different values for the CFL number are selected in order to identify the capability of each code when using different time steps. Finally, the results obtained from the JFNK method are compared with those from CATHENA. In all simulations, the Newton solver tolerance in Eq. 2.21 is set to a value of 10^{-3} and the tolerance in the GMRES solver is set to 10^{-6}.

5.6.1 Water Faucet

For this test case, the friction coefficient is set to zero ($C_f = 0$ in Eq. 3.36) and the low mach number model is not needed ($M_o = 1$ in Eq. 5.22). Several grid sizes and CFL numbers are tested as well as different values for σ. For the purpose of comparison

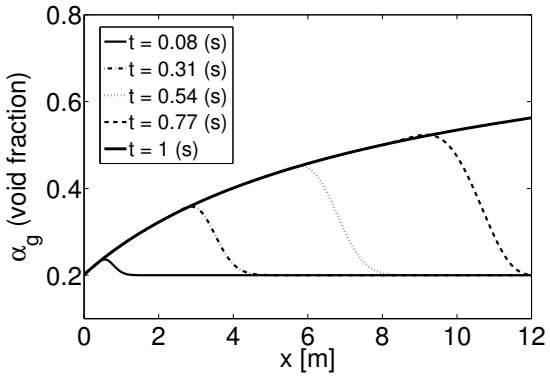
with previously published results [16, 61, 69] , the numerical predictions are focused on evolution of the void fraction throughout the domain at $t = 0.5$ s. However, similar results are obtained at any other time.

Time Evolution

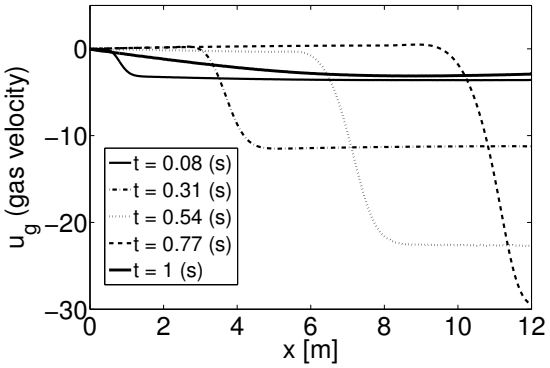
First, the time evolution of the void fraction is examined. Figure 5.5 a), b) and c) show the temporal profiles of the void fraction and the velocity for both phases, respectively. The results displayed in Fig. 5.5 are obtained by using the JFNK method, the non-hyperbolic model ($\sigma = 0$) and a mesh of 320 nodes. As can be seen in Fig.5.5 a), the void fraction begins to increase, starting from the top of the domain ($x = 0$) and as time passes, the rest of the domain is influenced by the upwards motion of the gas phase leading to an increase in the void fraction through the vertical tube. The motion of both phases can be seen by observing the gas and liquid velocities, as shown in 5.5 b) and c) . The gas inside the tube rises to the top of the domain (negative velocity) with time, leading to the increase in the void fraction. Due to gravity, the velocity of the liquid phase increases in time with a downwards direction (positive velocity). The present observations are in agreement with previous studies [63, 16, 61].

Comparison of Hyperbolic and Non-Hyperbolic Models

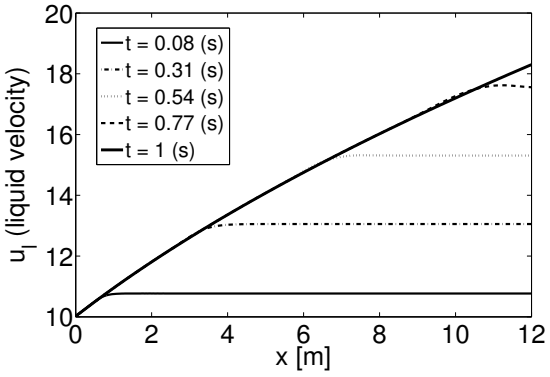
The original system of equations ($\sigma = 0$ in Eq. 3.29) is non-hyperbolic and regularization terms are added to make the system of equations hyperbolic. In the current study, a pressure correction term is added, as shown in Eq. 3.35. For the three methods described in Section 5.6, the choice of σ does not have an impact on the accuracy of the results for grids up to 320 nodes (not shown for brevity). When finer meshes are considered, oscillations begin to appear near the discontinuity of the void fraction. Figure 5.6 shows these oscillations for the explicit and JFNK methods, respectively, for two different grids, 640 nodes and 1280 nodes. Similar results are obtained from the NK method, but are not shown. These oscillations may be controlled to a certain extent by the addition of the regularization term (pressure correction) and making the system of equations hyperbolic



(a) void fraction

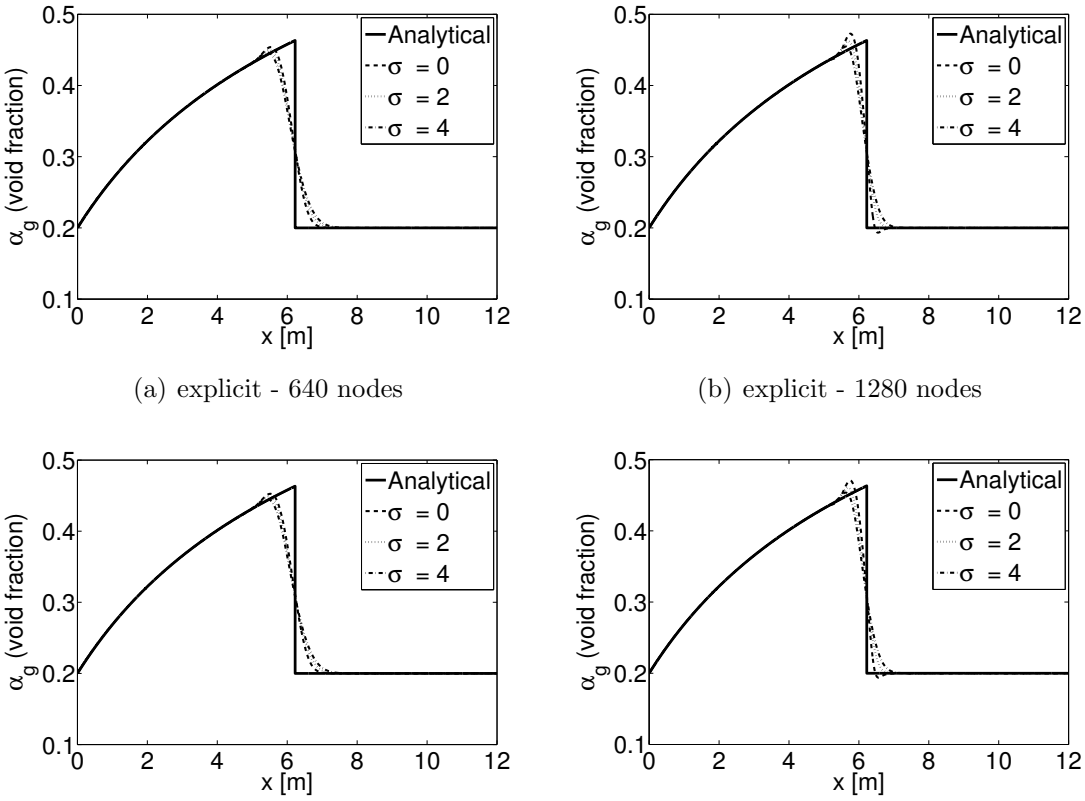


(b) gas velocity



(c) liquid velocity

Figure 5.5: Time evolution of the void fraction and the fluid velocities in Ransom's water faucet problem using the non-hyperbolic model ($\sigma = 0$) and 320 grid points



(a) explicit - 640 nodes

(b) explicit - 1280 nodes

(c) JFNK - 640 nodes

(d) JFNK - 1280 nodes

Figure 5.6: Comparison of hyperbolic ($\sigma = 2$ and $\sigma = 4$) and non-hyperbolic ($\sigma = 0$) model for the explicit and JFNK methods at $t = 0.5$ s

(with $\sigma > 0$). For comparison, the results using different values of σ are also included in Figure 5.6. It can be seen that as the number of nodes increases, the effect of σ becomes clearer and the oscillations are further dampened.

