A FAST ALGORITHM FOR MODELLING MULTIPLE BUBBLES DYNAMICS

BUI THANH TU

(B.Sc, Vietnam National University)

A THESIS SUBMITTED FOR THE DEGREE OF

MASTER OF ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2005

Acknowledgements

My thesis is done with the support of many people. I would like to take this opportunity to express my deepest and sincere appreciation to them.

First, I would like to thank Prof. Khoo Boo Cheong and Dr. Hung Kin Chew, my supervisors, for their many suggestions and constant support during my research. They are wonderful people and their support makes this research possible.

Secondly, I would like to thank Dr. Evert Klaseboer for his guidance and suggestions. The many meetings with him help me to understand the Boundary Element Method and the implementation of the 3-D BEM bubble code.

I would like express my sincere thanks to Dr. Ong Eng Teo who helped me to understand the Fast Fourier Transform on Multipole (FFTM).

I would like to express my thanks to National University of Singapore (NUS) and the Institute of High Performance Computing (IHPC) which award the Research Scholarship to me for the period 2003–2005. The Research Scholarship was crucial to the successful completion of this project.

Finally, I am grateful to my parents and my friends for their love and supports.

National University of Singapore *Bui Thanh Tu*

January 2005.

Table of Contents

Summary

This work presents the development of a numerical strategy to combine the Fast Fourier Transform on Multipoles (FFTM) method and the Boundary Element Method (BEM) to study the dynamics of multiple bubbles physics in a *moving boundary problem*. The disadvantage of the BEM is to solve the boundary integral equation by generating a very dense matrix system which requires much memory storage and calculations. The FFTM method speeds up the calculation of the boundary integral equation by approximating the far field potentials with multipole and local expansions. It is demonstrated that FFTM is an accurate and efficient method. However, one major drawback of the method is that its efficiency deteriorates quite significantly when the problem is full of empty spaces if the multiple bubbles are well-separated. To overcome this limitation, a new version of FFTM Clustering is proposed. The original FFTM is used to compute the potential contributions from the bubbles within its own group, while contributions from the other separated groups are evaluated via the multipole to local expansions translations operations directly. We tested the FFTM Clustering on some multiple bubble examples to demonstrate its improvement in efficiency over the original method. The efficiency of the FFTM and FFTM Clustering allows us to extend the number of bubbles in a simulation.

Physical behavior of multiple bubbles is also presented in this work.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

A better understanding of the physics of multiple bubble dynamics is important for a wide range of applications including underwater warfare, biomedical and chemical processes [1]-[8]. Previous works have identified the violent collapse of multiple cavitation bubbles thereby causing damages to the solid surface. Various techniques have been developed to model the dynamics of bubble in a fluid. Among them, the Boundary Element Method (BEM) is touted as one of the most effective tools for solving the dynamical boundary value problem and can be found in Wang et at. [9], [10], Zhang et al. [11], [12], Rungsiyaphornrat et al. [13], Best and Kucera [17], Blake and Gibson [2], Guerri et al. [8], and others.

BEM has the distinct feature of reducing the problem dimension by one. In bubble-structure interaction, only the boundaries of the bubble and structure need to be discretized. The cost of preprocessing and mesh generation is thus greatly

reduced. But this advantage is often compromised by the fact that BEM solves the boundary integral equation by generating a very dense matrix system, which requires $O(N^2)$ memory storage requirements and $O(N^2)$ operations to solve with iterative methods [32, 33]. This poses new challenges to the simulations with the of large problems, such as in multiple bubble dynamics, where problem size can easily exceeds several thousands. Hence, this provides the motivation to search for more efficient methods that scale significantly better than $O(N^2)$. Generally, these methods are collectively known as the fast algorithms.

Various fast algorithms have been developed for solving the boundary integral equation in electrostatic problem including the Fast Mutipole Method (FMM) [28] -[30] and the fast Fourier transform on multipoles (FFTM) [40],[41]. These methods are specially used in electrostatics to calculate the potential due to all the charges of a large system. The efficiency of the former comes from the use of multipole approximations and the highly effective hierarchical structures, where multipole and local expansions are translated efficiently up/down the oct-tree during evaluation of far field potentials. As for the FFTM methods, the speedup is obviously gained from the use of the Fast Fourier Transform algorithms for computing the discrete convolutions.

In this thesis, the Fast Fourier Transform on Multipoles (FFTM) coupled with BEM is chosen to solve the boundary integral equation which governs the dynamics of the multiple bubbles. The developed algorithm based FFTM is employed. It is demonstrated that the said method is accurate and efficient. The FFTM is much faster than the traditional BEM for large-scale problems and allows the application

to a large number of bubbles with refined mesh. However, its efficiency deteriorates significantly when the problem is spatially sparse or full of voids, that is much of the problem domain is empty as for the case where the bubbles were placed widely apart. Here, we suggested a simple fix to improve on the situation with the clustering approach. The new algorithm called FFTM Clustering first identifies and groups closely positioned bubbles which is based on their relative separation distances. Then the elements interactions within the self contained groups are evaluated rapidly using FFTM, while the interactions among the different groups are evaluated directly via the multipole to local translation operations. It is demonstrated that the new approach performs significantly faster than the original FFTM method without compromising the accuracy.

1.2 Previous work

The bubble dynamics was observed almost century ago, when Rayleigh (1917) considered the growth and collapse of spherical bubble in an infinite fluid [18]. The physical observation of the dynamics of cavitation bubble was carried out by Benjamin and Ellis [19]. Their experimental results show the collapse of an asymmetric bubble near a rigid boundary in which the jet was directed towards the rigid boundary. Other contributions on cavitation bubble were made by Kling and Hamitt [21], Lauterborn and Bolle [20], and Gibson and Blake [2]. In their paper, Gibson and Blake show the interaction between the collapsing bubble and the free surface.

To provide more in-depth understanding on this subject, Chapman and Plesset

[22] to develop the numerical modelling of cavitation bubble near rigid boundary with the assumption that 1) the liquid is incompressible, 2) the flow is invisid, 3) the vapor pressure inside the bubble is uniform and constant, 4) the ambient fluid pressure remains constant with time, 5) surface tension effects are neglected. Their method of solving Laplace equation is via finite difference integration method. The results show the formation of bubble jet which can explain the damage caused by cavitation bubble collapse.

Benvir and Fielding [1] successfully used the approximate integral scheme to model the early stage of the collapse phase of cavitation bubble, but it fails to compute for the latter stages of the collapse phase. Subsequently an improvements to this method was introduced by Blake and Gibson [2]. Their method is able to show the growth and collapse of a cavitation bubble near a rigid boundary and a free surface.

An alternate numerical technique was also proposed by Guerri, Lucca and Prosperetti [8]. In their work, the now-familiar boundary integral numerical method was used for the simulation of non-spherical cavitation bubble in an inviscid, incompressible liquid. Their work shows the efficiency of Boundary Integral Method in solving the Laplace equation. In essence, Boundary Integral Method avoids solving the problem for the entire fluid domain and limits the calculation to the boundaries.

In recent years, more advanced techniques have been proposed to improve the simulation of bubble dynamics in three-dimensions using Boundary Element Method [11], [12]. The conventional BEM generates a dense matrix system, which requires $O(N^3)$ and $O(kN^2)$ (k is the number of iterations) operations if solved using direct

methods such as Gaussian Elimination and iterative method such as GMRES [25], respectively. The computation obviously becomes prohibitively large if the problem size N exceeds several thousands.

To improve the computation, various fast algorithms have been developed including the Fast Mutipole Method (FMM) [28] which is one of the most widely implemented algorithms. FMM was developed by Greengard and Rohnklin [28] for solving potential fields in a N-body system. Later on, Nabor and White [31] implemented it in electrostatic analysis, mainly to calculate the capacitance of complex three-dimensional structure.

The efficiency of FMM comes from the effective usage of the multipole and local expansions. Greengard and Roklin [28][29] developed a new version of FMM for the Laplace equation by using the diagonal form of translation operator with exponential expansion, which reduces the $O(p^4)$ scaling factor to $O(p^2)$, where p is the order of expansion. On the other hand, Elliott and Board reduced the scaling factor to $O(p^2 \log p)$ by performing Fast Fourier Transfrom (FFT) on the multipole and local expansions, which required special technique to deal with the numerical instability for large values of p. The brief overview the FMM is presented on the above. However, we employ the FFTM method [40], [41] instead of FMM method due to the fact that FMM require higher order of expansion than FFTM does. For example, FMM [29] needs $p = 8$ to achieve 3-digits accuracy, whereas FFTM [41] needs only $p = 2$.

Alternatively, using multipole expansion alone can give rise to a fast algorithm, generally known as the tree algorithm [27] [26]. The basic idea is very similar to FMM algorithm, except that local expansion is not used. Instead, the multipole

expansion is evaluated directly on the potential node point. Hence, to a certain extent, FMM can bee seen as an enhancement of the tree algorithm.

Another group of fast methods utilizes FFTs to accelerate the potential evaluation task. They include the particle-mesh-based approach [35] and the precorrected-FFT method [36]. Generally, these methods approximate a given distribution of charges by an equivalent system of smoothed charge distribution that fall on a regular grid. Subsequently, the potential at the grid points due to the smoothed charge distribution is derived by discrete convolution, which is done rapidly using FFTs. However, local corrections are required for the "near" charges evaluations because these potential contributions are not accurately represented by the grid charges.

Alternative fast method that can also perform the potential evaluation rapidly called Fast Fourier Transform on Multipole (FFTM) is proposed [40], [41]. This method arises from an important observation that the multipole-to-local expansions translation operator can be expressed as series of discrete convolutions of the multipole moments with their associated spherical harmonic function. And the FFT algorithm can be employed to evaluate these discrete convolutions rapidly. It is demonstrated that FFTM performs efficiently and accurately for solving the Laplace equation in electrostatics problem [40],[41]. In this thesis, the Fast Fourier Transform on Multipoles (FFTM) coupled with BEM is chosen to solve the boundary integral equation which governs the dynamics of the multiple bubbles.

1.3 Outline of contents

This dissertation is divided into six chapters. Each of them consists of various subsections. A brief summary for each chapter is given as follows.

In Chapter 1 (Introduction), the motivation and scopes of the present work are presented. There is a brief on the background of bubble dynamics simulation, in which attention is centered on bubble dynamics simulation using BEM. Past works on bubble dynamics and fast algorithms like FMM and FFTM are also presented. In Chapter 2, we outline the mathematical modelling of bubble dynamics and boundary element method (BEM). In Chapter 3, the basic theory of FFTM is reiterated with sufficient details required for the development of the following section. Here, we present some results of implementation of FFTM for single and multiple bubbles simulation and compare efficiency with previous methods in term of accuracy and efficiency. The single bubble arrangement was chosen to compare with the theoretical solution. Several arrangement of multiple bubbles not previously computed via the standard BEM are presented to further illustrate the feasibility with FFTM incorporated. In Chapter 4, the new version of FFTM Clustering is presented. Discussion on the accuracy and efficiency of FFTM Clustering is also given. Numerical results for several examples of multiple bubble configurations are shown in Chapter 5. Finally, conclusions are drawn and the directions for future work are discussed in Chapter 6.

Chapter 2

BEM for Bubble Simulation

2.1 Mathematical Formulations

Mathematical formulation is used to describe the physical problem under certain assumptions. The mathematical formulations are derived from the conservation laws of momentum and mass with other assumptions that flow is irrotational and incompressible. As such, the Bernoulli's and Laplace's equation are the basic governing equation used to model and describe the physical behavior of bubble dynamics in a fluid.

Take the z-axis direct in the fluid as along the direction of gravity, the pressure inside the fluid domain can be evaluated by the unsteady Bernoulli's equations:

$$p = p_\infty - \rho\frac{\partial \Phi}{\partial t} - \frac{1}{2}\rho|\mathbf{u}|^2 + \rho g z. \tag{2.1}$$

Here p is the pressure at a point in the fluid domain Ω, p_∞ is the reference pressure or

pressure at a large distance from the bubble on the plane $z = 0$ (which can be given as atmospheric pressure), ρ is the density of the liquid, g is the gravity acceleration and z is the vertical component of the position vector $\overrightarrow{\mathbf{r}}$.

We consider the evolution of gas bubble inside the fluid, as shown in Fig. 2.1. The pressure inside bubble is a combination of: vapor and non-condensing gas. The gas pressure inside the bubble is assumed to be uniform and adiabatic. Pressure inside the bubble is therefore computed by:

$$p = p_v + p_{g,0} \left(\frac{V_0}{V} \right)^{\lambda},$$

where p_v is vapor pressure of the bubble, V_0 is the initial volume, V is the current volume of the bubble, $p_{g,0}$ is the initial internal gas pressure, and λ is the adiabatic constant or specific heat. λ is set to 1.25 for the gaseous explosion products resulting from an TNT explosion [6], and equals 1.4 for an ideal diatomic gas.

For the explosion bubble, the vapor pressure is much smaller compared with the gas pressure term inside the bubble. Therefore, pressure inside bubble explosion is determined by:

$$p = p_{g,0} \left(\frac{V_0}{V} \right)^{\lambda}. \tag{2.2}$$

Assume that the fluid domain Ω is inviscid, incompressible and irrotational, the governing equation for such potential flow is the Laplace equation:

$$\nabla^2 \Phi = 0, \tag{2.3}$$

where Φ is the velocity potential. The velocity must be a gradient of the potential which we define as:

$$\mathbf{u} = \nabla\Phi. \tag{2.4}$$

Provided that either the potential Φ (Dirichlet condition) or the normal velocity, $\partial\Phi/\partial n$ (Neumann condition) is given on the boundaries of the problem, solution can always be computed since (2.3) is an elliptic equation. According to the Green's theorem, the potential at every point on the boundary $\partial\Omega$ must satisfies the boundary integral equation:

$$c(\mathbf{x})\Phi(\mathbf{x}) = \int_{\partial\Omega} \left(\frac{\partial\Phi(\mathbf{y})}{\partial\mathbf{n}} G(\mathbf{x},\mathbf{y}) - \Phi(\mathbf{y}) \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial\mathbf{n}} \right) \mathrm{dS}, \tag{2.5}$$

where $\mathbf{x}$ is the field point and $\mathbf{y}$ is the source point on boundary Ω; The normal derivative at the boundary Ω is given by $\partial/\partial\mathbf{n} = \mathbf{n}\cdot\nabla$, where $\mathbf{n}$ is directed out of the fluid. The solid angle at location $\mathbf{x}$ is represented by c($\mathbf{x}$):

$$c(\mathbf{x}) = \begin{cases} 2\pi, & \mathbf{x} \in \partial\Omega \\ 4\pi, & \mathbf{x} \in \Omega \end{cases} \tag{2.6}$$

G is the Green function or kernel function defined in a three-dimensional domain as:

$$G(\mathbf{x},\mathbf{y}) = \frac{1}{|\mathbf{x}-\mathbf{y}|}, \tag{2.7}$$

and the normal derivative of the Green function can be written as:

$$\frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}} = \mathbf{n}\cdot\nabla\left(\frac{1}{(\mathbf{x}-\mathbf{y})}\right) = \mathbf{n}\cdot\frac{\mathbf{x}-\mathbf{y}}{|\mathbf{x}-\mathbf{y}|^3}. \tag{2.8}$$

An axis-symmetrical formulation as described in Wang et al. [10] can be used. Liu and Rudolphi [42] suggested new identities for the fundamental solution that can be imposed in the integral representation of internal problem to eliminate the strongest singularity. Equation (2.5) is integrated using linear approximation of the value of normal velocity and potential at the discretized point on the boundary. This procedure gives us a system of linear equations

$$\mathbf{G}\frac{\partial \Phi}{\partial \mathbf{n}} = \mathbf{H}\Phi. \tag{2.9}$$

Matrices $\mathbf{G}$ and $\mathbf{H}$ contain singular integrals with respect to the Green's function. Once the distribution of potential on the boundary is known, its normal derivative can be calculated by solving the matrix Eq. (2.9). The unsteady Bernoulli equation valid at the bubble interface is written as:

$$p = p_{ref} - \rho\frac{\partial \Phi}{\partial t} + \frac{1}{2}\rho|\mathbf{u}|^2 + \rho g(z-H), \tag{2.10}$$

where p_{ref} is the referenced pressure, and H is the depth of the inception of the bubble with respect to the z-datum.

For non-dimensionalization purpose, it is suitable to choose the maximum radius R_m of the Rayleigh bubble to be the scaling factor. According to Wilkerson [5],

the maximum radius R_m of the bubble is obtained in an infinite fluid domain under the uniform pressure p_∞. The reference pressure at the depth H is defined as: $p_{ref} = p_\infty + \rho g H$, where g is the gravitational acceleration and p_∞ is the atmospheric pressure, and ρ is the density of the surrounding fluid. The other scaling factors for the velocity potential and time are $\Phi_{ref} = R_m\sqrt{p_{ref}/\rho}$ and $t_{ref} = R_m\sqrt{\rho/p_{ref}}$, respectively. A dimensionless strength parameter ε is defined by the ratio of the initial gas pressure inside the bubble, p_0, at the inception and the hydrostatic pressure, p_{ref}, at the depth H: $\varepsilon = p_0/p_{ref}$. Pressure inside bubbles in dimensionless form is:

$$p' = \varepsilon \left(\frac{V_0'}{V'}\right)^{\lambda}. \tag{2.11}$$

Here the superscript ('), and henceforth, denotes the dimensionless variable. V_0' and V' are the initial and instantaneous volumes of the gas bubble in dimensionless form, and λ is the ratio of specific heats.

On the solid surface, the boundary condition in dimensionless form is expressed with the normal velocity set equal to zero:

$$\frac{\partial \Phi'}{\partial \mathbf{n}'} = 0. \tag{2.12}$$

The bubble deforms with time according to the following kinematic and dynamic

boundary conditions:

$$\frac{D\mathbf{x}'}{Dt'} = \nabla\Phi' \tag{2.13}$$

$$\frac{D\Phi'}{Dt'} = 1 - \varepsilon\left(\frac{V_0'}{V'}\right)^{\lambda} + \frac{1}{2}\nabla'\Phi'^2 + \delta^2(z' - \gamma), \tag{2.14}$$

where the vector $\mathbf{x}$ denotes the spatial position of the discretized point on the bubble interface; the terms $\delta = \sqrt{\rho g R_m / p_{ref}}$ and $\gamma = H/R_m$ are non-dimensional parameters which characterize the buoyancy and the initial inception position, respectively.