The computational time required for the simulations to reach a solution at t = 0.5 s, using a hyperbolic (with $\sigma = 2$) and non-hyperbolic model with $\epsilon = 1.4 \times 10^{-7}$ are presented in Table 5.1. It can be seen that no significant difference is noticed when the hyperbolic model is used. However, when a different value for the perturbation number used for the Jacobian approximation is selected, different computational times and trends are obtained.

Table 5.2 shows the CPU time required for the JFNK simulations with $\sigma = 0$ and σ =2

Table 5.1: Comparison of computational time in seconds of the hyperbolic ($\sigma = 2$) and the non-hyperbolic ($\sigma = 0$) model with CFL = 50 for the JFNK method using $\epsilon = 1.4 \times 10^{-7}$ to reach a solution at t = 0.5 s

	No. of nodes				
	40	80	160	320	640
Non-hyperbolic	10	46	233	1840	15478
Hyperbolic	10.70	46	232	1835	15125
% decrease	0	0	0	0	2

for different grid sizes and $\epsilon = \sqrt{\epsilon_{machine}} = 1.4 \times 10^{-8}$. It can be seen that a large amount of computational time can be saved by using the hyperbolic model. For example, a computational saving of 40 % is observed for a grid of 320 nodes with a hyperbolic model (σ =2) compared to the non-hyperbolic form. This is due to the ill-posedness of the non-hyperbolic model which requires more GMRES iterations at each time-step to converge compared to the hyperbolic model.

Table 5.2: Comparison of computational time in seconds of the hyperbolic ($\sigma = 2$) and the non-hyperbolic ($\sigma = 0$) model with CFL=10 using $\epsilon = 1.4 \times 10^{-8}$ to reach a solution at t = 0.5 s

	No. of nodes			
	40	80	160	320
Non-hyperbolic	18.98	133	963	13186
Hyperbolic	18.88	101	680	7920
% decrease	0	24	29	40

Grid Sensitivity Analysis

The void fraction profiles throughout the domain at $t = 0.5$ s for the JFNK method with σ = 0 are shown in Figure 5.7 for different mesh sizes. As the number of nodes increases, the

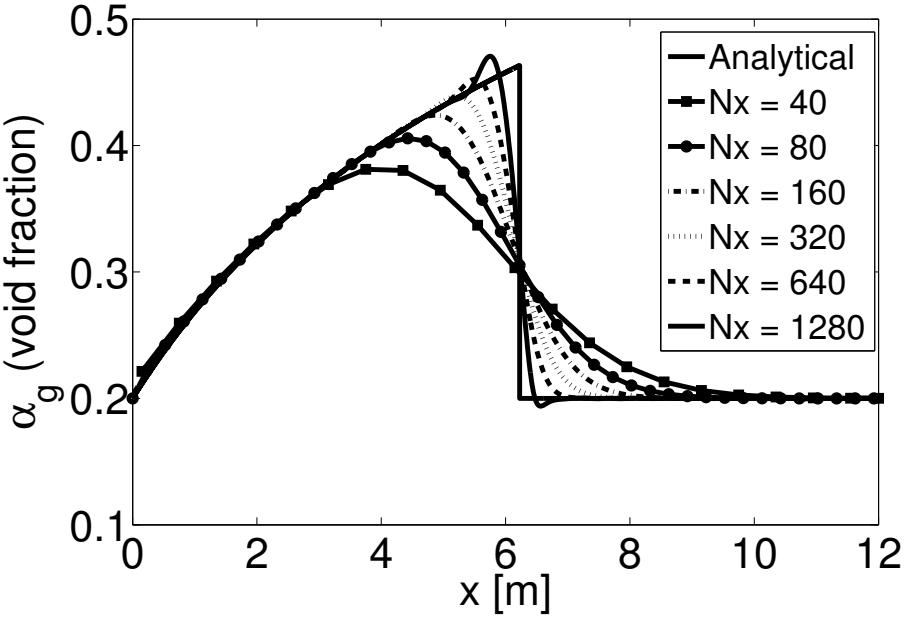


Figure 5.7: Grid convergence study for the implicit code at $t = 0.5$ s with $\sigma = 0$

results get closer to the analytical solution at the given time of 0.5 s. However, similar to the explicit method (shown in [63] and not included for brevity), some oscillations occur when fine meshes are used, as shown for 640 and 1280 nodes in Figure 5.7. These oscillations may be controlled by the use of the pressure correction term to make the system of equations hyperbolic. By doing so, the oscillations are dampened at the cost of losing accuracy near the discontinuity located at $x = 6$ m.

Comparison Between the Three Methods, Explicit, NK (Implicit) and JFNK (Implicit)

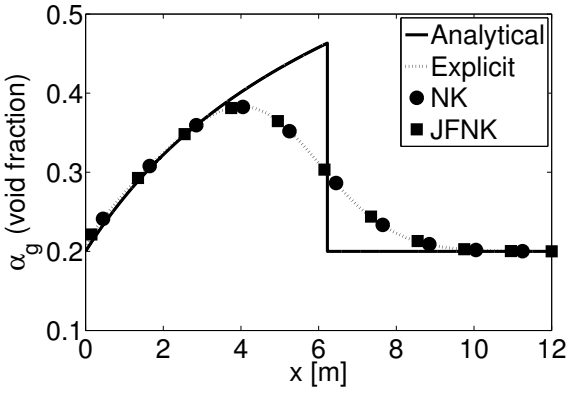
The numerical predictions obtained from the three methods described in Section 5.6, are compared. Figure 5.8 shows the void fraction predictions obtained from the three methods on three different mesh sizes of 80, 160 and 320 nodes. As mentioned at the beginning of the present section, negligible differences in the predicted values obtained the hyperbolic and non-hyperbolic model with $\sigma = 2$ are found. Therefore, only the results using the non-

hyperbolic model ($\sigma = 0$) are presented here. The solution obtained from both implicit methods, NK and JFNK, almost exactly overlap with the explicit solution and no significant difference is noticed. With the finest mesh ($N_x = 1280$), the solution for the JFNK case decreases the error caused by the oscillation near the discontinuity, by almost 15%, as shown in Figure 5.9 (the magnitude of the overshoot and undershoot have decreased). For the remainder of this section, since similar results are obtained from the NK and JFNK methods, only the JFNK results are included.

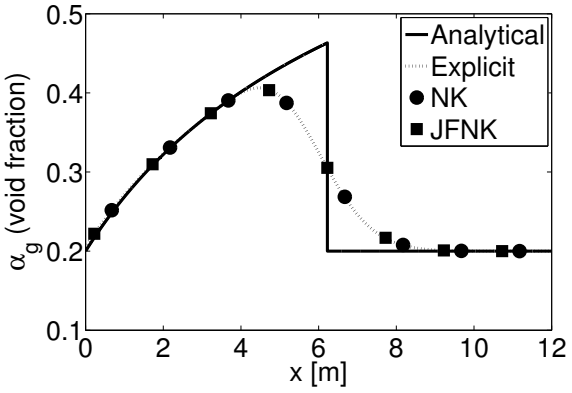
It is well known that explicit schemes are restricted to a certain value for the CFL number. Using the definition in Eq. 5.6, the time step can be calculated according to the specified CFL number. A higher value for the CFL number means a larger time step. The explicit code can handle a CFL number up to 2. Higher values chosen for the CFL number lead to large oscillations and eventually cause the code to diverge. On coarse meshes with a maximum nodes of 340, the solution remains stable with any value up to CFL = 2. However, as can be seen in Figure 5.10, for a CFL number equal to 2 on the two finest meshes (640 and 1280 nodes), some instabilities are visible. Further, as the number of nodes is increased from 640 points to 1280 points, these instabilities are significantly amplified.

Due to the implicit nature of the JFNK method, higher CFL numbers are expected to be handled compared to the explicit method. Figure 5.11 shows the range of CFL numbers included with the JFNK method on four different mesh sizes with a maximum number of nodes equal to 340 to avoid the oscillations observed for the non-hyperbolic formulation (Fig. 5.7). As expected, the JFNK method is capable of handling very high CFL numbers on all mesh sizes. During the calculations, the significance of the value of ϵ was also noticed. If the perturbation value is set to 1.4×10^{-8}, high CFL numbers cannot be used for fine meshes due to instabilities observed, which would lead to divergence in the outer Newton loop. With an appropriate choice of the perturbation value, high CFL may be used to reach a converged solution. In the current study, a trial an error procedure was performed to find a suitable value for ϵ (1.4×10^{-7}). However, other approaches have been proposed for selecting ϵ which can be found in [42, 12] and have not been investigated in this study.

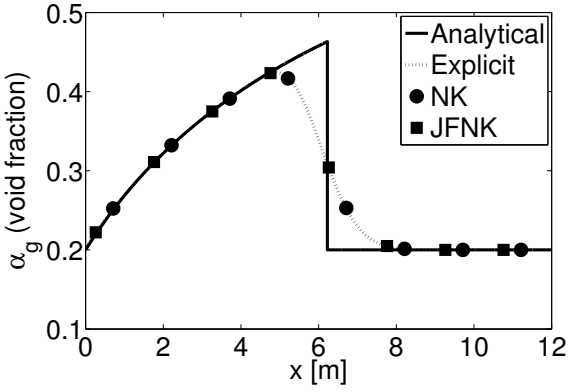
For further comparison, the computational time for the NK and JFNK methods are examined. Table 5.3 shows the CPU time required for the two methods in seconds to reach a solution at t = 0.5 s. It can be seen that a great amount of computational time can be



(a) $N_x = 80$



(b) $N_x = 160$



(c) $N_x = 320$

Figure 5.8: Comparison of accuracy between the different methods for coarse meshes using $\sigma = 0$ at $t = 0.5$ s

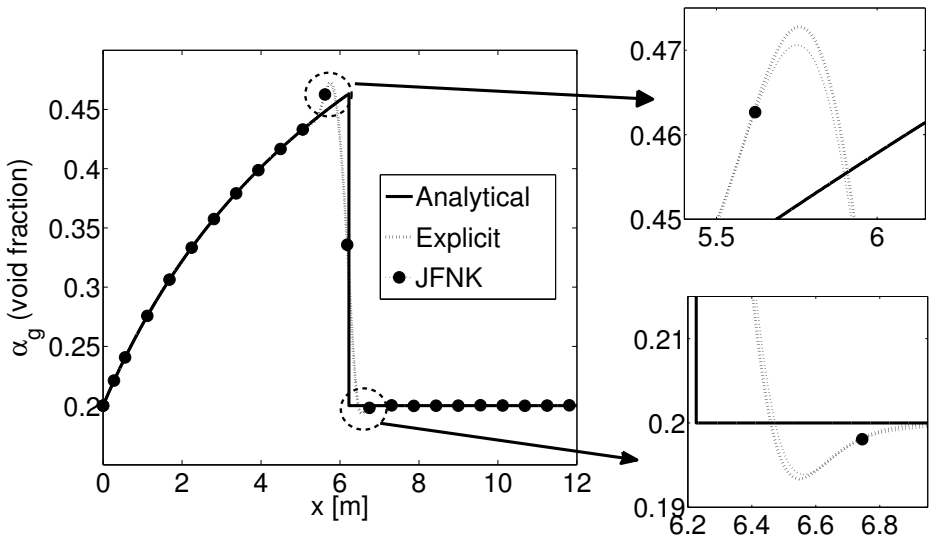
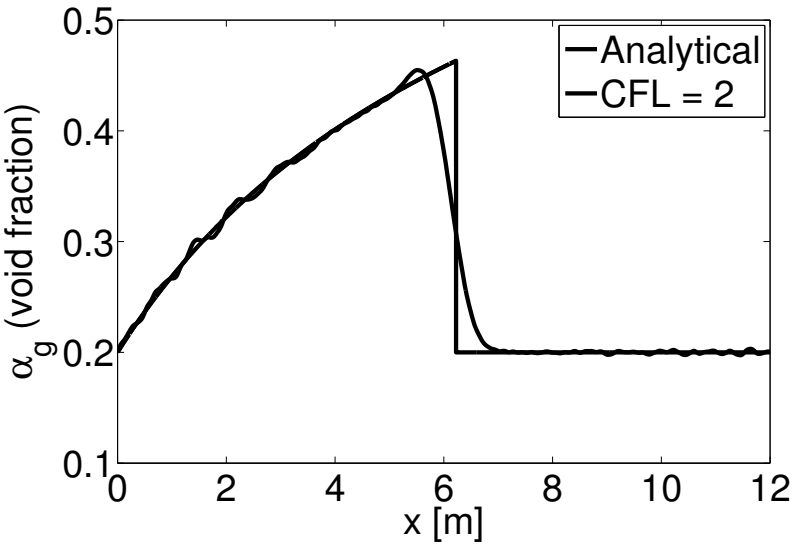


Figure 5.9: Comparison of accuracy between the explicit and JFNK methods for the finest mesh ($N_x = 1280$) using $\sigma = 0$ at $t = 0.5$ s

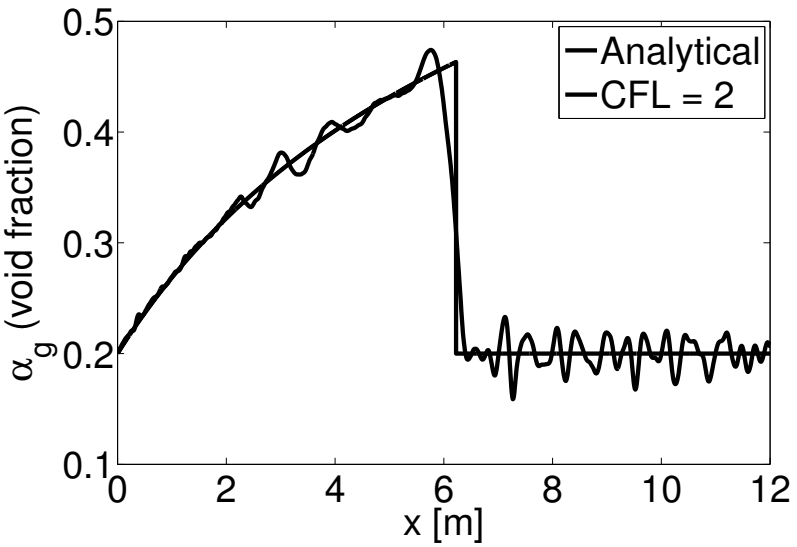
saved by the approximation method in JFNK, instead of explicitly calculating the Jacobian matrix. As the system size increases, the computational time benefit becomes smaller, but still significant on the order of 70% for 640 nodes. Due to the first order accuracy of the discretization scheme, at any given node, each conservation law requires 12 input variables (α_g, u_g, u_l and p for itself and its neighboring nodes) to calculate the residual. Therefore, the Jacobian matrix, consists of 12 non-zero values on each row (corresponding to a single conservation law at a specific node). It is clear that avoiding the construction of such a matrix is highly beneficial in terms of computational time.