In Eq. (2.14), the term κ/We can be added on the right side to incorporate the discontinuity in pressure across the bubble surface due to surface tension effects, where κ is the sum of the principal curvatures of the bubble surface and the Weber number, $We = R_m p_0/\sigma$, is a function of the surface tension σ. As shown by Rungsiyaphornrat et al. [13], surface tension has a very small influence on the calculations for explosion bubbles, giving almost indistinguishable results.

2.2 Initial Conditions

It is shown by Rungsiyaphornrat [13] that there is an empirical formulation for estimating the maximum radius of the bubble in term of the amount of charges (TNT) used, W, and the depth of explosion, H, which is expressed as:

$$R_m = 3.38\left(\frac{W}{H + 10}\right)^{1/3}. \tag{2.15}$$

Another empirical relationship for the pressure of the explosion products of TNT is:

$$p_0 = 1.39 \times 10^5 \left(\frac{W}{V_0}\right)^{\lambda}. \tag{2.16}$$

The bubble is assumed to be initially spherical, thus

$$V_0 = \frac{4}{3}\pi R_0^3. \tag{2.17}$$

As pointed out in Best and Kucera [17], this assumes that the initial radial velocity of the explosion bubble is zero, with the motion driven from rest by very high initial partial pressure p_0. The initial velocity potential on the bubble surface is assumed to be zero. The initial radius of the bubble is chosen such that the maximum radius to which the bubble would expand in an infinite fluid is unity and this value is obtained from the solution of the Rayleigh-Plesset equation (written in dimensionless form)

$$R'\frac{\mathrm{d}^2R'}{\mathrm{d}t'^2} + \frac{3}{2}\left(\frac{\mathrm{d}R'}{\mathrm{d}t'}\right)^2 = \varepsilon\left(\frac{R'_0}{R'}\right)^{3\lambda} - 1, \tag{2.18}$$

where R'_0 and R' are the initial and instantaneous radius of the gas bubble in dimensionless form. The initial gas pressure p_0 can be calculated with Eq. (2.16). In summary, all the parameters for the model are known. For example a 500 kg TNT explosion at a depth of 100 m gives rise to a maximum bubble radius $R_m = 5.6m$ and $p_0 = 1.07 \times 10^8 Pa$. The dimensionless parameters for this case are $\lambda = 1.25, \varepsilon = 98.49$, and $R'_0 = 0.1495$. A typical time scale is $t_{ref} = 0.17s$.

At the inception time $t = 0$, the free surface is quiescent. Therefore, the initial

velocity and potential are equal to zero.

2.3 Mesh discretization

If the geometry is simple the integral can be approximated directly. However, with the even more complex geometry such as the spherical bubble surface, we have to discretize the boundary to approximate the integral in Eq. (2.5) accordingly. In this part, we present the grid generation on the bubble surface in a three dimension problem.

The icosahedron is taken to be the approximated bubble shape at the first level of discretization. This shape consists of 20 equal-sized equilateral triangles and 12 nodes, all of which lie on the surface of a sphere (see Fig. 2.2-a). The mesh can be refined by dividing each of original triangles into small one and projecting the new nodes on the spherical surface. To generate a second mesh level, the original triangle is divided into 4 sub-triangles. On each edge of the original triangles at the first level of approximation, the midpoint is chosen. From the midpoint we draw a line perpendicular to the original triangle which the node belong to. The intersection of this line and the spherical surface is the new node. The process of level two refinement is plot in the Fig. 2.2.

In the discretization of mesh level three, each side of the original triangle is divided equally by two new points and the triangle of the first approximation is divided into 9 sub-triangles in Fig. 2.3. From the new points on the apexes of the 9 sub-triangles, we plot the line perpendicular to the origin triangles. The intersections between

theses line and the spherical surface are the location of new nodes on the bubble surface. Fig. 2.3 shows the bubble grid surface at mesh level 6.

The higher mesh level is obtained by dividing more sub-triangles with more nodes on each side of each original triangle. The level of mesh is defined as the number of nodes on each side of the original triangle. The total number of nodes N_n is obtained from the relation $N_n = 10 * n^2 + 2$, where n is an integer representing the mesh level refinement. Similarly, the total number of triangles is determined from the relation $N_t = 20 * n^2$. Fig. 2.4 shows the bubble surface with mesh discretization at various levels of $4, 5, 6, 7, 8$ and 12. The higher the mesh level generation the more accurate approximation of the bubble shape is possible.

2.4 Numerical Procedures

The physical state of the problem is fully specified when the position of the boundary $\partial\Omega$ and the distribution of the velocity potential Φ are known. With these, the velocity $v = \nabla\Phi = (\partial\Phi/\partial\mathbf{n},\ \partial\Phi/\partial\tau)$ on the nodes are evaluated. The normal velocity $\partial\Phi/\partial\mathbf{n}$ can be obtained directly by solving the boundary integral equation (2.5). The tangential velocity $\partial\Phi/\partial\tau$ is estimated using the interpolation of potential values at the nodes on the boundary. Then, Equations (2.13) and (2.14) are numerically integrated in time to find the new position of the boundary and a corresponding new velocity potential Φ. The forward time integration is carried out using the 2^{nd}-order predictor-corrector scheme. To maintain the stability of the solution, the time-step size is controlled such that the changes in potential are bounded at each time step.

Each subsequent time step forward is calculated using [10]:

$$\Delta t' = \frac{\Delta\Phi'}{\max|1 + \frac{1}{2}|\mathbf{u}'|^2 + \delta^2(z' - \gamma) + \varepsilon\left(\frac{V_0'}{V'}\right)^{\lambda}|}. \tag{2.19}$$

This procedure ensures that the maximum potential change anywhere is limited to $\Delta\Phi'$ in order to avoid the development of numerical instabilities. Unless otherwise stated $\Delta\Phi' = 0.003$ is used in this work. This procedure ensures that the change in the potential at each node on the boundary is bounded by $\Delta\Phi'$. Details of the overall implementation are given in [9], [10], [12].

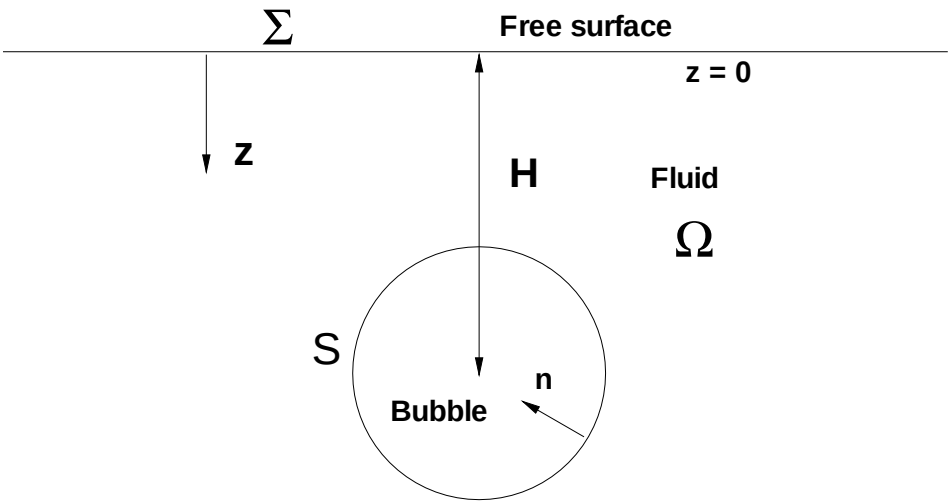

Figure 2.1: Bubble in Cartesian coordinate system.

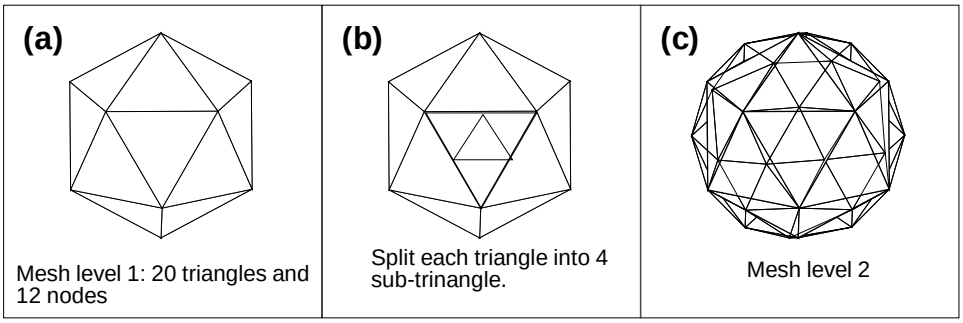

Figure 2.2: Mesh refinement at level 2.

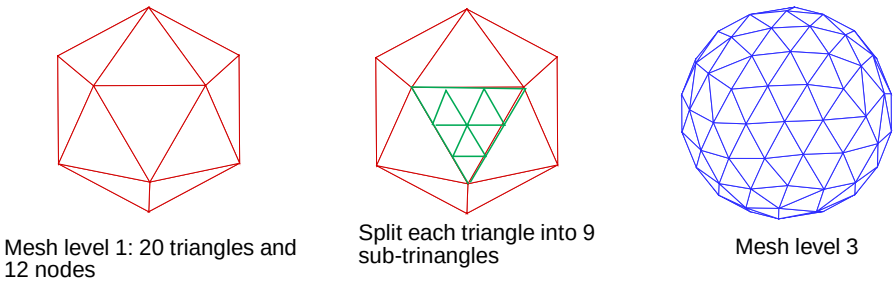

Figure 2.3: Mesh refinement at level 3.

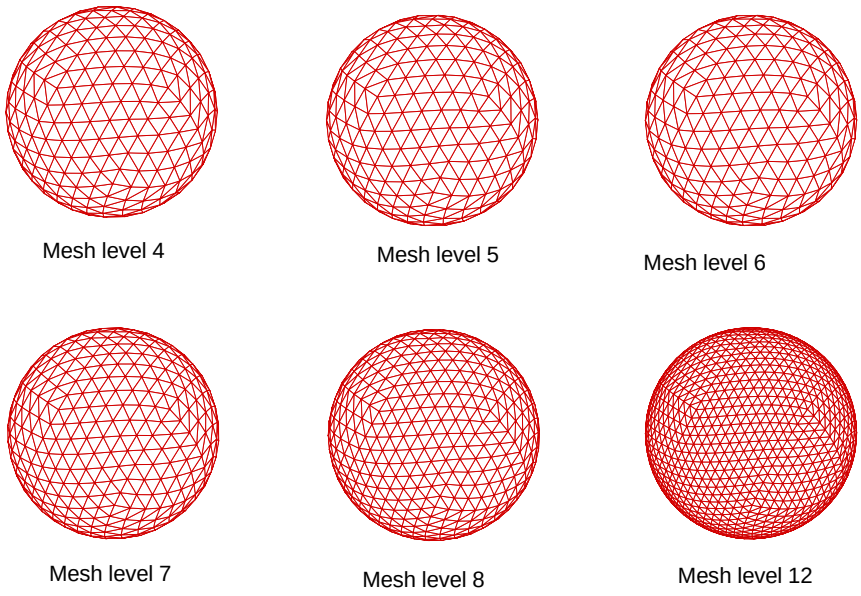

Figure 2.4: Mesh refinement at higher level.

Chapter 3

Bubbles simulation using FFTM

3.1 Implementation of FFTM in bubbles simulation

First we will briefly present the implementation of BEM. The implementation of BEM for solving boundary equation (2.5) comprises the following step: 1) Boundary element discretization, 2) solving the dense linear matrix system of equations generated by BEM.

The starting point of the discretization process consists of approximating the boundary by a set of elements, such that: $\Gamma = \bigcup_e^{\Gamma_e}$, where Γ is boundary of the problem domain: $\Gamma = \partial\Omega$. Both the geometry and field variables are approximated by simple basis function, that is

$$\widehat{\Phi}(\mathbf{x}) = \sum_{i=1}^{N} \tilde{\Phi}(\mathbf{x})\Psi_i(\mathbf{x}) \tag{3.1}$$

where $\widehat{\Phi}(\mathbf{x})$ is represented as a linear combination of a set of N linearly independent expansion function $\Psi_i(\mathbf{x})$ that is weighed by $\tilde{\Phi}(\mathbf{x})$ at N discrete points. After the discretization, (2.5) becomes:

$$c(\mathbf{x})\Phi(\mathbf{x}) = \sum_{i=1}^{N} \int_{\Gamma_i} \left(\frac{\partial \Phi_k}{\partial \mathbf{n}} \Psi_i^{\Phi}(\mathbf{y}) G(\mathbf{x}, \mathbf{y}) \right) \mathrm{d}\Gamma - \sum_{i=1}^{N} \int_{\Gamma_i} \left(\Phi_k \Psi_i^{V}(\mathbf{y}) \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}} \right) \mathrm{d}\Gamma, \tag{3.2}$$

where $\Psi_i^V(\mathbf{y})$ and $\Psi_i^{\Phi}(\mathbf{y})$ are the expansion functions of potential and its normal potential respectively; $\mathbf{x}$ and $\mathbf{y}$ correspond to the field and source points, respectively.

Applying the boundary conditions, the problem is reduced to a dense linear system of equations:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.3}$$

where $\mathbf{A}$ is a fully-populated $N \times N$ matrix.

Standard BEM (Std-BEM) solves the dense linear system (3.3) with iterative methods [32], [33], such as Gauss Iterative method and Generalized Minimal Residual (GMRES). These solvers need to compute a dense matrix-vector multiplication at every iteration. It is noted that the multiplication step is equivalent to the potential field calculation of (3.2), which comprises the single layer and dipole layer sources (first and second terms in (3.2), respectively). It is also noted that this operation is the most computationally expensive part of the method. However, this process is matrix-free, that is, the dense coefficients matrix need not be explicitly formed. FFTM is a fast algorithms which exploits this feature to accelerate the resolution of the dense linear system with various approximations techniques. This algorithm

speeds up the resolution of the boundary integral equation by approximating the far field potentials with multipole and local expansions.

In this work, we will couple the FFTM with BEM to simulate the dynamics of bubbles. We briefly present the algorithm of FFTM for moving boundary problem simulation. At beginning of each step, all the source points on the bubble surface are known. Then, FFTM is employed to calculate the normal velocity of the bubble surfaces by solving the dense linear system (3.3). The location of every node on the boundary is updated from values of all nodes computed by FFTM. The final calculation entails the updating of bubble volume, pressure and others physical values of bubbles. As noted above, we can use FFTM to simulate a problem with moving boundary.

In the following section, we present the multipole expansions approximation method, which is the basis of the FFTM algorithms.

3.2 Multipole translation theory for Laplace equation

3.2.1 Inner and Outer functions

First, we define two basic functions to simplify the mathematical expressions for the various translation operators used in the multipole approximation method. They are

referred to as the inner and outer functions, and given respectively by:

$$I_m^n(x) = \frac{r^n Y_n^{-m}(\theta, \phi)}{\sqrt{(n - |m|)(n + |m|)}}, \tag{3.4}$$

$$O_m^n(x) = \sqrt{(n - |m|)(n + |m|)} \frac{Y_n^{-m}(\theta, \phi)}{r^{n+1}}, \tag{3.5}$$

where $\mathbf{x} = (r, \theta, \phi) \in R^3$, and $Y_n^{-m}(\theta, \phi)$ corresponds to the spherical harmonic function of degree n and order m [44], given by:

$$Y_n^m(\theta, \phi) = \sqrt{\frac{(n - |m|)}{(n + |m|)}} P_n^{|m|}(cos(\theta)) e^{im\phi}, \tag{3.6}$$

and $P_n^{|m|}(cos(\theta))$ is the associated Legendre functions of the first kind with degree n and order m. Recurrence formulas for these two functions can be obtained from the recurrence relations of the associated Legendre functions [31], and they are summarized in the Appendix.

3.2.2 Multipole and local expansions and their translation operators

Multipole expansion

Given a set of n_e elements, be it single layer or dipole layer sources, which are clustered about the origin and bounded with a sphere of radius a. The potential generated at some arbitrary field point $\mathbf{y}$, where $\|\mathbf{y}\|$ is significantly larger than a,

can be approximated by the multipole expansion,

$$\Phi(\mathbf{y}) = \sum_{n=0}^{p} \sum_{m=-n}^{n} M_n^m O_n^m(\mathbf{y}), \tag{3.7}$$

where p is the order of expansion, and the multipole moments M_n^m is given by:

$$M_n^m = \sum_{i=1}^{n_e} \int_{\Gamma_i} \frac{\partial \Phi}{\partial n}(x_i) I_n^m(x_i) d\Gamma(x_i), \quad \text{for the single layer source,} \tag{3.8}$$

and

$$M_n^m = \sum_{i=1}^{n_e} \int_{\Gamma_i} \Phi(x_i) \nabla I_n^m(x_i) \mathbf{n} d\Gamma(x_i), \quad \text{for the dipole layer source.} \tag{3.9}$$

Equations 3.8 and 3.9 can be evaluated numerically with the standard Gaussian quadrature schemes [45], since both the integrands are non-singular functions. Recurrence formulas for the spatial derivatives of the inner function in 3.9 are also derived and presented in the Appendix.

Formula (3.8) or (3.8) defines the linear operator converting a system of source points that is arbitrarily distributed within a sphere centered at the origin into its multipole moments M_n^m defined at the origin. This linear mapping is denoted by T_{S2M}.

Local expansion

Given a set of n_e elements, be it single layer or dipole layer sources, which are located outside a sphere S_a of radius a. The potential they generate at some arbitrary field

point $\mathbf{y}$, where $\|\mathbf{y}\|$ is significantly smaller than a, can be approximated by the local expansion,

$$\Phi(\mathbf{y}) = \sum_{j=0}^{p} \sum_{k=-j}^{j} L_j^k I_j^k(\mathbf{y}), \tag{3.10}$$

where the local expansion coefficients, which are essential the potential gradients defined about the origin, are given by:

$$L_j^k = \sum_{i=1}^{n_e} \int_{\Gamma_i} \frac{\partial \Phi}{\partial n}(x_i) O_n^m(x_i) d\Gamma(x_i), \quad \text{for the single layer source} \tag{3.11}$$

and

$$L_j^k = \sum_{i=1}^{n_e} \int_{\Gamma_i} \Phi(x_i) \nabla O_n^m(x_i) \mathbf{n} d\Gamma(x_i), \quad \text{for the dipole layer source.} \tag{3.12}$$

The physical interpolations of local expansion coefficients correspond to the potential and its gradient evaluated at the origin, which are generated by N source points outside S_a. Formula (3.10) is the linear operator converting the local expansion coefficients defined at the origin into the potential at any arbitrary points $\mathbf{y} \in S_a$. This linear mapping is denoted by T_{L2P}.

Note that the local expansion is not used directly in the FMM and FFTM algorithms, and hence we did not derive the spatial derivatives of the outer functions. Nevertheless, we have presented it here so as to keep the multipole expansion theory complete.