Steady-state solution

As shown in Figure 5.11, with increasing CFL numbers, the numerical predictions depart further from the analytical solution at an intermediate time (0.5 s), in particular visible for the first two meshes consisting of 40 and 80 nodes. It is important to verify if these discrepancies remain in the steady-state solutions for high CFL numbers. Figure 5.12 shows

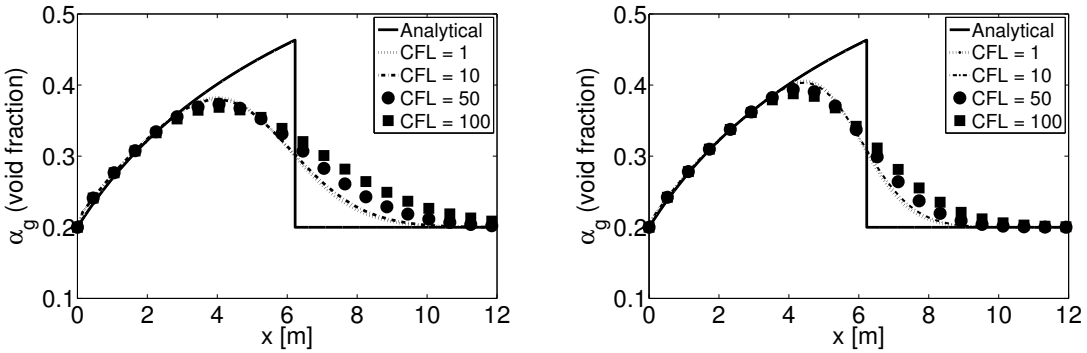


(a) $N_x = 640$



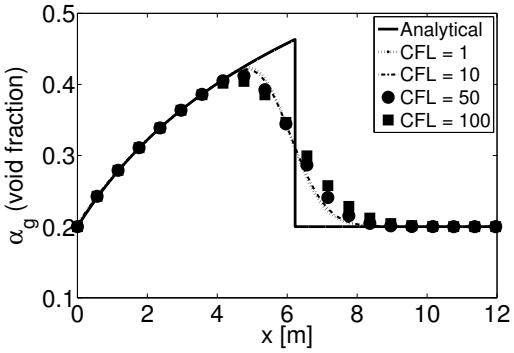
(b) $N_x = 1280$

Figure 5.10: Instability of the explicit method when using high CFL numbers

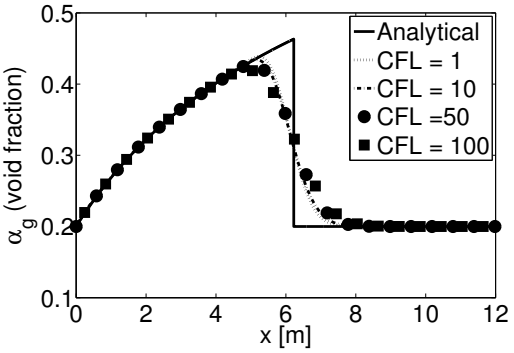


(a) $N_x = 40$

(b) $N_x = 80$



(c) $N_x = 160$



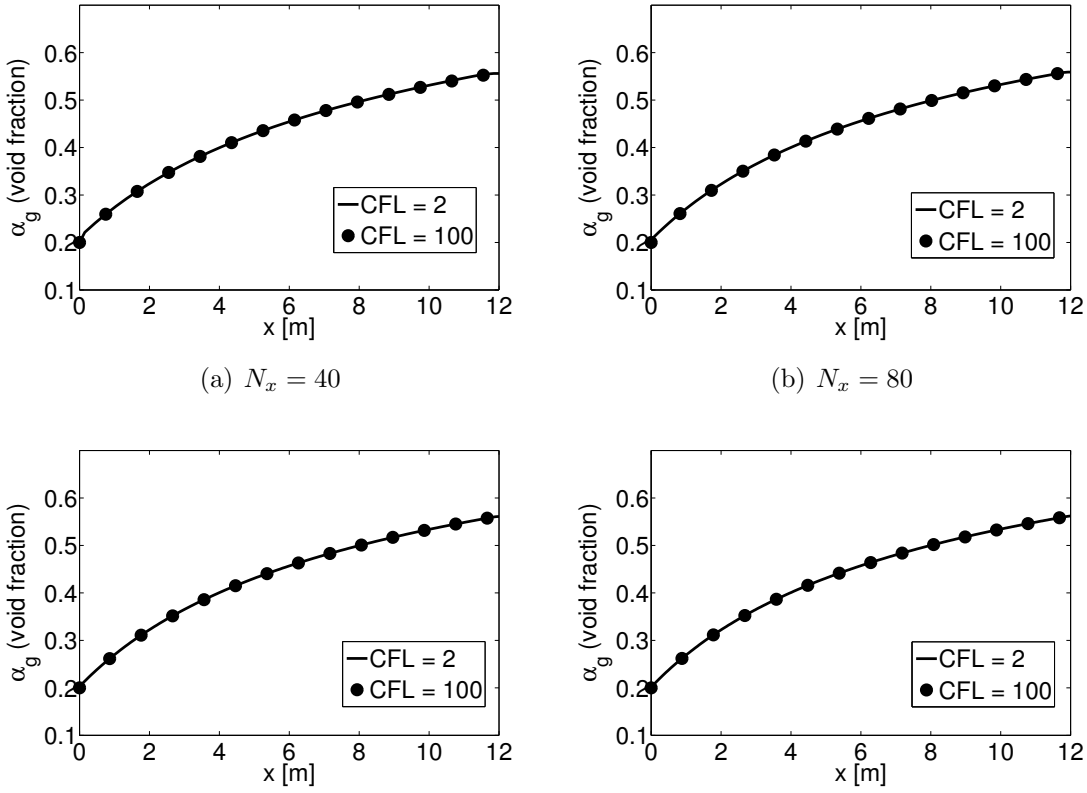
(d) $N_x = 320$

Figure 5.11: Range of CFL numbers using the JFNK method using $\sigma = 0$ at $t = 0.5$ s

Table 5.3: Comparison of CPU time, in seconds for the NK and JFNK methods (the non-hyperbolic model, $\sigma = 0$, is used for both methods)

No. of nodes	CFL	NK	JFNK	% decrease
40	10	64	11	82
80	10	332	46	86
160	50	414	77	81
320	100	1625	386	76
640	100	15062	4601	69

the steady state solution for the same four grids than shown in Fig. 5.11 and two CFL values of 2 and 100. As can seen in Fig. 5.12, no significant difference can be noticed in the void fraction predictions, demonstrating that high CFL numbers can be used without a loss in accuracy when the simulation is to be run until steady state. The computational is also considered and Table 5.4 presents the CPU time required to reach the steady state solution using the same four meshes. As displayed in Table 5.4, a significant decrease in computational time can be achieved when higher CFL numbers are imposed. Using high CFL numbers will require fewer timesteps to reach the steady state solution.



(a) $N_x = 40$ (b) $N_x = 80$

(c) $N_x = 160$ (d) $N_x = 320$

Figure 5.12: Steady state solutions using low and high CFL numbers for different mesh sizes

Table 5.4: Comparison of computational time in seconds until steady state is reached for high and low CFL numbers.

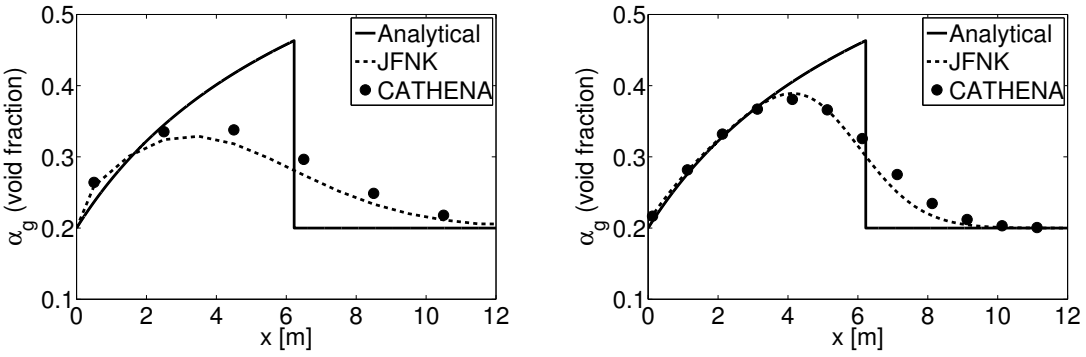
	No. of nodes			
	40	80	160	320
Low CFL	193	782	840	22773
High CFL	7	31	182	1569
% decrease	96	96	87	93

Comparison with CATHENA 4 predictions

As the ultimate goal of the current work is to incorporate JFNK into CATHENA, it is interesting to make a comparison between the predictions obtained by the current JFNK implemented in Matlab and those from the current stable version of CATHENA. Figure 5.13 shows the results obtained from CATHENA and the JFNK method at t = 0.5 s. A CFL number equal to 1 is used for the JFNK results and CATHENA uses an adaptive time step resulting in a variable CFL number during the calculation. As can be seen in Fig.5.13, the results from the JFNK method are closer to the analytical solution compared to the profiles from CATHENA. Further, the shock is also captured more accurately in the present JFNK results. These differences are mostly attributed to the use of the AUSM+ flux in the JFNK code compared to the simple upwind scheme in CATHENA which does not produce high resolutions near the discontinuity. A direct comparison between the CPU time required by CATHENA and that by JFNK is not possible due to the differences in the coding language and implementation.

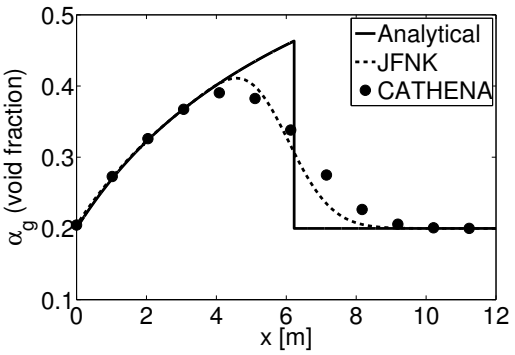
5.6.2 Oscillating Manometer

The oscillating manometer is a much more complicated numerical benchmark due to the existence of a column of pure liquid in contact with pure vapor. The low Mach number model must be used and an appropriate value for M_o must be found (refer to Eq. 5.22). A value of $M_o = 0.2$ is found to produce good results and prevent numerical oscillations caused by odd-even decoupling. Also, for this test case, a friction coefficient of $C_f = 50000$

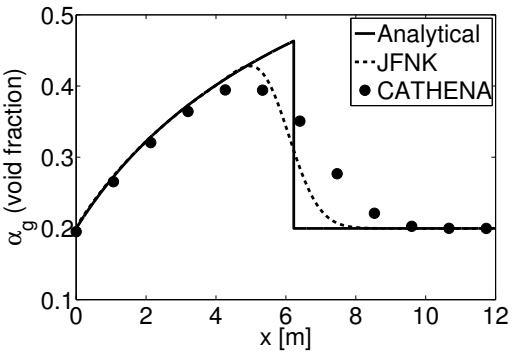


(a) $N_x = 12$

(b) $N_x = 48$



(c) $N_x = 96$



(d) $N_x = 192$

Figure 5.13: Comparison between JFNK predictions and CATHENA results at $t = 0.5$ s

in Eq. 3.36 is found to be suitable in order to reach a converged solution. The liquid velocity at the bottom of the manometer (corresponding to $x = 10$) is monitored through time. Therefore, small CFL numbers are used to increase the accuracy of the transient results. For this test case, two grid sizes are tested, one including 50 nodes and one with 100 nodes. However, no significant difference is noticed in the liquid velocity profiles. Therefore, the predictions shown in the current section are obtained by using $N_x = 50$.

Comparison between the three methods, explicit, NK (implicit) and JFNK (implicit)

Figure 5.14 shows the liquid velocity at the bottom of the manometer as a function of time. For the explicit method, very small timesteps must be used in order to avoid divergence (CFL=0.05). The JFNK and NK methods are capable of handling higher CFL numbers, however, due to the transient nature of the property under consideration, CFL = 0.1 is set. The results obtained by the NK method are similar to those obtained from the JFNK method and no significant difference is noticed. Thus, the NK predictions are not included in Figure 5.14. For both the explicit and JFNK methods, a phase error is noticed with respect to the analytical solution, although this error is amplified for the JFNK method. It becomes more visible after a few oscillations of the water column. A small amount of damping due to numerical diffusion is also noticed for both methods. In terms of computational time, the CPU time required for the JFNK method is about 80% lower than the NK method. No significant difference in accuracy or CPU time was noticed between the hyperbolic and non-hyperbolic models.

Comparison with CATHENA 4 predictions

Figure 5.15 shows the results obtained by CATHENA and the JFNK method. Both results use $N_x = 50$ and for the JFNK results, the non-hyperbolic model is used. Due to the first order discretization used in CATHENA, which is highly diffusive, a large amount of damping is noticed compared to the JFNK method, which is a high resolution AUSM+ scheme and therefore has an order of magnitude less diffusion. Again, the phase error can be seen from CATHENA and it is slightly less than the JFNK method.

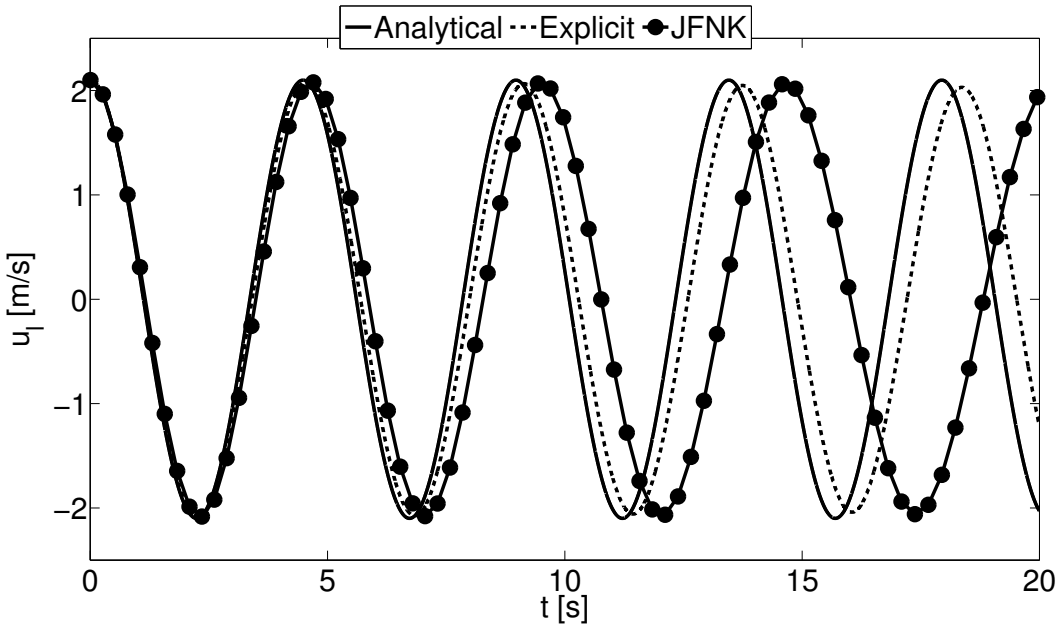


Figure 5.14: Liquid velocity at the bottom of the manometer as a function of time showing the analytical solution, explicit and JFNK predictions