3.2.3 Translation of multipole expansion

Given a set of n_e elements which are clustered around some arbitrary point $\mathbf{x}$, and their multipole moments evaluated with respect to $\mathbf{x}$ (using (3.8) or (3.9)) is denoted by N_n^m. Then the multipole moments M_j^k for the same set of elements but defined at another point $\mathbf{y}$ can be derived from N_n^m by using the following multipole to multipole translation operator

$$M_j^k = \sum_{n=0}^{j} \sum_{m=-n}^{n} i^{|k|-|m|-|k-m|} I_n^m(\mathbf{x}-\mathbf{y}) N_{j-n}^{k-m}. \tag{3.13}$$

Formula 3.13 defines the linear operator converting the multipole expansion coefficients N_n^m into the multipole expansion coefficients M_j^k. This linear mapping is denoted by T_{M2M}.

3.2.4 Conversion of multipole expansion to local expansion

Given a set of n_e elements which are clustered around some arbitrary point $\mathbf{x}$, and their multipole moments evaluated with respect to $\mathbf{x}$ (using (3.8) or (3.9)) is denoted by M_n^m. Then the local expansion coefficients L_j^k due to M_n^m at some arbitrary distant point $\mathbf{y}$ can be obtained by the following multipole moments to local expansion coefficients conversion operator

$$L_j^k = \sum_{n=0}^{p} \sum_{m=-n}^{n} (-1)^n i^{|k|-|m|-|k-m|} O_{j+n}^{m-k}(\mathbf{x}-\mathbf{y}) M_n^m. \tag{3.14}$$

Formula 3.14 defines the linear operator converting the multiple moments M_n^m into the local expansion coefficients L_j^k. This linear mapping is denoted by T_{M2L}.

It is mentioned here that this operation is a series of discrete convolutions of the multipole moments and the signed outer functions. The FFTM algorithm exploits this feature to speed up the computation of (3.14) by using FFT algorithms [45]. A more detail description of the FFTM algorithm is given later in section 3.3.

3.2.5 Translation of local expansion

Given the local expansion coefficients K_n^m defined with respect to some point $\mathbf{x}$, which is due to a set of n_e elements that are located at faraway from $\mathbf{x}$ (computed using (3.11) and (3.12)). Then the local expansion coefficients L_j^k due to effects of the same set of elements but defined at another point $\mathbf{y}$ can be derived from K_n^m by using the following local to local translation operator

$$L_j^k = \sum_{n=0}^{p} \sum_{m=-n}^{n} (-1)^{n+j} i^{|k|-|m|-|k-m|} I_n^m(\mathbf{x}-\mathbf{y}) K_{j-n}^{k-m}. \tag{3.15}$$

Formula 3.15 defines the linear operator converting the local expansion coefficients K_n^m into the local expansion coefficients L_j^k. This linear mapping is denoted by T_{L2L}.

With this translation operator, we have completed the multipole expansion approximation theory. It is worth to mention that Greengard and Rokhlin [29] had devised new FMM algorithm that is based on a new exponential representations. The method was demonstrated to be more efficient than its original version [28], but at the expenses of having more complicated operations, such as rotating the

multipole and local coefficients, and also their transformations between the original spherical harmonics form and the exponential representation. We did not consider this new approach in this study because it is not suitable for the implementation of the FFTM algorithm. In the next section, we will describe the FFTM algorithm, and also perform some numerical studies to evaluate its accuracy and efficiency.

3.2.6 Accuracy of multipole expansion approximation

All the expressions given above are truncated to a finite order of expansion p, and the errors incurred by doing so are theoretical bounded by the following relation (3.7), (3.10):

$$|\Phi_{exact} - \Phi_{multipole}| \leq K \left(\frac{1}{c-1}\right) \left(\frac{1}{c}\right)^{p+1}, \tag{3.16}$$

where K is a constant related to the strengths of the sources field, and c is a measure of the relative distance between the centers of the multipole expansion (for sources fields) and local expansion (for the potential field points). It is important to note from (3.16) that the accuracy of the multipole expansion approximation improves when: (a) a higher order of expansion p is used, or/and (b) the separation between the expansions centers c increases. Then for a given order of expansion p, it was shown that FFTM can be significantly more accurate than FMM [28], [29], due to the second factor (b). In other words, FMM would need to use a higher order of expansion than FFTM in order to achieve the same level of accuracy. From the previous studies in [40], FFTM required $p = 2, 8$ to attain 3- and 6-digits accuracy, respectively. Whereas for FMM [29], it needs $p = 8$, 19 for the same accuracy levels,

respectively. However, one should not jump into the conclusion that FFTM is more efficient than FMM, because their computational complexities scale differently with p. We would like to remark that it is not our objective here to prove that FFTM is better FMM, or visa versa. Hence, there is no explicit comparison between the two algorithms.

3.3 Fast Fourier Transform on Multipoles (FFTM)

FFTM has been known as a fast method for evaluating the potential field rapidly. The main different between FMM and FFTM is that FFTM adopt the complicated hierarchical in FMM, which is the main procedure that provides the efficiency of FMM algorithms. Besides, FFTM used the Fast Fourier Transformation (FFT) to convert directly multipole expansion to local expansion at highest level. It does not require to compute the multipole and local expansion at lower level.

3.3.1 FFTM algolrithm

The FFTM algorithm comprises of the following four steps, which are more easily understood with the illustrations of a two-dimensional problem in Fig. 3.1. The steps for the FFTM algorithm are given below as follows:

A. **Spatial discretization:** this step subdivides the problem domain into many smaller cells, and assigns the elements among them, which are to be represented by multipole moments. It also helps to identify the "near" and "distant" elements with respect to any given cell. By definitions, two cells are considered as neighbors if they

share at least one common vertex. Hence, a cell has at most 27 nearest neighbors (in 3-D context), including itself. Only those elements that belong to the nearest neighboring cells of that given cell are considered "near", and their potential effects are deal with in the conventional way as in standard BEM. The rest of the elements are considered "distant" ones, and are approximated by multipole expansions.

B. **Conversion of elements sources to multipole moments:** this step converts the clustered groups of elements within each of the cells into multipole moments **M**. This is accomplished with the transformation operators T_{S2M} in (3.8) and (3.9), according to the type of element sources.

C. **Calculating local expansions due to multipole moments:** this step evaluates the local expansion coefficients **L** at all the cells centers due to the multipole moments. This is accomplished with the transformation operators T_{M2L} in (3.14).

Convolution theorem. Suppose $\{u_i\}$ and $\{v_i\}$ are sequences of period N with coincide Fourier transforms given by $\{U_i\}$ and $\{V_i\}$, respectively. Then the discrete Fourier transform of $\{u \otimes v\}$ is $\{U_k V_k\}$, where $\{u \otimes v = \sum_{j=0}^{N-1} u_j v_{j-m}\}$ denotes the cyclic convolution of $\{u_i\}$ and $\{v_i\}$.

The multipole moments to local expansion coefficients conversion operator (3.14) is convolution in nature, which can be written as:

$$L_s^t \simeq \sum_{n=0}^{p} \sum_{m=-n}^{n} \left[\sum_{x'} \sum_{y'} \sum_{z'} T_{s,n}^{t,m}(x-x', y-y', z-z') M_n^m(x', y', z') \right], \qquad (3.17)$$

where $\mathbf{y} = (y_1, y_2, y_3)$ and $\mathbf{x} = (x_1, x_2, x_3)$ are the centers of the local expansion cell and multipole moment cells, respectively, and $T_{s,n}^{t,m}(\mathbf{x}) = (-1)^n i^{|k-m|-|k|-|m|} O_{j+n}^{m-k}$.

Since these two sets of centriods coincide and are regularly positioned, the series of discrete convolutions can be evaluated rapidly using the FFT algorithms (c.f. discrete convolution theorem [34]). Note that (3.17) is used to compute the "distant" elements contributions only.

D. **Calculating potentials at nodal positions:** with the local expansion coefficients computed at all the cells centers using (3.17), the potentials at the nodal positions can be then computed using the local expansion (3.10) via transformation operator T_{L2P} in 3.11, and this accounts only for the effects of the "distant" elements. The "near" elements potential contributions are finally added onto the nodal points directly via the conventional BEM approach. With this completes the FFTM algorithm.

3.3.2 Algorithmic complexity of FFTM

This section give estimates on the time and memory complexities of the FFTM algorithm associated with order of expansion p. The analysis looks at two parts, namely at the initialization and iteration stages. At the initial stage, we are interested in the time complexity.

Complexity at initialization stage

The main computational cost at this stage is due to the formations of the various transformation matrices, which include T_{Q2M}, T_{P2P} and T_{Q2P} (defined in 3.2), and also computation of the response functions $T_{s,n}^{t,m}$ and their Fourier transforms.

The complexities for computing and storing T_{S2M} and T_{L2P} are $O((p+1))^2 N$, and

T_{Q2P} is $O(NMs)$, respectively, where s denotes the average number source points or elements on one cell, and M is the maximum number of near by neighboring cells. Computing the response funtions and their Fourier transforms have complexities $O((p+1))^4 N_c)$ and $O((p+1)^4 N_c log N_c)$, respectively, where N_c is the total number of cells used to discretize the problem. Finally, the total time and memory complexities at the initialization stage are given as

$$Time = O(2(p+1)^2 N + (p+1)^4 [N_c + N_c log N_c] + NMs), \quad (3.18)$$

$$Memory = O(2(p+1)^2 N + (p+1)^4 N_c + NMs). \quad (3.19)$$

Complexity at iteration stage

At the iteration stage, the main concern is the time complexity as it determines the efficiency of the algorithm. The memory complexity is considerably less as compared to the initialization stage. The major memory requirement are:

1. $O((p+1)^2 8 N_c)$ matrices to store the FFTs of the multipole moments M_n^m and local expansion L_s^t

2. One matrix for storing the basis vector generated by the GMRES at each iteration. Generally, the memory required is $O(K_{iters} N)$, where K_{iters} is the number of iterations for the solution to converge to the desired accuracy.

Time complexity at the iteration stage comprises the following components:

1. $O((p+1)^2 N)$operation to compute the multipole moments using T_{Q2M}, and to compute the "distant potential contribution using T_{L2P}

2. $O(2(p+1)^2(8N_c log 8N_c)) + (p+1)^4 8N_c)$ operations to compute the discrete convolutions, which includes performing $(p+1)^2$ FFTs of the multipole moments, $(p+1)^2$ times of complex multiplication of the associated pairs of multipole moment and the response functions.

3. ($O(NMs)$ operation to compute the "near" potential contribution using T_{Q2P}.

The total time and memory complexities for one iteration is given by:

$$\begin{aligned} Time &= O(2(p+1)^2 N + NMs + (p+1)^1 6N_c log(8N_c) + (p+1)^4 (8N_c)), \\ Memory &= O((p+1)^2 8N_c + K_{iters} N). \end{aligned}$$

3.4 Results and discussion

In this section, we present results of implementation of FFTM for single and multiple bubbles simulation and compare with the standard BEM in term of accuracy and efficiency. The single bubble arrangement was chosen to compare with the theoretical solution. Several arrangement of multiple bubbles not previously computed before via the standard BEM are presented to further illustrate the feasibility with FFTM incorporated.

3.4.1 Single bubble

Let us consider a single explosion bubble far away from any boundaries and no gravitational effect. The charge is given as 500 kg of TNT placed at a depth of 100 m away from the free water surface. The bubble reaches its maximum radius of

$5.6m$. The bubble is generated with an initial dimensionless radius of $R_0' = 0.149856$ and a strength parameter $\varepsilon = 97.526715$ and ratio of specific heat $\lambda = 1.25$. The initial condition and initial velocity is $\Phi' = 0$ at $t' = 0$ on the bubble surface. The initial dimensionless time step taken is $dt' = 0.000304$. At the first time step the dimensionless potential and dimensionless velocity are equal to zero: $\Phi'^{t'} = 0$ and $u' = 0$. The potential at the first time step is: $\Phi'^{t'+dt'} = \Phi'^{t'} + (1-\epsilon)dt'$. Solving Laplace equation $\nabla^2\Phi' = 0$ for spherical bubble, we have the relationship

$$\Phi'^{dt'+t'} = -\frac{\Phi'(dt')}{R_0'} = \frac{(1-\epsilon)dt'}{R_0}. \tag{3.20}$$

Formula (3.20) shows the normal velocity normal velocity at the first step. (The results do not include the error made by approximating the boundary integral equation (2.5) and error associated with solving the linear system equation (3.3). From formula (3.20), the exact solution for normal velocity at any point on the bubble surface is:

$$\frac{\partial\Phi'}{\partial\mathbf{n}}_{exact} = -0.196128. \tag{3.21}$$

The average error of the normal velocity between numerical results and exact results is measured in the L_2 norm as:

$$\text{Aver. Error} = \left(\frac{\sum_{i=1}^{N} |\frac{\partial\Phi'}{\partial\mathbf{n}}_{exact} - \frac{\partial\Phi'}{\partial\mathbf{n}}^{i}_{numr}|^2}{\sum_{i=1}^{N} |\frac{\partial\Phi'}{\partial\mathbf{n}}_{exact}|^2} \right)^{\frac{1}{2}} \times 100\%, \tag{3.22}$$

where $\frac{\partial\Phi'}{\partial\mathbf{n}}^{i}_{numr}$ is the value of the normal velocity of node i produced by the numerical methods of FFTM and standard BEM.

The bubble is discretized by 642 nodes and 1280 elements. We carried out different tests of FFTM with different values of order of expansion p which is defined in Section 3.2.5. The accuracy of the FFTM improves when a higher order of expansion p is used. The test is run on a single computer IBM p690 Regatta 1.3 GHz CPU speed, 250Mb RAM memory. The efficiency of FFTM is evaluated by the speed-up factor which is defined as:

$$\text{SP} = \frac{T_{std.BEM}}{T_{FFTM}}. \tag{3.23}$$

Here, $T_{std.BEM}$ and T_{FFTM} are the CPU time taken by standard BEM and FFTM on the above at first time step, respectively.

The standard BEM needed $T_{std.BEM} = 5.26$ seconds of CPU time of computation. The average error and maximum error of the standard BEM method compared to the exact results are 0.334% and 0.465%, respectively.

Table 3.1 shows the performance of FFTM method for differences value of p. It is obvious from Table 3.1 that higher order of expansion of FFTM gives more accurate results FFTM. Both the maximum and average errors pertaining to the FFTM tends towards to the standard BEM. The order of expansion $p = 4$ gives about an optimal result in term of accuracy and efficiency. The FFTM(4) (ie. p=4) is faster than the standard BEM by 4.87 times. The average error for the FFTM(4) and the standard BEM is 0.342% and 0.344%, respectively, as compared to the exact solution. The difference between the standard BEM and exact solution is attributed to the spatial discretization of the mesh approximating the bubble surface. A higher mesh would produce more accurate results. Henceforth, the performance of FFTM

will be carried out by choosing the order of expansion $p = 4$. Fig. 3.2 shows the comparison of normal velocities of 642 nodes on bubble surface for the FFTM(4) and standard BEM.

Next, we compare the numerical results computed by the FFTM and standard BEM with the analytic results of the Rayleigh bubble. Both numerical methods were carried out with 2200 steps and dimensionless time-step size is taken to be constant $\Delta t = 0.001$. This time-step size is small enough to ensure stability of the numerical scheme. The purpose of choosing constant time-step is to compare the results conducted by FFTM, standard BEM at the same time. The test was ran on a single computer IBM p690 Regatta 1.3 GHz CPU speed, 250Mb RAM memory. While, the standard BEM took 11572 seconds of CPU time to compute 2200 steps, the FFTM needed only 2376 seconds. In this sample, the FFTM performed faster than the standard BEM by about 4.9 times.

During the life time, the Rayleigh bubble remains spherical in shape. The bubble is initiated as a small sphere and it expands until reaches the maximum volume. Then, the bubble contracts symmetrically in the collapse phase. The evolution of bubble is shown in Fig. 3.3.

A comparison of the bubble radius in its evolution is made between the numerics and theoretical results (as governed by the Rayleigh–Plesset equation (2.18). Since the bubble remains spherical in shape throughout its lifetime, we can calculate its radius by computing its volume. Fig. 3.4 shows the bubble radius as a function of time obtained from the FFTM and standard BEM together with the solution of the Rayleigh Plesset equation (using the fourth-order RungeKutta method). As shown

in Fig. 3.4, the FFTM curve agrees well with the analytic curve and it practically coincides with the results from the standard BEM.

The dimensionless maximum radius of bubble calculated by the FFTM and the standard BEM are 1.006741 and 1.006739, respectively. The numerical errors procedured by the FFTM at the maximum bubble radius (which is 1.0 according to analytic solution), is thus 0.6741%, while for the BEM is 0.6739%. Both the maximum bubble radius and the oscillation period are well predicted by the FFTM and standard BEM with no more than 1% error or derivation from the exact solution. The numerical error in the maximum bubble radius between FFTM and standard BEM solver is 0.02%. This suggests that FFTM is a good approximation to the standard BEM solver.

Further comparison is made with respect to the relationship between the bubble surface velocity dR'/dt' and radius R'. The analytic solution is provided in the equation [7]:

$$\frac{3}{2}\left(\frac{\mathrm{d}R'}{\mathrm{d}t'}\right)^2 = -1 + R'^{-3}\left[1 + \frac{\varepsilon R_0'^{3\lambda}}{\lambda - 1}(1 - R'^{-3(\lambda-1)})\right]. \tag{3.24}$$

Fig. 3.5 shows the results of the numerics and analysis. There are three curves: the theoretical curve from analysis and the curves computed by the FFTM and the standard BEM. Numerical results provided by the FFTM mostly or effectively coincides with the standard BEM' resutls. It is apparent that there is reasonably good agreements between the analysis and numerics with slightly larger over prediction of dR'/dt' at the maximum and minimum. The analytic curve in Fig. 3.5

is symmetrical with respect to the R'-axis. It is perceptible from the same figure that the numerical results of positive section is not symmetrical to the negative section taken with respect to the R'-axis. The positive and negative values of dR'/dt' represent the expansion phase and collapse phase of bubble evolution, respectively. At the extreme of the negative part we have relatively larger error because error is accumulated during the bubble simulation all the way right from the beginning.

The results show that the bubble surface velocity reduces to zero when its radius attains the maximum value. The present numerical results on the Rayleigh bubble suggest that the FFTM is a fast and accurate solver. In the following simulation, we will use FFTM to solve multiple bubbles where the usual BEM method would have required inordinately much computational time and memory.