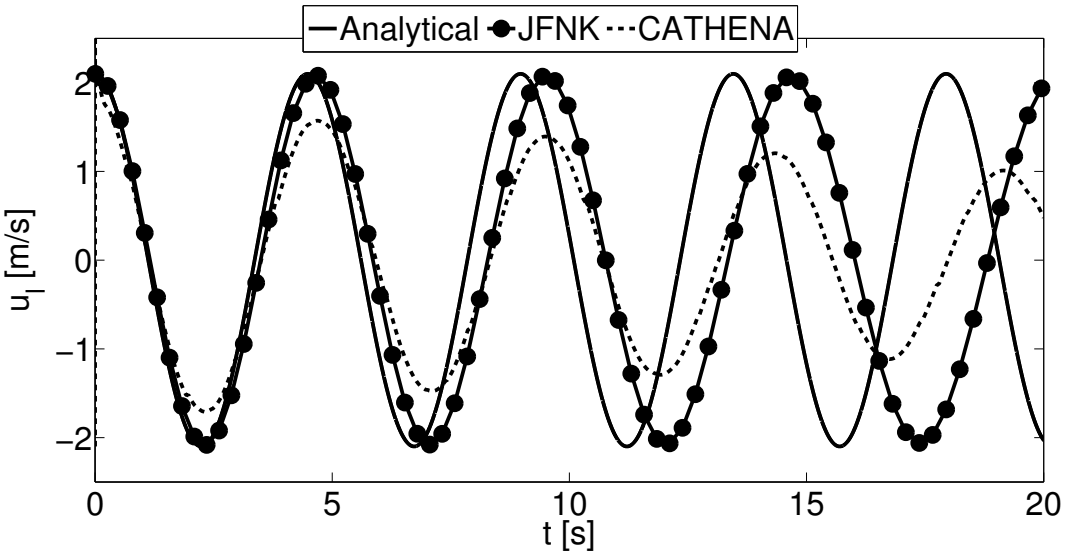


Figure 5.15: Comparison between liquid velocity at the bottom of the manometer as a function of timeshowing the analytical solution, JFNK and CATHENA predictions

5.7 Summary

In the current chapter, the JFNK is implemented to solve the governing equations related to the isentropic two-fluid model. Two main numerical benchmarks are used for the simulations, namely the water faucet and the oscillating manometer test case. A detailed investigation is performed in order to identify the advantages and disadvantages of the JFNK method compared to the NK method and a fully explicit discretization scheme. The numerical predictions are also compared to solutions obtained from CATHENA for both test cases. All methods predict similar results through time for the water faucet problem and are in agreement with previous numerical work. Regarding the oscillating manometer test case, the phase shift caused due to the drag coefficient is amplified when the JFNK method is selected. Relatively higher CFL numbers can be selected for the simulations when the implicit methods (JFNK and NK) are used to solve the system of equations compared to an explicit approach. CPU time for both test cases decrease considerably when JFNK is chosen instead of the NK method. The decrease in computational time is in the order of 80%. Compared to CATHENA, finer grids can be used for the

simulations in addition to the more accurate shock resolving discretization scheme selected for the present investigation (AUSM+) which lead to a closer prediction compared to the analytical solution for each test case.

Chapter 6

Conclusions

Throughout the present investigation, the performance of the JFNK method is examined. Specifically, the method is applied to solve a fully implicit discretization of the 1D SWWE and the four equation two-fluid model equations. Prior to the presentation of the numerical results, the governing equations for each case are derived and mathematical properties of each set of equations are discussed. The discretization schemes used are presented and appropriate boundary and initial conditions corresponding to each test case are included. Numerous simulations are performed and the predictions obtained are compared to an explicit method, and also the NK method. The results are also compared to previous numerical work which utilize an explicit approach to update the solution parameters in time.

In the present section, the conclusions obtained from the current investigation are summarized and recommendations for future work are also included.

6.1 Summary of Findings

Regarding the 1D SWWE system of equations (corresponding to single phase flow), a fully implicit discretization scheme is constructed by utilizing a second-order central-upwind Godunov scheme for the spatial discretization of the governing equations and a backward

Euler scheme for the temporal discretization (unless stated otherwise). The test case under consideration corresponds to the 1D dam break problem. Initial conditions are selected in such a way to define the flow regime after the virtual dam is removed (sub critical, supercritical and a dry bed case). In order to observe the effect of the numerical discretization on the performance of the JFNK method, a Lax-Friedrich spatial discretization and a Crank-Nicolson temporal discretization are also implemented. The conclusions obtained are as follows.

- The converged solution obtained from the JFNK method is in agreement with previous numerical studies.
- As the number of grid points are increased by a factor of 2, the computational time required for the JFNK method to reach a converged solution is increased by a factor of 4.
- Using constant CFL values, JFNK, NK and explicit methods converge to the same solution for all flow regimes.
- The CPU time required for the explicit method is extremely smaller than the other two implicit schemes due to the large system of equations which needs to be solved in implicit schemes.
- The JFNK method is able to approximately save 90% computational time compared to the NK method.
- Instabilities appear in the explicit method predictions when CFL values higher than 0.1 are used, however both implicit methods are capable of handling relatively larger CFL values up to 40 with the cost of loosing accuracy compared to the analytical solution.
- Higher CFL values cause the numerical predictions to smear along the spatial domain and not capture the shock accurately.
- The maximum CFL value leading to a converged solutions varies for different flow regimes. For the sub critical flow regime a value of 40 may be used and for the super

critical and dry bed case values equal to 3 and 1 may be used respectively. Higher values result in an infinite Newton loop.

- Changing the perturbation value for the Jacobian approximation does not resolve the issue with the infinite Newton loop. The NK method is also not able to use high CFL values meaning that the issue is not a result of a poor Jacobian approximation.
- Different discretization schemes do not behave the same with JFNK. Although using a higher order scheme improves the numerical results, the CFL range and instabilities noticed vary from case to case.

Moving forward towards a set of equations describing the motion of two-phase flows, the isentropic two-fluid model, a system of fully implicit discrete equations are constructed. For the temporal discretization, a backward Euler scheme is selected and for the convective fluxes, the AUSM+ scheme is chosen which has proven to be robust and accurate for multiphase problems. Two test cases are chosen as numerical benchmarks, the water faucet and the oscillating manometer problem. A detailed investigation is performed on the behavior of JFNK on these two test cases. The results are compared with those obtained from a NK method and a fully explicit method. In addition, numerical predictions are also compared with results from CATHENA. The major findings are summarized below.

- The time evolution for the faucet problem obtained from JFNK is in agreement with previous numerical work (explicit results).
- The hyperbolic model does not affect the accuracy of the results obtained for the water faucet problem for grids up to 320 points. Using finer meshes, 640 and 1280 points, oscillations appear which can be controlled to a certain extent with the inclusion of a pressure correction term ($\sigma \neq 0$). A larger value for *sigma* further dampens the oscillations observed near the discontinuity.
- The value of the perturbation number, ϵ, has an impact on the computational time required to reach a converged solution for the faucet problem. This implies that with different values of ϵ, the hyperbolic parameter σ needs to be tuned to obtain optimal performance.

- Utilizing a finer mesh for the faucet problem increases the accuracy of the numerical predictions compared to the analytical solution. However, at a certain point (640 cells), oscillations appear near the discontinuity. These oscillations are not of numerical nature and are caused due to the the fact that the system of equations are ill-posed. The addition of the pressure correction term solves this issue.

- For the faucet problem, all three solution methods (JFNK, NK and explicit) converge to the same result for grid sizes up to 640 using the same CFL number. On the finest mesh, 1280 points, the oscillation near the discontinuity decreases by almost 15% using the implicit methods.

- The CFL number permitted for the implicit methods is considerably larger compared to the explicit method. Using CFL numbers higher than 2 causes numerical instabilities to occur with the explicit method. The JFNK method converges even with CFL values up to 100.

- The value of ϵ effects the maximum CFL number that can be handled with JFNK for the faucet problem. A value of $\epsilon = 1.4 \times 10^{-8}$ is found to permit large CFL numbers.

- Using JFNK can decrease the computational time required by almost 80% compared to the NK method. However, the CPU time required for the explicit method is significantly less than both implicit methods.

- For the water faucet problem, the JFNK method is able to give more accurate numerical predictions compared to CATHENA for the same grid sizes. This is due to the shock capturing scheme used in JFNK, AUSM+. Also, JFNK is able to converge with finer grids compared to CATHENA.

- The oscillating manometer converges only with a specific numerical setup. The low Mach number model must be used with $M_o = 0.2$ and the drag coefficient must be set to $C_f = 50000$. Other configurations lead to divergence of the numerical predictions.

- The implicit schemes predict a phase shift in the liquid velocity profile through time for the oscillating manometer compared to the analytical solution. The explicit

method also predicts this phase shift but with a smaller magnitude. The phase shift is due to the inclusion of the drag coefficient.

- JFNK predicts the magnitude of the oscillation of liquid velocity with more accuracy compared to CATHENA for the oscillating manometer problem, however, the phase shift is larger.