3.4.2 Multiple bubbles

In this section, we present both the efficiency and accuracy of FFTM as applied to the simulation of multiple bubbles. As there is no analytic solution for the different arrangement of multiple bubbles, the reference is taken from the standard BEM and the deviation for the FFTM solver is reckoned to be the error incurred. The standard BEM used iterative solvers like Gauss Iterative Method, Generalized Minimal Residual (GMRES) to solve the explicit matrix generated from Eq. (3.2). FFTM is an approximate method and does not form the explicit matrix directly; instead it forms the corresponding multipole moments and local moments and then solve the linear matrix system implicitly.

Performance of the FFTM on two bubbles with dimensionless distance between the centers of them initially at $d' = 2.6785$.

First, we consider two explosion bubbles far away from any boundaries and no gravitational effect. Each charge is given as 500 kg of TNT placed at a depth of 100 m away from the free water surface, and strength parameter $\varepsilon = 97.52$ and ratio of specific heat $\lambda = 1.25$. The centers of the two bubbles are initially placed on the x-axis at a distance of $d = 15$ m apart equivalent to dimensionless distance of $d' = 2.6785$. Each bubble is generated with 642 nodes and 1280 triangle elements on its surface. We will compare the normal velocity $\partial\Phi/\partial n$ on the bubbles surface between the FFTM and standard BEM at a typical time instance. Both the methods were carried out with 2200 time steps and the dimensionless time-step size is taken to be a constant at $\Delta t' = 0.001$. This time-step size is small enough to ensure stability of the numerical scheme.

At each time step, both the FFTM and standard BEM solve the boundary integral equation (3.2) and return the value of normal velocity $\frac{\partial\Phi'}{\partial\mathbf{n}}$ for all the nodes on the bubbles surface. The average error incurred for the FFTM solver is measured in the L_2 norm as:

$$\text{Aver. Error} = \left(\frac{\sum_{i=1}^{N} \left| \frac{\partial\Phi'}{\partial\mathbf{n}}^{i}_{FFTM} - \frac{\partial\Phi'}{\partial\mathbf{n}}^{i}_{std.BEM} \right|^2}{\sum_{i=1}^{N} \left| \frac{\partial\Phi'}{\partial\mathbf{n}}^{i}_{std.BEM} \right|^2} \right)^{\frac{1}{2}} \times 100\% \tag{3.25}$$

where $\frac{\partial\Phi'}{\partial\mathbf{n}}^{i}_{FFTM}$ and $\frac{\partial\Phi'}{\partial\mathbf{n}}^{i}_{std.BEM}$ are the normal velocity value of node i produced by FFTM solver and standard BEM, respectively. N is the total number of nodes. A quantitative comparison of the FFTM and standard BEM is given in Table 3.2.

As seen in Table 3.2, the percentage numerical error between FFTM and standard BEM solver during the simulation is less than 1% which is considered acceptable in most engineering applications.

Figures 3.7 and 3.8 show the bubbles shapes during its expansion and collapse phase based on FFTM and standard BEM solver, respectively. It is obviously that there is essentially no difference in the deformation of the bubbles shapes between the two methods. During the expansion phase, the bubbles expand rather symmetrically. When the maximum is reached, both the bubbles start to contract. During the contraction or collapse phase, not only are the bubbles contracting non-symmetrically with a jet developing or originating on the bubble surfaces on the outer sides away from the center, the bubbles are also attracted towards each other. The direction of the bubble induced jet points towards each other.

Next, the bubble volume is compared to test further the accuracy of the FFTM method. Figure 4.5 shows the dimensionless volume of one bubble produced by the FFTM and standard BEM. There are not much differences between the two methods. The standard BEM produces the maximum dimensionless volume V' at 4.3308 while the FFTM gives the corresponding V' at 4.3313 at dimensionless time $t' = 1.153$. FFTM is a good approximation solver that produces 0.012% error, compared with the standard BEM.

The tests were conducted on a single IBM p690 Regatta 1.3 GHz CPU speed. While the standard BEM took about 17843 seconds of CPU time to run 2200 steps, FFTM took only 3964 seconds. In this example, the FFTM performed faster than the standard BEM by about 4.5 times. While the standard BEM utilized 16.5 Mb

of memory storage, FFTM needed only 5.1 Mb; this is a savings of about 3.1 times. The efficiency of the FFTM at each time step is evaluated by the speed-up factor which is defined as:

$$\text{speed-up} = \frac{T_{std.BEM}}{T_{FFTM}}, \tag{3.26}$$

where, $T_{std.BEM}$ and T_{FFTM} are the CPU time taken by the standard BEM and FFTM at each time step, respectively.

Fig. 3.9 shows the time taken for one-time step computation of the two methods during bubble evolution. Fig. 3.10 shows the speed-up factor of the FFTM method during bubble simulation. It is obviously from this figure that FFTM runs faster than standard BEM throughout even at the initial stage. It is interesting to note that the time taken by FFTM reduces as the bubble volume increases, the speed-up factor increases when the bubbles expand. When the bubble volume reaches the maximum state at dimensionless time $t' = 1.153$, the FFTM solver took least time to solve the problem. After the bubbles reach their maximum volume, they start to contract. Then the percentage of blank space or void in the domain increases. This is the reason why the time required by the FFTM algorithm rises and the speed-up factor reduces beyond the bubble maximum volume.

Performance of the FFTM on two bubbles with dimensionless distance between centers of them initially at $d' = 5.357$.

Next, we consider two explosion bubbles from two charges with the centers further apart. The centers of the two bubbles are initially placed on the x-axis at a distance

of $d = 30$ m apart equivalent to dimensionless distance of $d' = 5.357$. Each charge is given as 500 kg of TNT placed at a depth of 100 m away from the free water surface, and strength parameter $\varepsilon = 97.52$ and ratio of specific heat $\lambda = 1.25$. The maximum bubble radius will be 5.6 m for the 500 kg TNT charge placed at a depth of 100 m away from the free water surface. Each bubble is generated with 642 nodes and 1280 triangle elements on its surface. Both the methods were carried out with 2200 time steps and dimensionless time-step size is taken to be constant at $\Delta t' = 0.001$. This time-step size is small enough to ensure stability of the numerical scheme.

To test the accuracy, we show the average error in normal velocity field computed by the FFTM and standard BEM. A quantitative comparison of the FFTM and standard BEM is provided in Table 3.3. The average error incurred is limited to less than 0.8%, while most of the time the error is less than 0.40%.

The evolution of the bubbles shapes from the FFTM calculation is shown in Fig. 3.11; the counterpart for the standard BEM is provided in Fig. 3.12 for comparison. Both the methods give almost the same bubbles shapes. It is obvious that during the expansion, the bubble expands rather symmetrically. When these reach the maximum, both the bubbles start to contract and keep to the fairly symmetrical shapes till about $t' = 2.109$.

Next, the bubble volume is compared to test further the accuracy of the FFTM method. Fig. 3.13 shows the dimensionless volume of one bubble produced by the FFTM and standard BEM. There are not much differences between the two methods. Both the methods produce the maximum dimensionless volume V' at 4.3083 at dimensionless time $t' = 1.077$.

The tests were conducted on a single IBM p690 Regatta 1.3 GHz CPU speed. While the standard BEM took about 17843 seconds of CPU time to run 2200 steps, the FFTM took 5916 seconds. In this example, the FFTM performed faster than the standard BEM by about 3 times. While the standard BEM utilized 16.5 Mb of memory storage, FFTM needed 9.3 Mb; this is a savings of about 1.78 times. In this test, the distance between the centers of the two bubbles increases by doubling, the speed-up factor reduces by 1.5 times and the saving memory reduces 1.74 times compared to previous test. Fig. 3.14 shows the time taken for one-time step computation of the two methods during bubble evolution. Fig. 3.15 shows the speed-up factor of the FFTM method during bubble simulation. (Here the speed-up factor term is defined as in 3.26). Similar to the previous test, the time taken by FFTM reduces as the bubble volume increases, the speed-up factor increases when the bubble expand. In each time step, when the bubbles are placed further apart, the FFTM solver requires more CPU time.

Performance of the FFTM on two bubbles with dimensionless distance between centers of them initially at $d' = 7.95$.

Next, the two explosion bubbles are positioned even further apart. Center of two bubble are initially placed on the x-axis at a distance $d = 45$ m apart equivalent to dimensionless distance of $d' = 7.95$. Each charge is given as 500 kg of TNT placed at a depth of 100 m away from the free water surface, and strength parameter $\varepsilon = 97.52$ and ratio of specific heat $\lambda = 1.25$. Each bubble is generated with 642 nodes and 1280 triangle elements on its surface. Both the methods were carried out with 2200

time steps and the dimensionless time-step size is taken to be constant at $\Delta t = 0.001$. This time-step size is small enough to ensure stability of the numerical scheme.

Once again, we show the average error in the normal velocities computed via the FFTM and standard BEM. A quantitative comparison of the FFTM and standard BEM is provided in Table 3.4. The average error incurred is limited to less than 0.95%, while most of the time the error is less than 0.40%.

The evolution of the bubbles shapes from the FFTM calculation is shown in Fig. 3.16; the counterpart for the standard BEM is provided in Fig. 3.17 for comparison. Both the methods give almost the same bubbles shapes. It is obvious that during the expansion, the bubble expands rather symmetrically. When these reach the maximum, both the bubbles start to contract and keep to the fairly symmetrical shape till about $t' = 2.109$.

Next, the bubble volume is compared to test further the accuracy of the FFTM method. Fig. 3.18 shows the dimensionless volume of one bubble produced by the FFTM and standard BEM. There are not much differences between the two methods. Both the two methods produce the maximum dimensionless volume V' at 4.310 at dimensionless time $t' = 1.050$.

The tests were conducted on a single IBM p690 Regatta 1.3 GHz CPU speed. While the standard BEM took about 17843 seconds of CPU time to run 2200 steps, the FFTM took 8874 seconds. In this example, the FFTM took longer time than previous tests. The FFTM performed faster than the standard BEM by only about 2 times. While the standard BEM utilized 16.5 Mb of memory storage, FFTM needed 18.0 Mb. In this test, FFTM required more memory storage than standard

BEM, and the time taken has also increased comparatively to previous FFTM versus standard BEM. Fig. 3.19 shows the time taken for one-time step computation of the two methods during bubbles evolution. Fig. 3.20 shows the speed-up factor of the FFTM method during bubble simulation. Similar to the previous test, the time taken by FFTM reduces as the bubble volume increases, the speed-up factor increases when the bubbles expand. In each time step, when the bubbles are placed further apart, the FFTM solver requires more CPU time.

Performance of the FFTM on three bubbles

Consider three charges placed under water far away from any boundaries and no gravitational effect. The location of initial first charge is $(0, 0, 0)$ and the weight of the first charge is 500 kg TNT placed at a depth of 100 m away from the free water surface. The location of second charge in the dimensionless value is $(2.68, 0, 0)$ and the weight of the second charge is 500 kg TNT placed at a depth of 100 m away from the free water surface. The location of the third charge is $(8.93, 5.36, 0)$ and the weight of the third charge is 2000 kg TNT. The maximum bubble radius will be 5.6 m for the 500 kg TNT charge placed at a depth of 100 m away from the free water surface. The maximum bubble radius will be 8.9 m for the 2000 kg TNT charge. The dimensionless initial radius of bubble for the 500 kg TNT charge is 0.1499. The dimensionless initial radius of bubble for the 2000 kg TNT charge is 0.2379. Some parameters for all the bubbles are strength parameter $\varepsilon = 97.52$ and ratio of specific heat $\lambda = 1.25$. Each bubble is generated with 362 nodes and 720 triangle elements on its surface. Both the methods were carried out with 2200

steps and the dimensionless time-step size is taken to be constant $\Delta t' = 0.001$. This time-step size is small enough to ensure stability of the numerical scheme.

A quantitative comparison of the FFTM and standard BEM for the normal velocity field is provided in Table 3.5. The maximum average error incurred is less than 1.01%, while most of the time the error is less than 0.60%.

Fig. 3.21-3.23 shows the variation of the dimensionless volume of the three bubbles with time produced by the FFTM and standard BEM. There is hardly any perceived difference between the two methods. For the first bubble, the standard BEM produces the maximum V' at 3.92210 while the FFTM gives $V' = 3.92215$ at $t' = 1.143$. The deviation is a only about 0.001%. For the second bubble, both methods produce the maximum $V' = 3.83493$ at $t' = 1.165$. For the third bubble, the standard BEM produces the maximum V' at 17.9816 while the FFTM gives $V' = 17.9818$ at $t' = 1.725$. The deviation is less than 0.001%. This show that FFTM is a good approximation solver, compared with the standard BEM.

For bubbles shapes comparison, there are not much difference between the results produced by the FFTM and standard BEM. Bubbles start to grow at the same time with small spherical shape with dimensionless radii of 0.1499. Fig. 3.24 shows the bubbles shapes at the dimensionless time $t' = 0.271$. At this time bubbles are in the expansion phase and three of them keep to their spherical shapes. Fig. 3.25 shows the bubbles shapes at the dimensionless time $t' = 2.221$. The first bubble forms a jetting directed towards the second bubble. The second bubble is strengthened by the effect of the third bubble. The simulation stop at the time $t' = 2.3014$. Fig. 3.26 shows the bubbles shapes at $t' = 2.3014$. Fig. 3.27 shows the shapes of the the two

bubbles at $t' = 2.3014$ for better view.

The tests were conducted on a single IBM p690 Regatta 1.3 GHz CPU speed. While the standard BEM took about 13231 seconds of CPU time to run 2200 steps, the FFTM took 6547 seconds. In this example, the FFTM performed faster than the standard BEM by about 2.0 times. Fig. 3.28 shows time taken using FFTM and standard BEM in each time step during the bubble evolution. It is obvious from Fig. 3.28 that as bubbles evolve, the time taken in each time step by the FFTM reduces rapidly while standard BEM does not change much. Fig. 3.29 shows the speed-up factor of the FFTM method during bubble simulation. The speed-up factor term is defined in 3.26. The speed-up factor increases until it reaches the maximum at dimensionless time around 1.25 and then it reduces. The main reason for the change in the trend of the speed-up factor is due to the changes of the blank spaces or voids between the bubbles in the domain. When the bubbles reach maximum volume, bubble become closer, therefore there are less blank spaces or voids in the computational domain. In this case, FFTM work more efficiently. During the bubbles dynamics simulation, there are relatively more blank spaces or voids between the bubbles during the initiate phase or in the contraction phase when the bubble volumes become smaller under those circumstances, the FFTM performed less efficiently. This is the drawback of FFTM that it does not work efficiently when there are too much black spaces or voids between the bubbles.

Table 3.1: Summary of performance of FFTM on single bubble with 642 nodes and 1280 elements at first step using IBM p690 Regatta, 1.3 GHz CPU speed, 250Mb RAM memory. Comparison of error is made with regards to exact solution.

Order of expansion	**CPU time (sec)**	**SP (times)**	**Aver. Error (%)**	**Max. Error (%)**
2	0.63	8.34	2.014	5.584
3	1.02	5.15	0.488	1.846
4	1.08	4.87	0.342	0.481
5	1.91	2.75	0.334	0.472
6	2.78	1.89	0.334	0.466

Table 3.2: Average error produced by FFTM of the normal velocities on two bubbles with separate distance between centers $d' = 2.6785$.

Dimensionless time	**Average Error (%)**
0.015	0.21
0.045	0.17
0.075	0.12
0.180	0.14
0.285	0.19
0.315	0.17
0.375	0.22
0.465	0.33
0.585	0.34
0.615	0.34
0.765	0.40
0.855	0.55
0.915	0.72
1.015	0.98
1.290	0.99
1.365	0.72
1.440	0.52
1515	0.42
1.665	0.34
1755	0.50
1845	0.43
1.965	0.58

Table 3.3: Average error produced by FFTM of the normal velocities on two bubbles with separate distance between centers $d' = 5.357$.

Dimensionless time	**Average Error (%)**
0.015	0.14
0.045	0.15
0.075	0.21
0.180	0.26
0.285	0.14
0.315	0.15
0.375	0.20
0.465	0.24
0.585	0.25
0.615	0.26
0.765	0.31
0.855	0.38
0.915	0.57
1.015	0.78
1.290	0.49
1.365	0.36
1.440	0.31
1515	0.26
1.665	0.28
1755	0.23
1845	0.23
1.965	0.31

Table 3.4: Average error produced by FFTM of the normal velocities on two bubbles with separate distance between centers $d' = 7.95$.

Dimensionless time	**Average Error (%)**
0.015	0.16
0.045	0.19
0.075	0.13
0.180	0.17
0.285	0.17
0.315	0.14
0.375	0.17
0.465	0.23
0.585	0.24
0.615	0.23
0.765	0.25
0.855	0.47
0.915	0.76
1.015	0.92
1.290	0.40
1.365	0.28
1.440	0.28
1515	0.30
1.665	0.36
1755	0.33
1845	0.29
1.965	0.59

Table 3.5: Average error of the normal velocities on three bubbles produced by the FFTM method.

Dimensionless time	**Average Error (%)**
0.015	0.21
0.045	0.21
0.075	0.12
0.180	0.12
0.285	0.20
0.315	0.21
0.375	0.22
0.465	0.23
0.585	0.27
0.615	0.30
0.765	0.34
0.780	0.34
0.915	0.49
1.065	0.76
1.290	0.101
1.365	0.74
1.440	0.60
1515	0.48
1.665	0.31
1755	0.34
1845	0.30
1.965	0.32

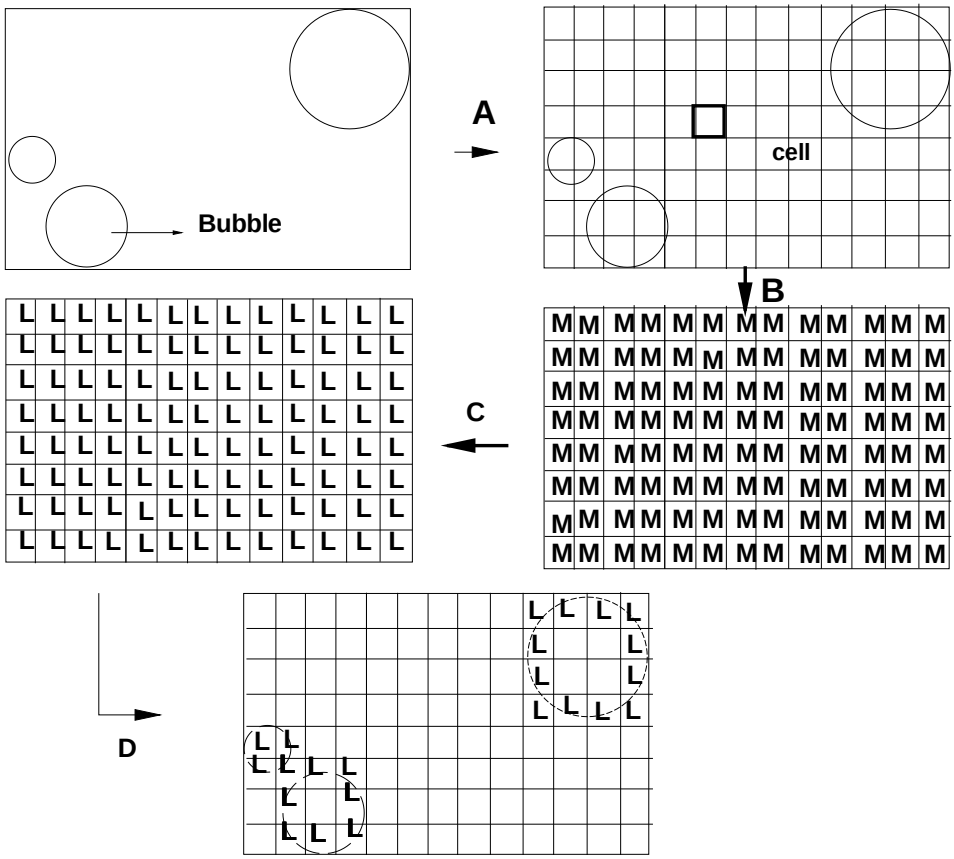

Figure 3.1: Two-dimensional pictorial representation of the FFTM algorithm. Step A: Division of problem domain into many smaller cells. Step B: Computation of multipole moments M for all cells. Step C: Evaluation of local expansion coefficients L at cell centers by discrete convolutions via FFT. Step D: For a given cell, compute the potentials contributed from distant and near sources.

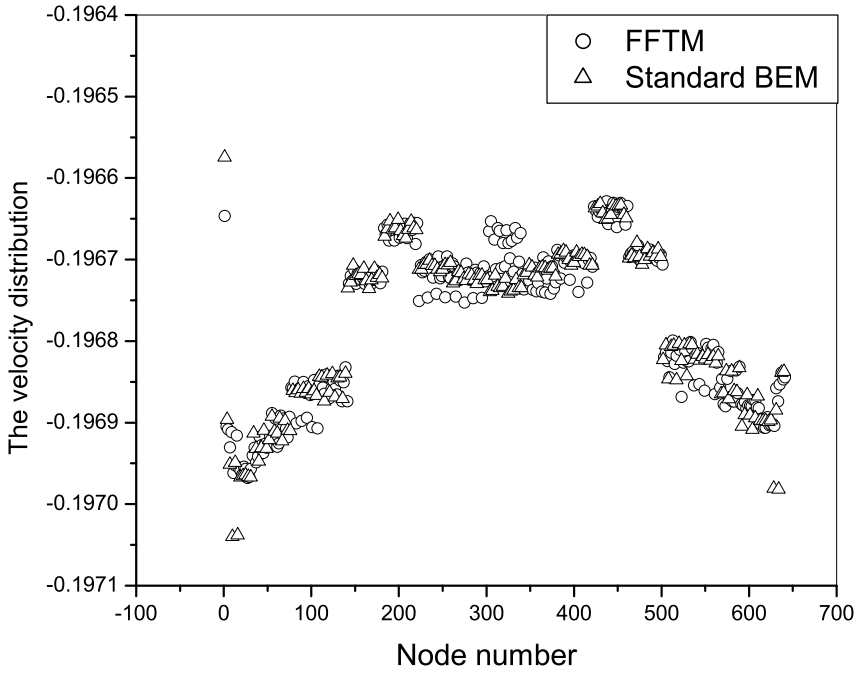

Figure 3.2: Distribution of dimensionless normal velocity on single bubble with 642 node and 1280 triangle elements at first time step. Circle and triangle represent solutions given by FFTM and standard BEM, respectively.

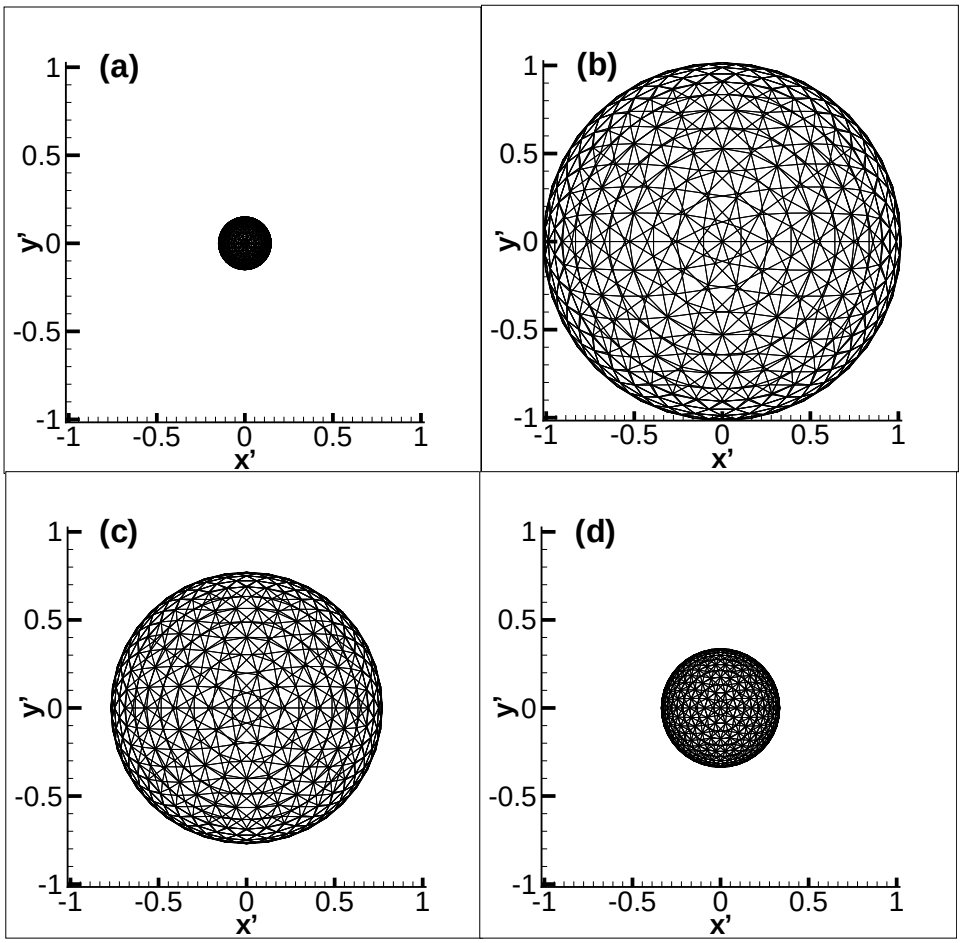

Figure 3.3: Evolution of Rayleigh bubble at mesh level 8, with 642 node and 1280 triangle elements. Solutions are given by FFTM. (a) is bubbles shapes during expansion phase at dimensionless time $t' = 0.06$. (b) is bubbles shapes at dimensionless time $t' = 1.00$. (c) and (d) are bubbles shapes during collapse phase $t' = 1.50$ and $t' = 1.80$, respectively.

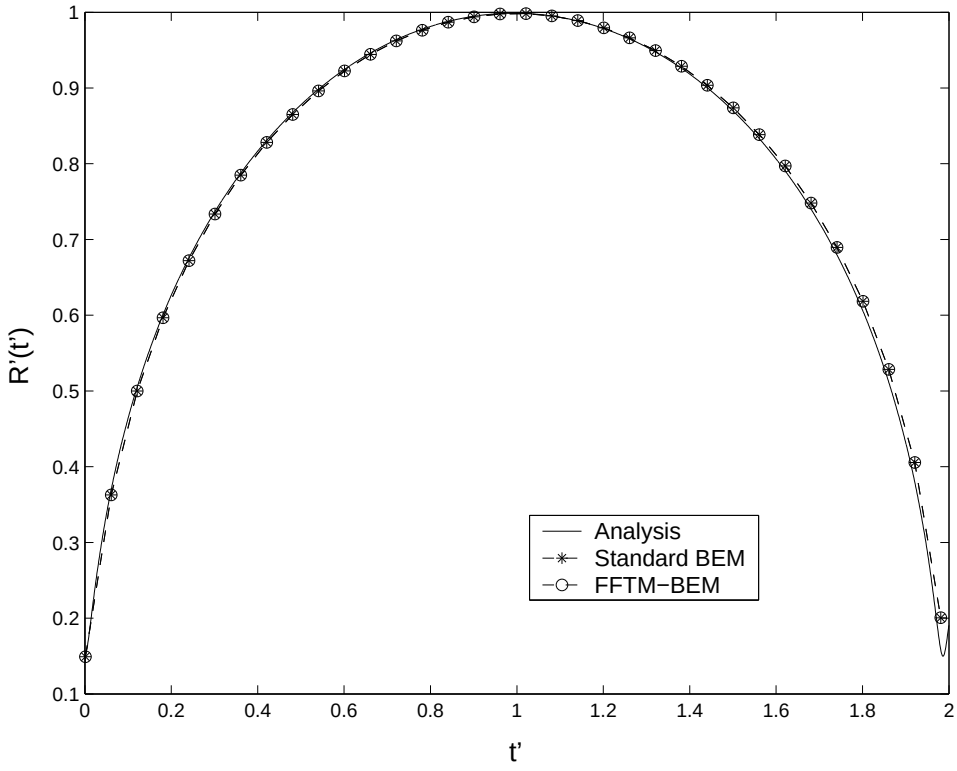

Figure 3.4: Comparison of analytic Rayleigh bubble radius R' with FFTM and standard BEM. For the numerics, the bubble was generated with 642 nodes and 1280 triangle elements.

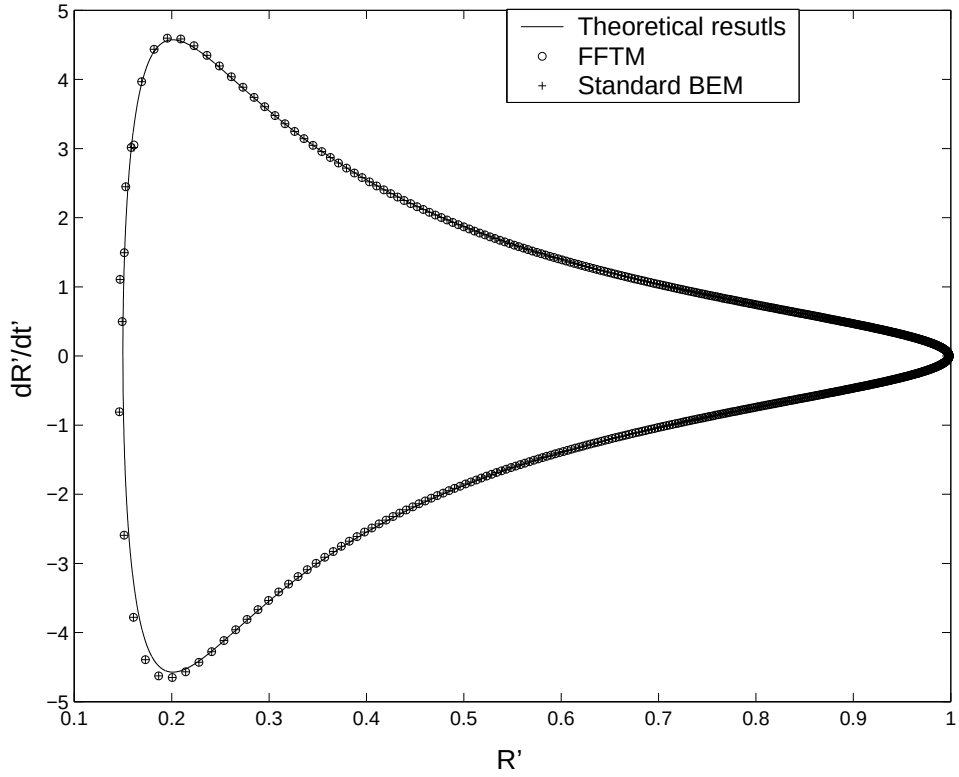

Figure 3.5: Comparison of dR'/dt' vs R' with FFTM and standard BEM. For the numerics, the bubble was generated with 642 nodes and 1280 triangle elements.

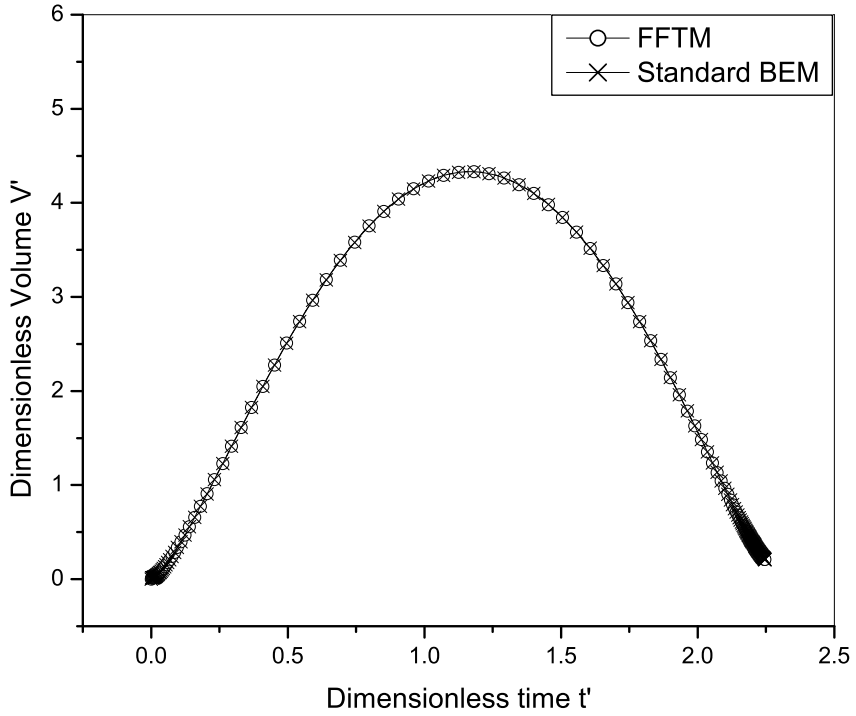

Figure 3.6: Comparison in dimensionless bubble volume produced by the FFTM and standard BEM.

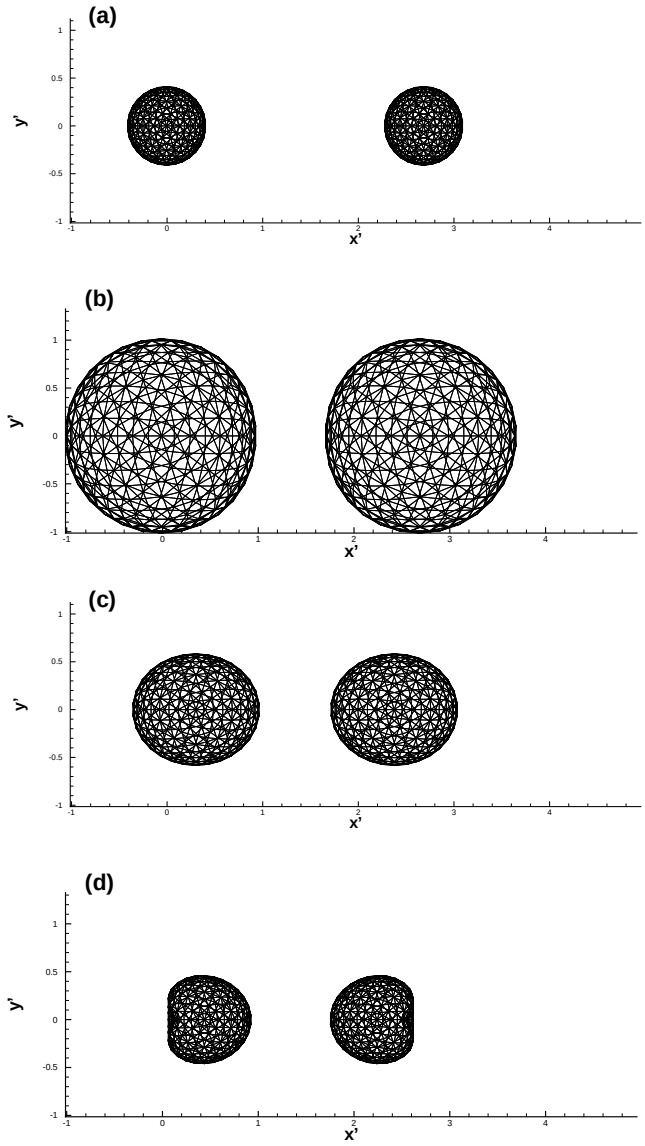

Figure 3.7: Evolution of two bubbles with initial dimensionless distance $d' = 2.6785$. Each bubble has 642 node and 1280 triangle elements on its mesh. Solutions are given by the FFTM. (a) and (b) are bubbles shapes during expansion phase at dimensionless time $t' = 0.080$ and $t' = 0.851$ respectively; (c) and (d) are bubbles shapes during collapse phase at dimensionless time $t' = 2.010$ and $t' = 2.449$ respectively.