6.2 Recommendation for Future Work

The result of the present investigation was the successful application of the JFNK method to the 4-equation two-fluid model. The method should be extended to the 6-equation two-fluid model which includes the energy equation for each phase as well. A first attempt was taken but the Newton loop would not converge. Most likely, this is due to a poor approximation of the Jacobian matrix caused by a large difference in the order of magnitude of the energy equation and the mass and momentum equation. To solve this issue, the governing equations need to be cast in a non-dimensional form so that all equations are in the same order of magnitude.

Another aspect of the JFNK method is the use of effective pre-conditioners. Pre-conditioners can greatly decrease the computational time required to solve the linear system of equations at each Newton update. Some general pre-conditioners may be applied to reduce the computational time, however, a physics based pre-conditioner specifically constructed for each system of equations can be more effective.

Finding a suitable perturbation value was based on a trial and error method. A more detailed study on the value of ϵ can lead to a more efficient set of simulations and further decrease the computational time.

References

[1] J.H Ferziger andM. Perić. *Computational methods for fluid dynamics*, volume 3. Springer Berlin, 1996.

[2] N.U. Aydemir. A Discussion of Hyperbolicity in CATHENA 4: Virtual Mass and Phase-Interface Pressure Differences. *Kerntchnik*, 77:128–133, 2012.

[3] N.U. Aydemir. Recent Developments in CATHENA 4: A Fully-Implicit Network Thermalhydraulics Code. In *NURETH-15*, Pisa, Italy, May 12-15, 2013.

[4] S Banerjee and W.T. Hancox. Transient thermohydraulics analysis for nuclear reactors. In *Proc. 6-th IHTC, Toronto*, 1978.

[5] F. Barre and M. Bernard. The CATHARE code strategy and assessment. *Nucl. Eng. Des.*, 124(3):257 – 284, 1990.

[6] C. Bellos and J. Sakkas. 1d dambreak floodwave propagation on dry bed. *J. Hydraul. Eng.*, 113(12):1510–1524, 1987.

[7] M. Benzi. Preconditioning techniques for large linear systems: A survey. *J. Comput. Phys.*, 182(2):418 – 477, 2002.

[8] F. Berger and J. Colombeau. Numerical solutions of one-pressure models in multifluid flows. *SIAM J. Numer. Anal.*, 32(4):1139–1154, 1995.

[9] D. Bestion. The physical closure laws in the CATHARE code. *Nucl. Eng. Des.*, 124(3):229 – 245, 1990.

[10] J.A. Bour. On the form of the pressure terms in the momentum and energy equations of two-phase flow models. *Int. J. Multiphas. Flow*, 5(2):159 – 164, 1979.

[11] P. Brown. A Local Convergence Theory for Combined Inexact-Newton/Finite-Difference Projection Methods. *SIAM J. Numer. Anal.*, 24(2):407 – 434, 1987.

[12] P. Brown and Y. Saad. Hybrid Krylov Methods for Nonlinear Systems of Equations. *SIAM J. Sci. Stat. Comp.*, 11(3):450 – 481, 1990.

[13] P. Brown and Y. Saad. Convergence Theory of Nonlinear Newton-Krylov Algorithms. *SIAM J. Optimiz.*, 4(2):297 – 330, 1994.

[14] C.G. Broyden and M.T. Vespucci. *Krylov Solvers for Linear Algebraic Systems: Krylov Solvers*, volume 11. Elsevier, 2004.

[15] K.E. Carlson. *RELAP/MOD3 Code Manual*. US NRC, 1990.

[16] C.H. Chang and M.S. Liou. A robust and accurate approach to computing compressible multiphase flow: Stratified flow model and AUSM+-up scheme. *J. Comput. Phys.*, 225(1):840 – 873, 2007.

[17] A. Chertock, A. Kurganov, and G. Petrova. Finite-volume-particle methods for models of transport of pollutant in shallow water. *J. Sci. Comput.*, 27(1-3):189–199, 2006.

[18] F. Coquel and M.S. Liou. chapter Field by field hybrid upwind splitting methods. Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, Jul 1993. 0.

[19] C.T. Crowe. *Multiphase flow handbook*. CRC Press, 2005.

[20] I.G. Currie. *Fundamental mechanics of fluids*. CRC Press, 2012.

[21] J.M. Delhaye. *Two-phase Flow Measurements*. Thermal Transfer Agency, CEA, 1974.

[22] J.R. Edwards. A low-diffusion flux-splitting scheme for navier-stokes calculations. *Comput. Fluids*, 26(6):635 – 659, 1997.

[23] J.R. Edwards and M.S. Liou. Low-diffusion flux-splitting methods for flows at all speeds. *AIAA J.*, 36(9):1610 – 1617, 1998.

[24] S. Evje and T. Flåtten. Weakly implicit numerical schemes for a two-fluid model. *SIAM J. Sci. Comput.*, 26(5):1449 – 1484, 2005.

[25] R.P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method). *J. Comput. Phys.*, 152(2):457 – 492, 1999.

[26] R.J. Fennema and M. Hanif Chaudhry. Simulation of one-dimensional dam-break flows. *J. Hydraul. Res.*, 25(1):41–51, 1987.

[27] S.K. Godunov and V.S. Ryabenkii. *Difference schemes: an introduction to the underlying theory*. Elsevier, 1987.

[28] B.N. Hanna. CATHENA: A thermalhydraulic code for CANDU analysis. *Nucl. Eng. Des.*, 180(2):113 – 131, 1998.

[29] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, iii. *J. Comput. Phys.*, 71(2):231–303, 1987.

[30] C. Hirt and B. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.*, 39:201–225, 1975.

[31] D.D. Holm and B.A. Kupershmidt. A multipressure regularization for multiphase flow. *Int. J. Multiphas. Flow*, 12(4):681 – 697, 1986.

[32] M. Ishii. *Thermofluid dynamic theory of two-phase flow*. Eyrolles, Paris, France, 1975.

[33] M. Ishii and T. Hibiki. *Thermo-fluid dynamics of two-phase flow*. Springer, 2nd edition, 2010.

[34] M.J. Ivings, D.M. Causon, and E.F. Toro. On Riemann solvers for compressible liquids. *Int. J. Numer. Meth. Fl.*, 28(3):395 – 418, 1998.

[35] J. Jakeman. *On numerical solutions of the shallow water wave equations*. PhD thesis, B. Sc. Thesis, Dept. Mathematics, Australian National Univ., Canberra, Australia, 2006.

[36] C.T. Kelley. *Solving Nonlinear Equations with Newton's Method*. Fundamentals of Algorithms. SIAM, 2003.

[37] K.H. Kim, C. Kim, and O.H. Rho. Methods for the accurate computations of hypersonic flows: I. ausmpw+scheme. *J. Comput. Phys.*, 174(1):38 – 80, 2001.

[38] K. Kitamura and M.S Liou. Comparative Study of AUSM-Family Schemes in Compressible Multiphase Flow Simulations. In *ICCFD7*, Big Island, Hawaii, July 9-13, 2012.

[39] K. Kitamura and E. Shima. Towards shock-stable and accurate hypersonic heating computations: A new pressure flux for ausm-family schemes. *J. Comput. Phys.*, 245(0):62 – 83, 2013.

[40] C. Kleinstreuer. *Two-Phase Flow: Theory and Applications*. Taylor & Francis, 2003.

[41] D.A. Knoll, L. Chacon, L.G. Margolin, and V.A. Mousseau. On balanced approximations for time integration of multiple time scale systems. *J. Comput. Phys.*, 185(2):583 – 611, 2003.

[42] D.A. Knoll and D.E. Keyes. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *J. Comput. Phys.*, 193(2):357 – 397, 2004.

[43] D.A. Knoll, V.A. Mousseau, L. Chacon, and J. Reisner. Jacobian-Free Newton-Krylov Methods for the Accurate Time Integration of Stiff Wave Systems. *J. Sci. Comput.*, 25(1):213 – 230, 2005.