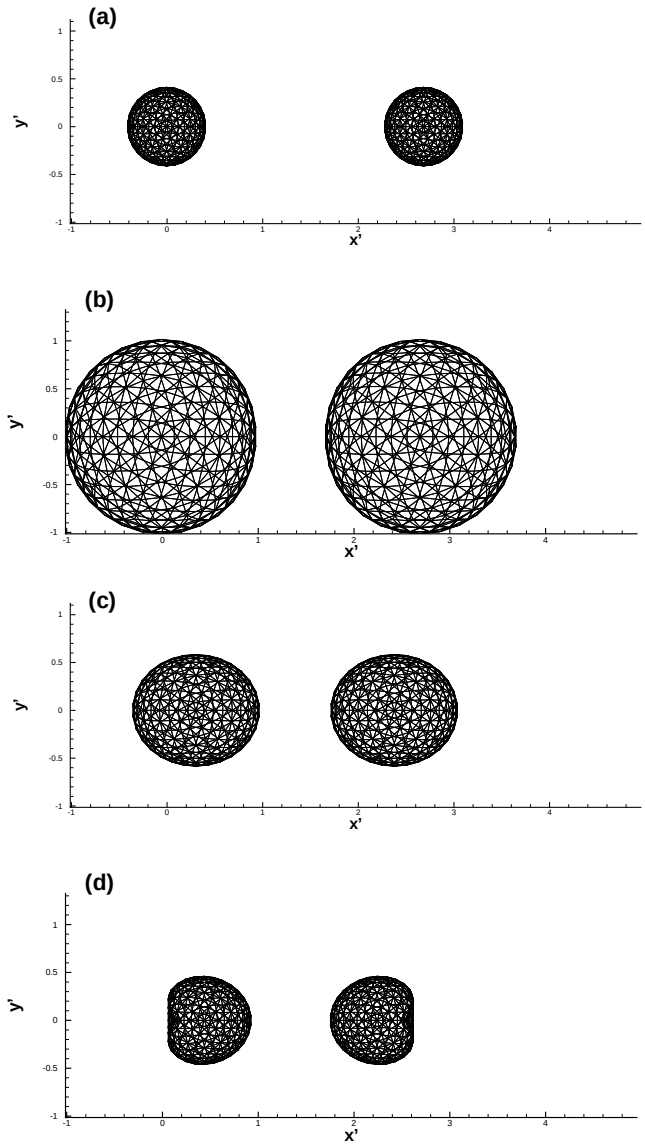

Figure 3.8: Evolution of two bubbles with initial dimensionless distance $d' = 2.6785$. Each bubble has 642 node and 1280 triangle elements on its mesh. Solutions are given by the standard BEM. (a) and (b) are bubbles shapes during the expansion phase at dimensionless time $t' = 0.080$ and $t' = 0.851$, respectively; (c) and (d) are bubbles shapes during collapse phase at dimensionless time $t' = 2.010$ and $t' = 2.449$, respectively.

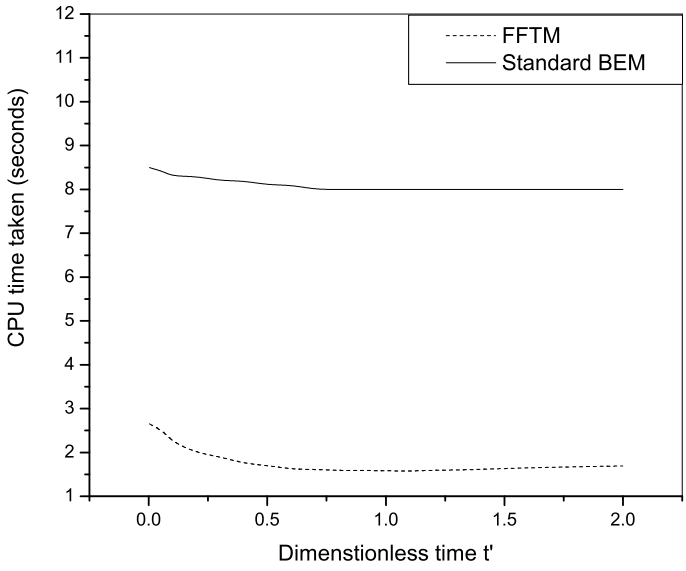

Figure 3.9: Comparison of CPU time for each time step taken by the FFTM method and standard BEM method during the bubbles evolution. Distance between the centers of two initial bubbles is $d' = 2.6785$.

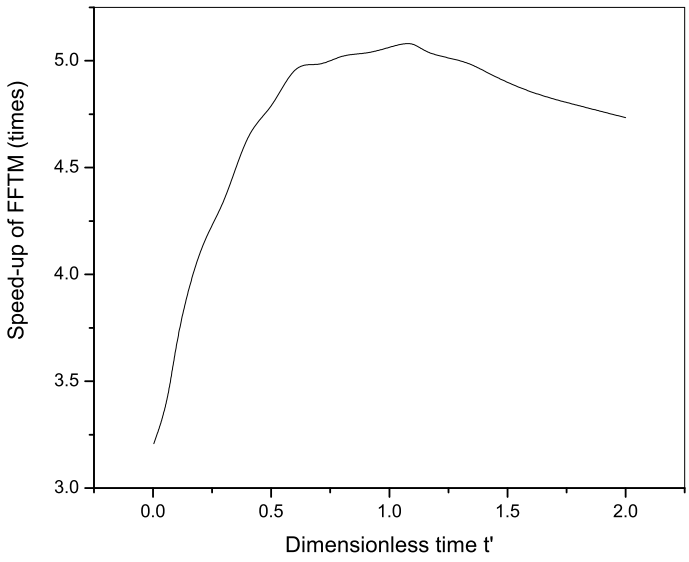

Figure 3.10: The speed-up factor of the FFTM method during bubbles simulation. Distance between the centers of two initial bubbles is $d' = 2.6785$.

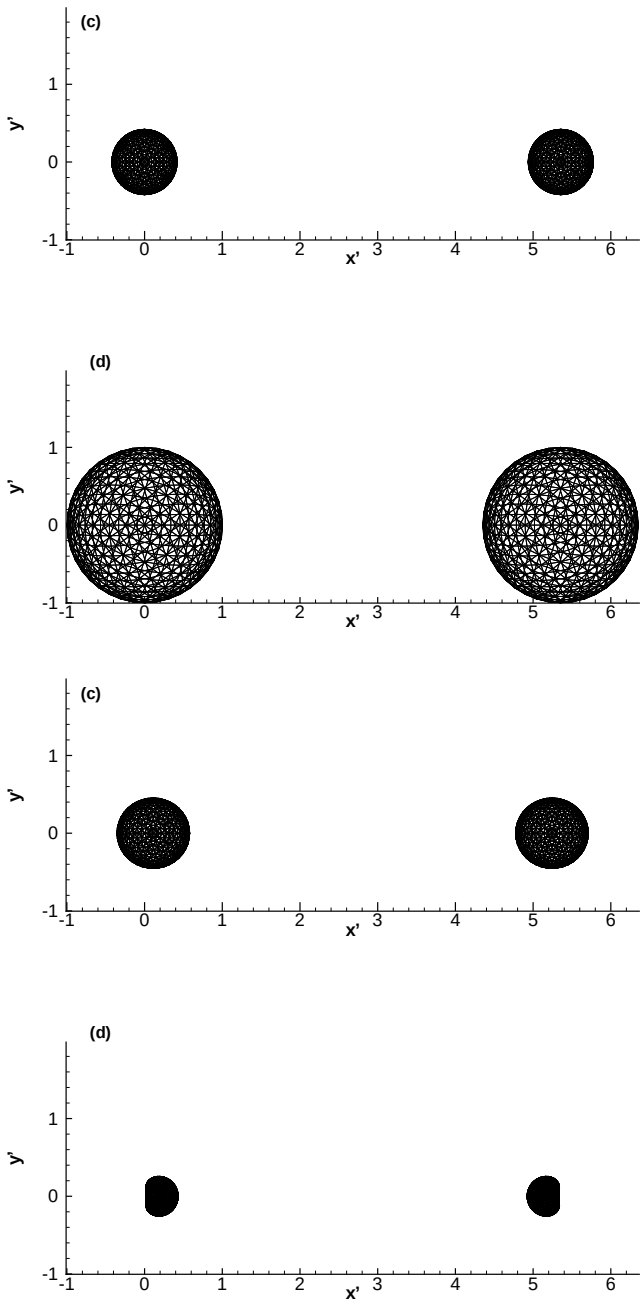

Figure 3.11: Evolution of two bubbles with initial centers distance of $d' = 5.357$. Each bubble has 642 node and 1280 triangle elements on its mesh. Solutions are given by the FFTM. (a) and (b) are bubbles shapes during expansion phase at dimensionless time $t' = 0.083$ and $t' = 0.881$ respectively. (c) and (d) are bubbles shapes during collapse phase at dimensionless time $t' = 2.109$ and $t' = 2.138$ respectively.

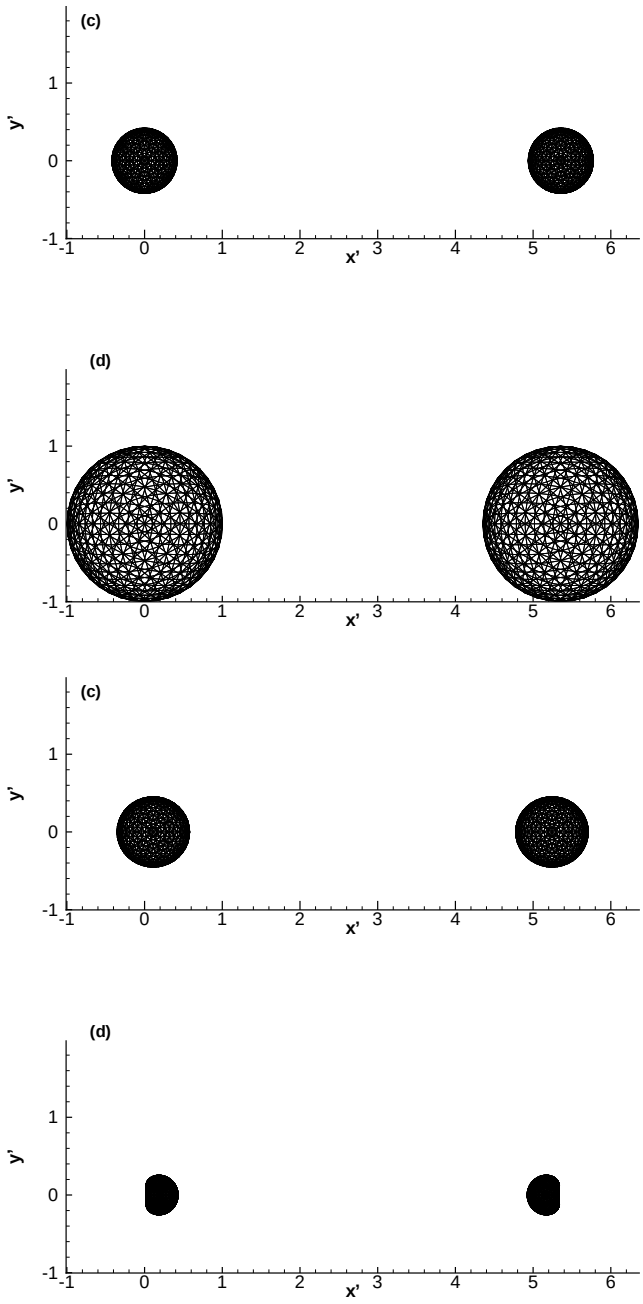

Figure 3.12: Evolution of two bubbles with initial centers distance of $d' = 5.357$. Each bubble has 642 node and 1280 triangle elements on its mesh. Solutions are given by the standard BEM. (a) and (b) are bubbles shapes during expansion phase at dimensionless time $t' = 0.083$ and $t' = 0.881$ respectively. (c) and (d) are bubbles shapes during collapse phase at dimensionless time $t' = 2.109$ and $t' = 2.138$ respectively.

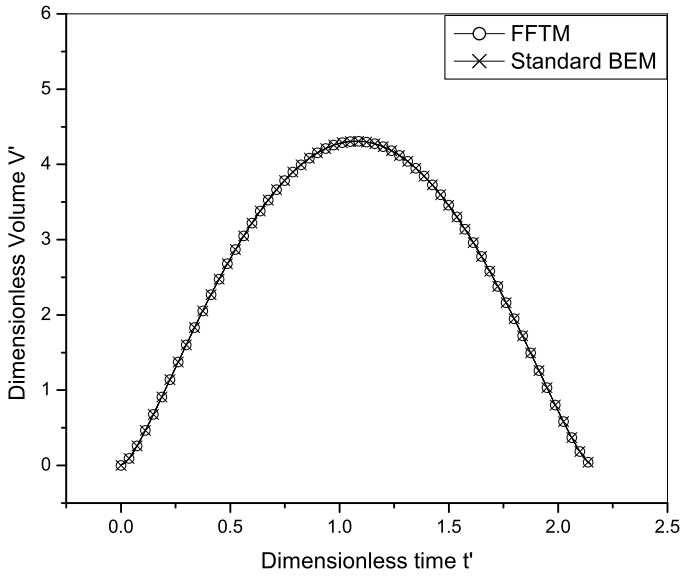

Figure 3.13: Comparison of dimensionless bubble volume produced by the FFTM and standard BEM.

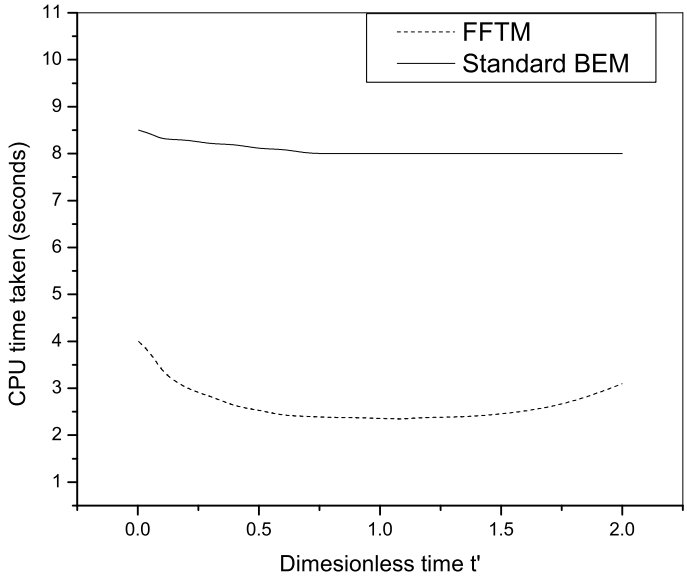

Figure 3.14: Comparison of CPU time for each time step taken by the FFTM method and the standard BEM method during the bubbles evolution. Distance between the centers of two initial bubbles is $d' = 5.357$.

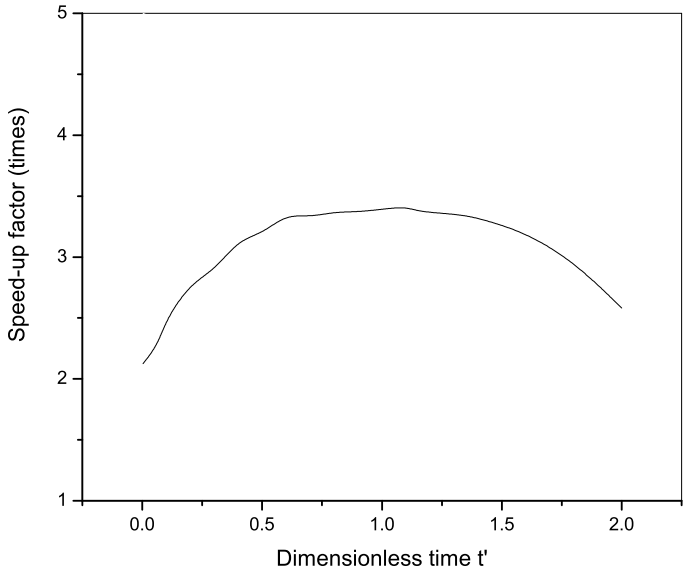

Figure 3.15: The speed-up factor of the FFTM method during bubbles simulation. Distance between the centers of two initial bubbles is $d' = 5.357$.

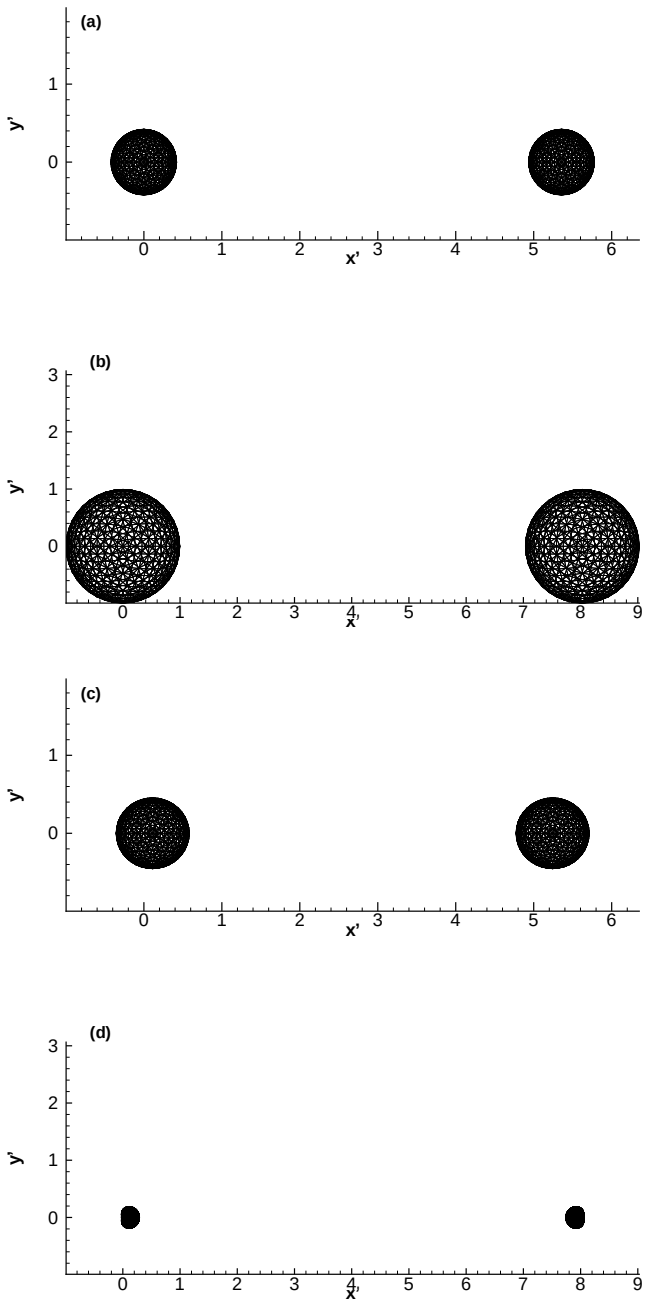

Figure 3.16: Evolution of two bubbles with initial centers distance of $d' = 7.95$. Each bubble has 642 node and 1280 triangle elements on its mesh. Solutions are given by the FFTM. (a) and (b) are bubbles shapes during expansion phase at dimensionless time $t' = 0.083$ and $t' = 0.887$ respectively. (c) and (d) are bubbles shapes during collapse phase at dimensionless time $t' = 2.109$ and $t' = 2.103$ respectively.

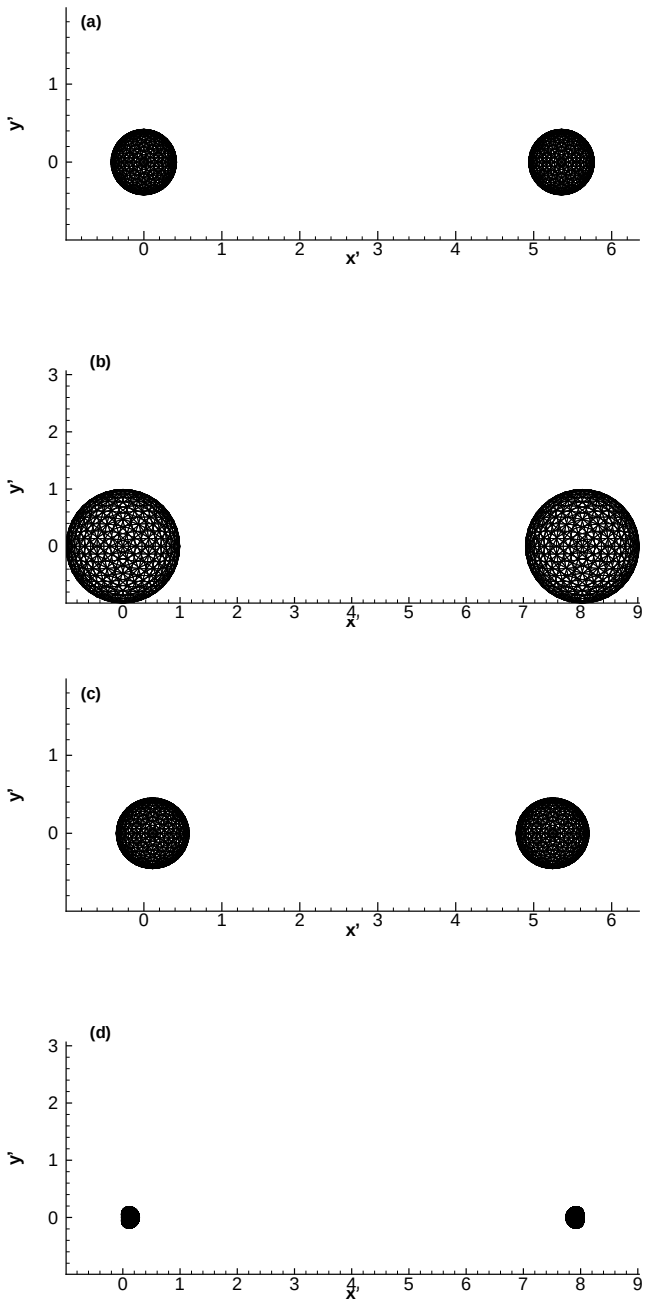

Figure 3.17: Evolution of two bubbles with initial centers distance of $d' = 7.95$. Each bubble has 642 node and 1280 triangle elements on its mesh. Solutions are given by the standard BEM. (a) and (b) are bubbles shapes during expansion phase at dimensionless time $t' = 0.083$ and $t' = 0.887$ respectively. (c) and (d) are bubbles shapes during collapse phase at dimensionless time $t' = 2.109$ and $t' = 2.103$ respectively.

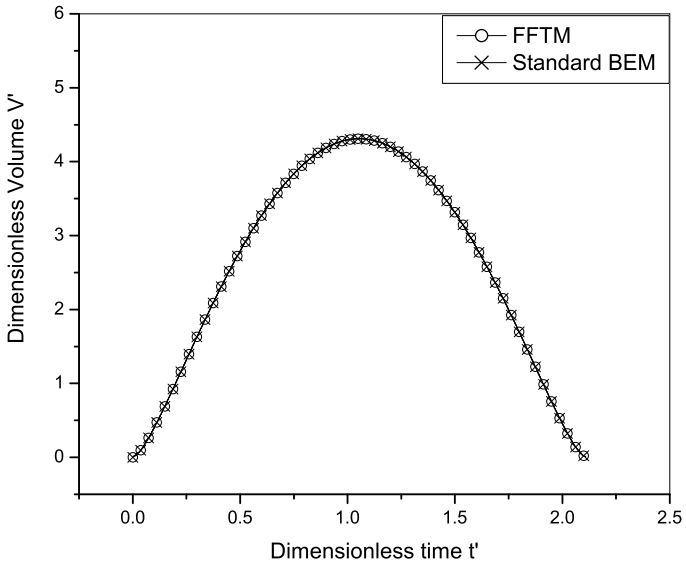

Figure 3.18: Comparison of dimensionless bubble volume produced by the FFTM and standard BEM.

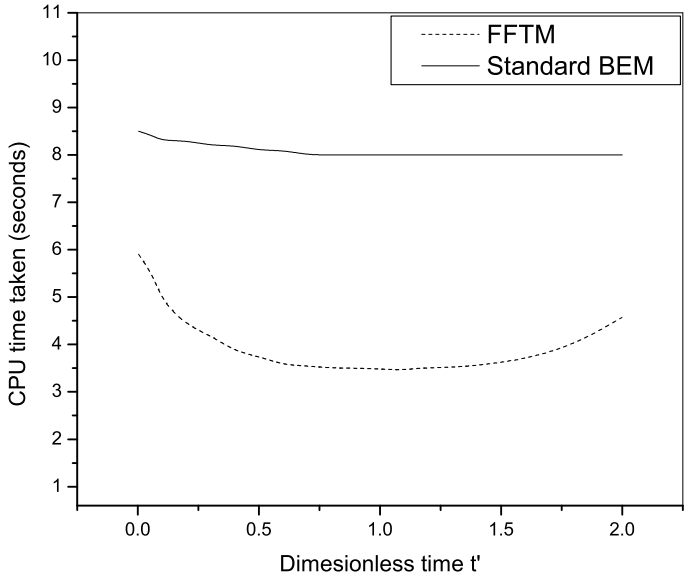

Figure 3.19: Comparison in time taken during bubbles evolution in each time step between the FFTM and standard BEM. Two bubbles with initial dimensionless distance $d' = 7.95$.

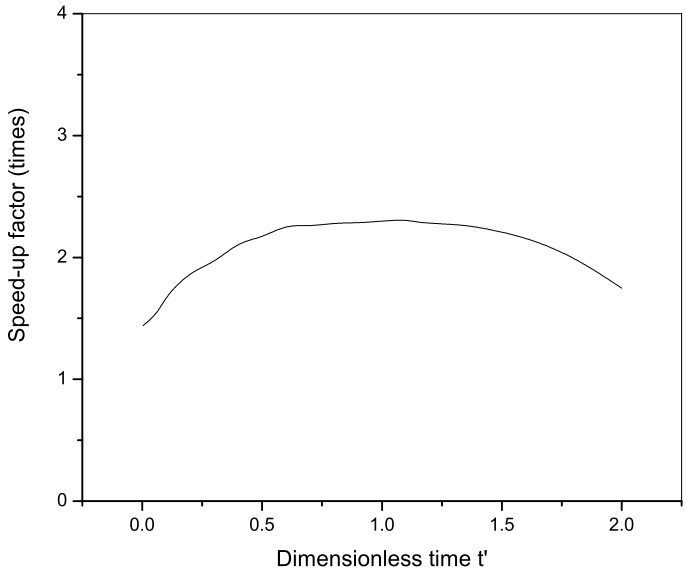

Figure 3.20: The speed-up factor of the FFTM method during bubbles simulation. Distance between the centers of two initial bubbles is $d' = 7.95$.

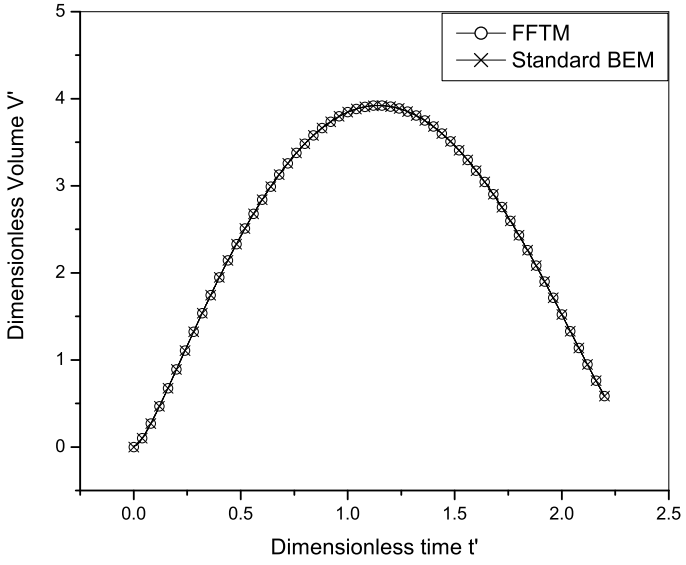

Figure 3.21: Comparison of dimensionless volume of the first bubble produced by the FFTM and standard BEM method.

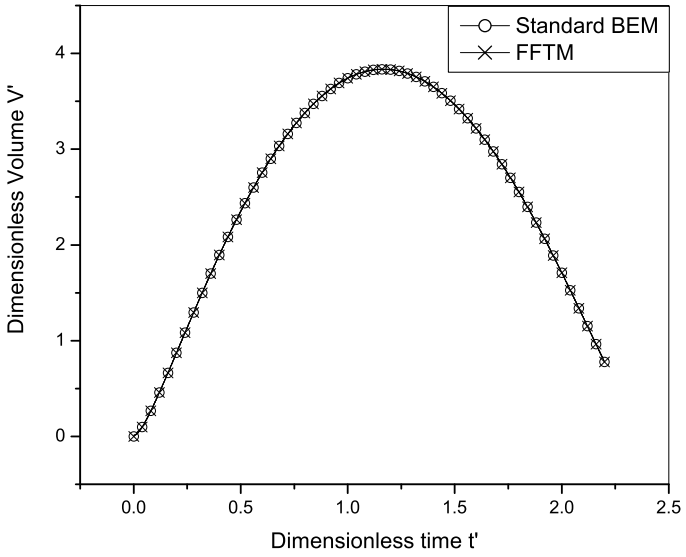

Figure 3.22: Comparison of dimensionless volume of the second bubble produced by the FFTM and standard BEM method.

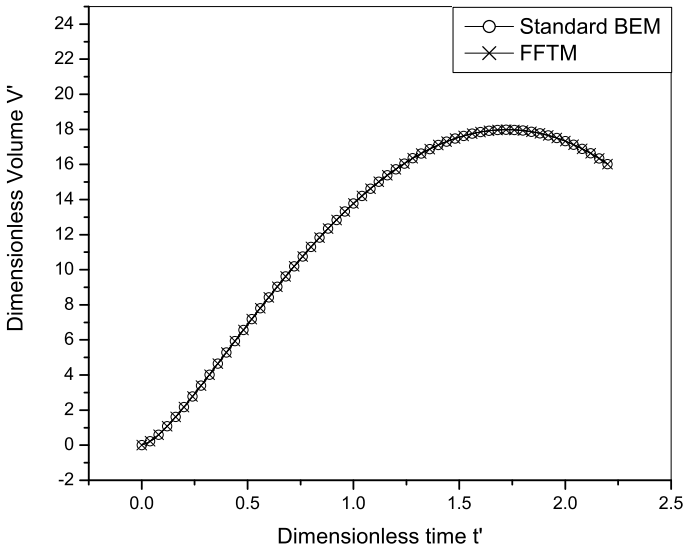

Figure 3.23: Comparison of dimensionless volume of the third bubble produced by the FFTM and standard BEM method.

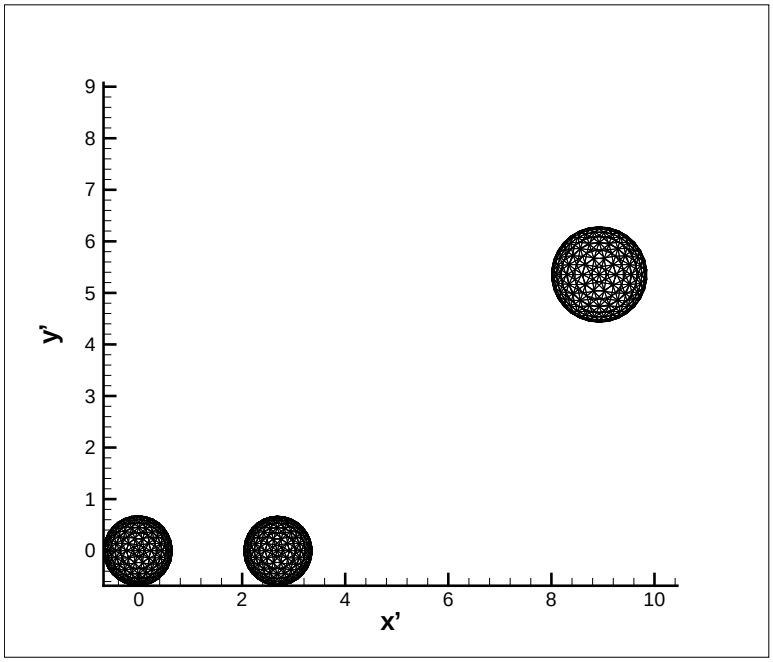

Figure 3.24: Bubbles shapes at dimensionless time t' = 0.271.

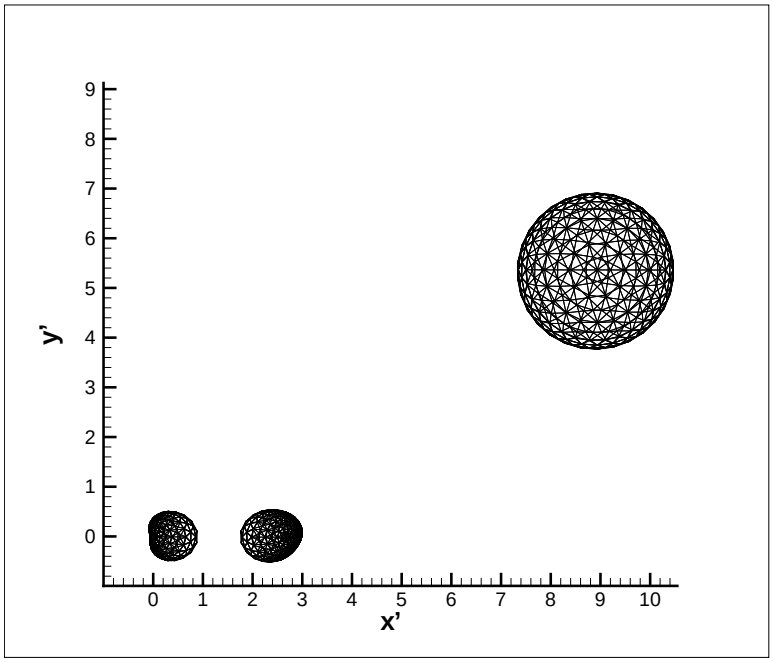

Figure 3.25: Bubbles shapes at dimensionless time t' = 2.221.

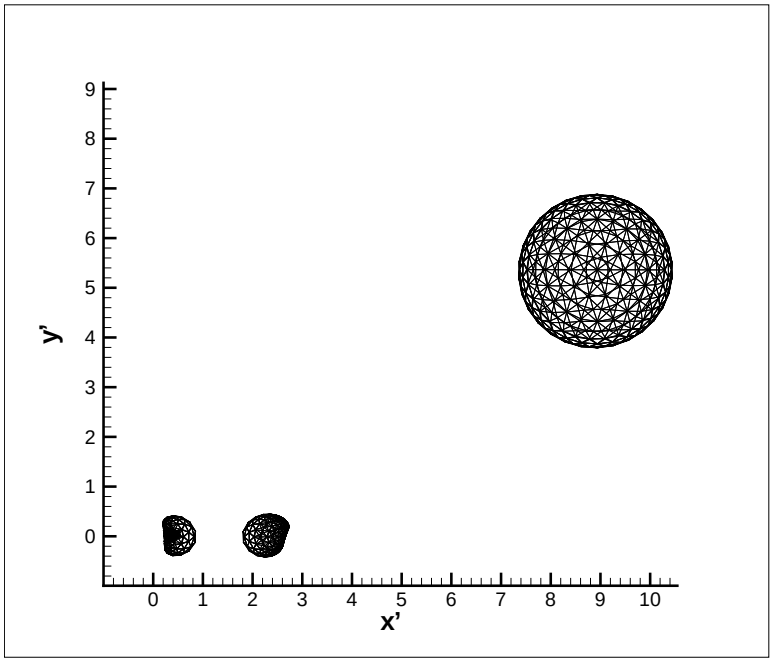

Figure 3.26: Bubbles shapes at dimensionless time t' = 2.3014.

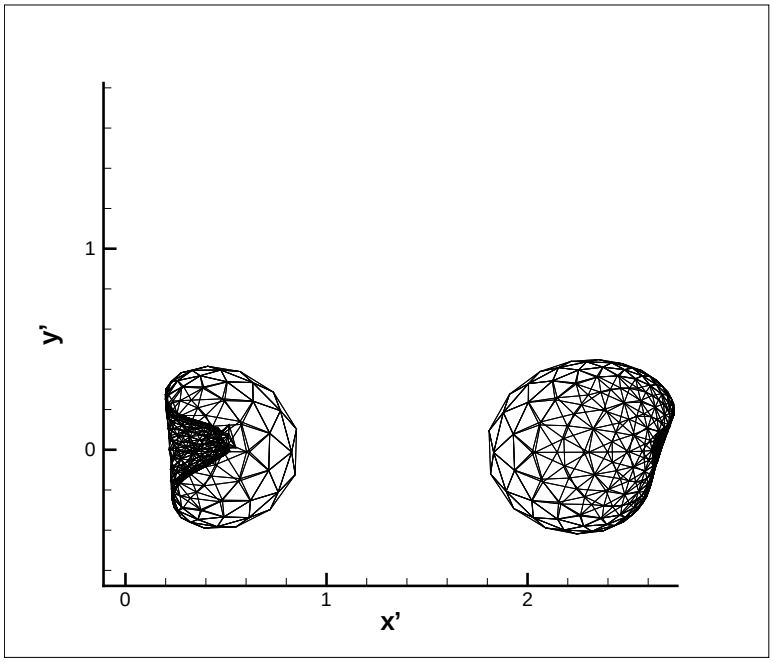

Figure 3.27: Zoom in of bubbles 1 and 2 at dimensionless time t' = 2.3014.

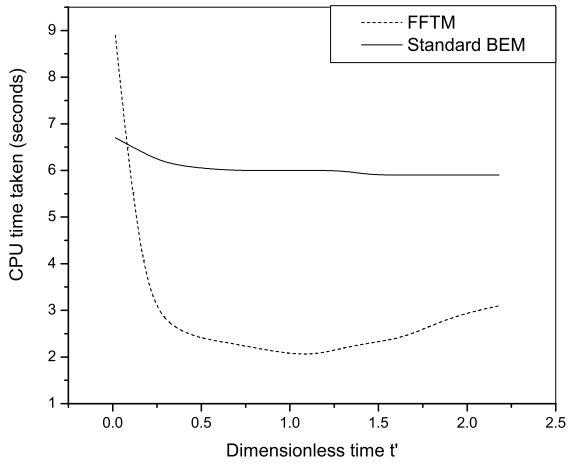

Figure 3.28: Comparison of time taken by using the FFTM and standard BEM in each step during evolution of three bubbles.