[44] D. Koeze. A Study of Possible Applications for Jacobian-Free Newton Krylov Methods in Nuclear Reactor Physics. Master's thesis, TU Delft, 2009.

[45] A. Kurganov, S. Noelle, and G. Petrova. Semidiscrete central-upwind schemes for hyperbolic conservation laws and hamilton–jacobi equations. *SIAM J. Sci. Comput.*, 23(3):707–740, 2001.

[46] C.B. Laney. *Computational gasdynamics*. Cambridge University Press, 1998.

[47] S. Latham. *A Free Surface Nonlinearity in the Simplified Steady-State Shallow Water Equations*. Australian National University, 1999.

[48] B. Leer. Flux-vector splitting for the euler equations. In E. Krause, editor, *Eighth International Conference on Numerical Methods in Fluid Dynamics*, volume 170 of *Lecture Notes in Physics*, pages 507–512. Springer Berlin Heidelberg, 1982.

[49] R.J. LeVeque. *Numerical methods for conservation laws*, volume 132. Birkhäuser Basel, 1992.

[50] R.J. LeVeque. *Finite-Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[51] J.A. Liggett. Fluid mechanics. 1994.

[52] M.S. Liou. A Sequel to AUSM: AUSM+. *J. Comput. Phys.*, 129(2):364 – 382, 1996.

[53] M.S. Liou. A sequel to ausm, part ii: Ausm+-up for all speeds. *J. Comput. Phys.*, 214(1):137 – 170, 2006.

[54] M.S. Liou and J.R. Edwards. AUSM schemes and extensions for low Mach and multiphase flows. *Lect. Ser. - Von Karman Inst.*, 3:A1 – A101, 1999.

[55] M.S. Liou and C.J. Steffen Jr. A new flux splitting scheme. *J. Comput. Phys.*, 107(1):23 – 39, 1993.

[56] C. Lu. Optimization of gmres based newton-krylov solvers for transient 1d heat transfer. *University of Waterloo, Co-op report submitted to AECL*, September 2012.

[57] R.W. Lyczkowski, D. Gidaspow, C.W. Solbrig, and E.D. Hughes. Characteristics and stability analyses of transient one-dimensional two-phase flow equations and their finite difference approximations. Technical report, Aerojet Nuclear Co., Idaho Falls, Idaho (USA), 1975.

[58] C. Moler. Experiments with matlab. *The MathWorks, Cochituate Place*, 24, 2011.

[59] V.A. Mousseau. A fully implicit, second order in time, simulation of a nuclear reactor core. In *Computational Fluid Dynamics, Neutronics Methods and Coupled Codes*, volume 4, pages 383–392, Miami, Florida, USA, July 17 - 20, 2006.

[60] B.R. Munson, D.F. Young, and T.H. Okiishi. *Fundamentals of fluid mechanics*. Wiley, 1990.

[61] Y.Y Niu, Y.C Lin, and C.H. Chang. A further work on multi-phase two-fluid approach for compressible multi-phase flows. *Int. J. Numer. Meth. Fl.*, 58(8):879 – 896, 2008.

[62] S. Osher. Riemann solvers, the entropy condition, and difference. *SIAM Journal on Numerical Analysis*, 21(2):217–235, 1984.

[63] H. Paillere, C. Corre, and J.R. Garcia Cascales. On the extension of the AUSM+ scheme to compressible two-fluid models. *Comput. Fluids*, 32(6):891 – 916, 2003.

[64] J.D. Ramshaw and J.A. Trapp. Characteristics, stability, and short-wavelength phenomena in two-phase flow equation systems. *Nucl. Sci. Eng*, 66:93–102, 1978.

[65] V.H. Ransom. NUMERICAL BENCHMARK TEST NO. 2.1: FAUCET FLOW. *Multiphase Sci. Technol.*, 3(1-4):465 – 467, 1987.

[66] V.H. Ransom. NUMERICAL BENCHMARK TEST NO. 2.2: OSCILLATING MANOMETER. *Multiphase Sci. Technol.*, 3(1-4):468 – 470, 1987.

[67] V.H Ransom and D.L Hicks. Hyperbolic two-pressure models for two-phase flow. *J. Comput. Phys.*, 53(1):124 – 151, 1984.

[68] A. Ritter. Die fortpflanzung de wasserwellen. *VDI-Z*, 36(33):947–954, 1892.

[69] D.J. Robbins, R.S. Cant, and L.F. Gladden. Development of accurate, robust liquid equations of state for multi-phase CFD simulations with a modified AUSM+-up scheme. *Comput. Fluids*, 77(0):166 – 180, 2013.

[70] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[71] Y. Saad and M. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. Stat. Comp.*, 7(3):856 – 869, 1986.

[72] R. Saurel and R. Abgrall. A Multiphase Godunov Method for Compressible Multifluid and Multiphase Flows. *J. Comput. Phys.*, 150(2):425 – 467, 1999.

[73] O. Schenk and K. Gärtner. PARDISO User Guide Version 5.0.0. *URL: http://www. pardiso-project. org*, 2014.

[74] E. Shima and K. Kitamura. Parameter-free simple low-dissipation ausm-family scheme for all speeds. *AIAA J.*, 49(8):1693–1709, Aug 2011.

[75] M. Siddique. Smoothing of cranknicolson scheme for the two-dimensional diffusion with an integral condition. *Appl. Math. Comput.*, 214(2):512 – 522, 2009.

[76] S. Soo. On one-dimensional motion of a single component in two phases. *Int. Multiphase Flow*, 3:79–82, 1976.

[77] J.L. Steger and R.F. Warming. Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods. *J. Comput. Phys.*, 40(2):263 – 293, 1981.

[78] H.B. Stewart. Stability of two-phase flow calculation using two-fluid models. *J. Comput. Phys.*, 33(2):259 – 270, 1979.

[79] H.B. Stewart and B. Wendroff. Two-phase flow: Models and methods. *J. Comput. Phys.*, 56(3):363 – 409, 1984.

[80] J.J. Stoker. *Water waves: The mathematical theory with applications*. Wiley-Interscience, 1957.

[81] J.H. Stuhmiller. The influence of interfacial pressure forces on the character of two-phase flow model equations. *Int. J. Multiphas. Flow*, 3(6):551 – 560, 1977.

[82] I. Tiselj and S. Petelin. Modelling of two-phase flow with second-order accurate scheme. *J. Comput. Phys.*, 136(2):503 – 521, 1997.

[83] E.F. Toro. *Riemann solvers and numerical methods for fluid dynamics*, volume 16. Springer, 1999.

[84] I. Toumi. A weak formulation of roe's approximate riemann solver. *J. Comput. Phys.*, 102(2):360 – 373, 1992.

[85] I. Toumi. An upwind numerical method for two-fluid two-phase flow models. *Nuclear Science and Engineering*, 123(2):147–168, 1996.

[86] I. Toumi and A. Kumbaro. An approximate linearized riemann solver for a two-fluid model. *J. Comput. Phys.*, 124(2):286 – 300, 1996.

[87] I. Toumi, A. Kumbaro, and H. Paillere. *Approximate Riemann Solvers and Flux Vector Splitting Schemes for Two-phase Flow*. Rapport CEA. CEA Saclay, Direction de l'information scientifique et technique, 1999.

[88] K. Turner and H.F. Walker. Efficient High Accuracy Solutions with GMRES(m). *SIAM J. Sci. Stat. Comp.*, 13(3):815–825, 1992.

[89] H.K. Versteeg and W. Malalasekera. Computational fluid dynamics. *Longman Scientific & Technical, Essex*, 1995.

[90] C. Wu, G. Huang, and Y. Zheng. Theoretical solution of dam-break shock wave. *J. Hydraul. Eng.*, 125(11):1210–1215, 1999.

[91] G. Yadigaroglu and R.T. Lahey Jr. On the various forms of the conservation equations in two-phase flow. *Int. J. Multiphas. Flow*, 2(56):477 – 494, 1976.