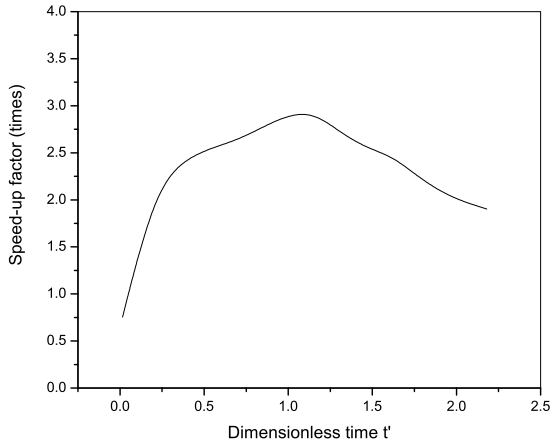

Figure 3.29: The speed-up factor of the FFTM method during evolution of three bubbles.

Chapter 4

New version of FFTM: FFTM Clustering

As shown in the previous chapter, the original version of FFTM does not work efficiently when the problem domain has much blank spaces or voids between the multiple bubbles. The efficiency of FFTM depends very much on the relative position of two bubbles and their radii. During inception time, each bubble is usually initiated as a small spheroids and the distance between two bubbles is usually much larger than subsequence states. For bubbles placed far away from each other, or bubbles radii are very small compared to the distance between the bubbles centers, the gain in speed up factor using the FFTM reduces.

The deterioration in the efficiency is largely due to the fact that FFTM has to perform calculations even for the empty cells. In other words, the computational efforts for evaluating the discrete convolutions have to be increased unnecessarily with many zero entries. In Fig 4.1(a), we shall use an example to describe the

problem domain and form multiple expansion coefficients of the FFTM algorithm. In usual practice, FFTM determine the smallest box with its edges parallel to the x, y, z axis. This box is the computational domain and is divided into many smaller cells as described in the FFTM algorithm encompassing all the multiple bubbles. Then it follows that multipole expansion coefficient and calculations are required for each cells. Where there are much blank spaces between the bubbles, we have many "zero cells" and an indiscriminate employment of the FFTM would be very wasteful.

On the other hand, FMM is noted to be hardly affected by this sparseness issue [29], since it ignores all the empty cells. Hence, this seems to suggest that FMM is a preferred fast solver for sparse problems over the FFT-based methods, including FFTM. But due to the accuracy requirement, where FMM needs significantly higher order of expansion than FFTM to achieve the same order of accuracy (see [41] for more detail discussions on this accuracy issue), the actual outcome may not be as desirable. To be more explicit, FMM [29] needs $p = 8$ to achieve 3-digits accuracy, whereas FFTM [41] needs only $p = 2$. An obvious impact is that the computational efforts for computing and storing the multipole and local expansion coefficients and also their related transformation matrices would be significantly more expensive for FMM than FFTM, since they scale like $O(p^2)$. From the above discussions, it seems that neither FMM nor FFTM are suitable for solving spatially sparse problems. To alleviate this sparseness issue, we propose the following FFTM Clustering method.

The main difference between FFTM Clustering and standalone FFTM is FFTM Clustering does the grouping of "near" bubbles together into one group and carries out all the calculation on this group domain. That is, FFTM Clustering will group

all the nearby bubbles together, and then the necessary calculations on the sub-domains containing bubbles. A brief pictoral comparison between the algorithms of standalone FFTM and FFTM Clustering is plotted in Fig. 4.1.

4.1 FFTM clustering algorithm

Basically, FFTM clustering includes the following steps:

I) Clustering the domain into "well-separated" groups (see Fig. 4.3-A). The formation of each groups depends on the distance between the bubbles and their radii.

First, we define a group. A group can contain one single bubble or multiple nearby bubbles. Each group is marked by a spheroid that contain all affected bubbles inside. The spheroid may not coincide with the bubble shape because bubbles deform and will not always keep to the spherical shape. When a group contains only single spherical bubble, the spheroid coincides with the spherical bubble.

Secondly, we describe the grouping process. Initially, each bubble is considered as a group surrounded by a spheroid. Assume that the radii of two spheroids surrounding group 1 and 2 are R_1 and R_2 with the centers at O_1 and O_2, respectively. The distance between their centers is D. If the distance between centers of the two spheroids is smaller than a critical distance, the two groups will be joined together and form a new group. The critical distance of forming a new group will be defined in the next part. The new group is represented as a spheroid with its new center O_{12} located at the midpoint of O_1 and O_2 and the radius $R_{12} = (R_1 + R_2 + D)/2$. Fig.

4.2 shows the grouping process of two bubbles centered at O_1 and O_2 with radii R_1 and R_2, respectively.

The relative distance of the new spheroid with other groups R_{12} is considered next during the grouping process. Hence, we will have separated groups after the clustering process.

II) Calculate the total contribution from the outside groups and the contribution within the group. To calculate the contribution of all source points in each group on a point within itself, the FFTM method is employed here.

To calculate the contribution of all sources outside the group, first we define sub-groups for all the groups, such that each sub-groups forms a hierarchical oct-tree. This is to facilitate the use of the various translation operators discussed in Section 3.1. The hierarchical oct-tree is formed in each cluster. A cluster is a parent cell, then each cell is divided into eight smaller child cells at the higher level. At the highest level, there are not sub-cells inside each cell. Now, the procedures of computing the multipole interactions among the sub-groups are identical to that used in FMM. First, the multipole expansion at the highest level is calculated, then the multipole expansions at the lower level are translated step by step until it reached the lowest level (as in Fig. 4.3-b) by using translation 3.13. Next step involves the conversions of multipole expansions to local expansions due to contribution of other cells outside the group via (3.14) and translations of the local expansion to local expansion coefficients at higher level using translation 3.15 (as in Fig. 4.3-c). In the figure, the larger $\mathbf{M}$ corresponds to the multipole moments at the higher levels of the oct-tree, which are used to compute the local expansion coefficients $\mathbf{L}$ at the

respective higher levels. The order of expansions used for each cell interactions can be different, which depends on the desired accuracy and the relative distance between the interacting cells. It is important to note that the number of multipole to local expansions conversion operations, which is the most computationally expensive of the FMM algorithm [28]-[29], is not an issue here. This is because most of the multipole interactions would be at the coarsest level (since the groups are well-separated). Hence, this part of the algorithm only scales like $O(N_{SG})$, where N_{SG} is the total number of sub-groups, which is significantly lesser than the number of multipole cells in the original discretization. Note also that the order of expansions used to compute the various sub-groups cells interactions are not fixed, but rather vary according to the desirable accuracy and also their relative distances.

Finally, the local expansion coefficients L on the smallest cells are finally derived by summing the contributions from the FFTM computation of its self-group, and also the local expansions from the higher level cell interactions due to multipole moments from the other groups (Fig. 4.3-c).

Remark: FFTM Clustering is a generalized method which encompasses the original FFTM method. When the problem at hand has only one group, the FFTM Clustering is simply the original FFTM. This is because the contribution of all sources within one group is itself.

Critical distance

The efficiency of the FFTM Clustering algorithm depends on the critical distance of the grouping process. Consider two spherical bubbles of radius $R_1 = R_2 = r = 1$, with the distance between their centers is D. The velocity potential on each node

on the two bubble are set at 1. Each bubble contains 642 nodes on the surface. The centers of the two bubbles will be varied at different tests from $D = 2.5r$ to $D = 6.0r$. The efficiency of FFTM Clustering in terms of accuracy and efficiency will be compared between two examples: grouping and non-grouping. For the non-grouping examples, the FFTM Clustering performs in the two individual groups, each group represents one bubble. For the grouping examples, the is only one group containing the two bubbles on the problem domain, the FFTM Clustering will then become the standalone FFTM. The previous chapter has shown that the standalone FFTM is an accurate method. The results produced by the FFTM Clustering in the two groups will be compared with the results from standalone FFTM to measure the accuracy the performance of the FFTM Clustering in the non-grouping example. The average error incurred for the FFTM Clustering solver in the non-grouping example is measured in the L_2 norm as:

$$\text{Average Error} = \left(\frac{\sum_{i=1}^{N} |\frac{\partial \Phi'}{\partial \mathbf{n}}^{i}_{non-grouping} - \frac{\partial \Phi'}{\partial \mathbf{n}}^{i}_{grouping}|^2}{\sum_{i=1}^{N} |\frac{\partial \Phi'}{\partial \mathbf{n}}^{i}_{grouping}|^2} \right)^{\frac{1}{2}} \times 100\% \qquad (4.1)$$

where $\frac{\partial \Phi'}{\partial \mathbf{n}}^{i}_{non-grouping}$ and $\frac{\partial \Phi'}{\partial \mathbf{n}}^{i}_{grouping}$ are the normal velocity value of node i produced by FFTM Clustering solver in non-grouping and grouping example, respectively. N is the total number of nodes.

The tests were conducted on a single IBM p690 Regatta 1.3 GHz CPU speed. The average error defined in 4.1 will be used to measure the FFTM Clustering solver in the non-grouping example. Table 4.1 shows the comparison between the two methods in term of in accuracy and CPU time taken.

As seen from Table 4.1, while the CPU time taken in the grouping example increases, the CPU time taken by the non-grouping examples reduces and converges to a constant value as the bubbles are placed further apart. The average error also reduces when the bubbles are more further apart. From that table, when $D = 4r$ the FFTM Clustering performs the most efficiently in terms of accuracy and CPU time taken. From this, we suggest that the critical grouping distance between centers of two bubbles is about optimal at $D = 2(R_1 + R_2)$, where R_1 and R_2 are radius of two bubbles.

4.2 Results and discussion

In this section, we test the efficiency and accuracy of FFTM Clustering on the same examples as in Section 3.4.2. The accuracy of the FFTM Clustering method is measured by the average error of the normal velocity between results produced by the FFTM Clustering and standard BEM which is defined as:

$$\text{Aver. Error} = \left(\frac{\sum_{i=1}^{N} |\frac{\partial \Phi'}{\partial \mathbf{n}}_{FFTM-Clt}{}^{i} - \frac{\partial \Phi'}{\partial \mathbf{n}}_{Stand.BEM}{}^{i}|^2}{\sum_{i=1}^{N} |\frac{\partial \Phi'}{\partial \mathbf{n}}_{Stand.BEM}{}^{i}|^2} \right)^{\frac{1}{2}} \times 100\%, \tag{4.2}$$

where $\frac{\partial \Phi'}{\partial \mathbf{n}}_{FFTM-Clt}{}^{i}$ and $\frac{\partial \Phi'}{\partial \mathbf{n}}_{std.BEM}{}^{i}$ are the normal velocity value of node i produced by FFTM Clustering solver and standard BEM respectively. N is total number of nodes.

The efficiency of the FFTM Clustering at each time step is evaluated by the speed-up factor which is defined as:

$$\text{speed-up} = \frac{T_{std.BEM}}{T_{FFTM-Clt}}, \tag{4.3}$$

here, $T_{std.BEM}$ and $T_{FFTM-Clt}$ are the CPU time taken by the standard BEM and FFTM Clustering at each time step, respectively.

4.2.1 Performance of the FFTM Clustering on two bubbles

We consider two explosion bubbles far away from any boundaries and no gravitational effect. Each charge is given as 500 kg of TNT placed at a depth of 100 m away from the free water surface, and strength parameter $\varepsilon = 97.52$ and ratio of specific heat $\lambda = 1.25$. Center of two bubble are initially placed on the x-axis at a distance $d = 15$ m equivalent to dimensionless distance $d' = 2.6785$ (same example as described in Section 3.4.2). Each bubble is generate with 642 nodes and 1280 triangle elements on its surface. We will compare the normal velocity field $\partial\Phi/\partial n$ between the FFTM Clustering and standard BEM at a typical time instant. Both the methods were carried out with 2200 time steps and the dimensionless time-step size is taken to be constant at $\Delta t' = 0.001$. This time-step size is small enough to ensure stability of the numerical scheme.

Table 4.2 shows the numerical error of normal velocity field between the FFTM Clustering method and standard BEM method. Compared to Table 3.2, the average error is almost same. The average error incurred is limited to less than 1%, while most of the time the error is less than 0.60%.

The evolution of the bubbles shapes from the FFTM Clustering calculation is

shown in Figure 4.4. The method gives almost the same bubbles shapes compared to the the former methods. Next, the bubble volume is compared to test further the accuracy of the FFTM Clustering method. Figure 4.5 shows the dimensionless volume of one bubble produced by the FFTM, FFTM Clustering and standard BEM method. There are not much differences between the three methods. The FFTM Clustering produces the dimensionless maximum 4.3313 at dimensionless time $t' = 1.153$, it makes only 0.012% error, compare with standard BEM.

The test were carried out on the single IBM p690 Regatta 1.3 GHz CPU speed. In this examples, the FFTM Clustering took 3748 seconds of CPU time to perform 2200 time-steps. The FFTM Clustering performed faster than standard BEM by about 4.8 times.

Fig. 4.6 shows the time taken for one-time step computation of the two methods during bubble evolution. Fig. 4.7 shows the speed-up factor of the FFTM method during bubble simulation. Bubbles are initiated with a small spheroid of the initial radius $R_0 = 0.149$. The distance between two bubble compared to the their radius is large enough to make two separated groups, and the CPU time taken by the FFTM Clustering in each time-step is about 1.67 seconds, while the standalone FFTM one changes rapidly. As the bubbles evolve, their radii increase and at dimensionless time $t' = 0.257$, their radius reaches $r' = 0.674$, while the separation distance between the centers is $d' = 2.67$. From this time instant onward, the bubbles are sufficiently close to form one group in the whole domain. Then FFTM Clustering automatically turn to standalone FFTM to solve the problem.

Next, consider two bubbles with the same flow conditions, but the separation dis-

tance between centers of initial bubbles further apart: at the dimensionless distance $d' = 5.357$ (see also Section 3.4.2). Table 4.3 show the relative error between the FFTM Clustering and standard BEM method. Table 3.3 shows the numerical error of normal velocity field between the FFTM Clustering method and standard BEM method. The average error incurred is limited to less than 1%, while most of the time the error is less than 0.50%.

The evolution of the bubbles shapes from the FFTM Clustering calculation is shown in Figure 4.8. The method gives almost the same bubbles shapes compared to the the former methods. Next, the bubble volume is compared to test further the accuracy of the FFTM Clustering method. Figure 4.9 shows the dimensionless volume of one bubble produced by the FFTM, FFTM Clustering and standard BEM method. There are not much differences between the three methods. Similar to the previous two methods, the FFTM Clustering method produces the maximum dimensionless volume V' at 4.3083 at dimensionless time $t' = 1.077$.

The tests were conducted on a single IBM p690 Regatta 1.3 GHz CPU speed. While the standard BEM took about 17843 seconds of CPU time to run 2200 steps, the FFTM took 5916 seconds, FFTM Clustering took 3674 seconds. In this example, the FFTM Clustering performed faster than the standard BEM by about 4.86 times. Comparison of the CPU time taken by FFTM Clustering, FFTM and standard BEM for each time step of calculation is shown in Figure 4.10. While the CPU time taken by the FFTM varied, the FFTM Clustering did not change much in each time-step during the simulation. Fig. 4.11 shows the comparison of the speed-up factor of the two methods FFTM and FFTM Clustering. It is shown in Fig. 4.11 that the FFTM

Clustering method ran faster than the standard BEM by above 5 times in each time step, while the FFTM ran faster than the standard BEM by only 3 times.

Further comparison is made for two bubbles in which the centers are separated at the dimensionless distance on the x-axis with $d' = 7.95$ (see also Section 3.4.2). Table 4.4 shows the relative error between the FFTM Clustering and standard BEM methods. The average error incurred is limited to less than 1%, while most of the time the error is less than 0.60%.

The evolution of the bubbles shapes from the FFTM Clustering calculation is shown in Figure 4.12. The method gives almost the same bubbles shapes compared to the former methods. Next, the bubble volume is compared to test further the accuracy of the FFTM Clustering method. Figure 4.13 shows the dimensionless volume of one bubble produced by the FFTM, FFTM Clustering and standard BEM method. There are not much differences between the three methods. Similar to the previous two methods, the FFTM Clustering method produces the maximum dimensionless volume V' at 4.310 at dimensionless time $t' = 1.050$.

Comparison of CPU taken by the FFTM Clustering, FFTM and standard BEM during bubble simulation for each time step calculation is shown in Fig. 4.14. The tests were conducted on a single IBM p690 Regatta 1.3 GHz CPU speed. While the standard BEM took about 17843 seconds of CPU time to run 2200 steps, FFTM took 8874 seconds, FFTM Clustering took 3674 seconds. In this example, the CPU time taken by FFTM Clustering is constant in each time step. In this example, the FFTM Clustering performed faster than the standard BEM by about 4.86 times and the FFTM performed faster than the standard BEM only by about 2 times. Comparison

of the CPU time taken by FFTM Clustering, FFTM and standard BEM for each time step of calculation is shown in Figure 4.14. While the CPU time taken by the FFTM varied, the FFTM Clustering did not change much in each time-step during the simulation. Fig. 4.15 shows the comparison of the speed-up factor of the two methods FFTM and FFTM Clustering. It is shown in Fig. 4.15 that the FFTM Clustering method ran faster than the standard BEM by around 5 times in each time step, while the FFTM ran faster than the standard BEM by around only 2 times.

4.2.2 Performance of the FFTM Clustering on three bubbles

In this section, we employ the FFTM Clustering for modelling three explosion bubbles far away from any boundaries and no gravitational effect. Centers of three bubbles are initially placed on the z-plane and located at the corners of an equilateral triangle of dimensionless length $a' = 3.57$ (see also Section 3.4.2). Each bubble starts to expand from initial bubble radius $R'_0 = 0.1499$. Other parameters of all the bubbles are strength parameter $\varepsilon = 97.52$ and ratio of specific heat $\lambda = 1.25$. Each bubble is generate with 362 nodes and 720 triangle elements on its surface. FFTM Clustering method was carried out with 2200 time steps and the dimensionless time-step size is taken to be constant at $\Delta t' = 0.001$. This example is as same as the example carried out for the FFTM test (see Section 3.4.2).

A quantitative comparison of the FFTM Clustering and standard BEM in normal velocity field is provided in Table 4.5. The average error incurred is limited to less than 1.02%, while most of the time the error is less than 0.60%. For bubbles shapes

comparison, there are not much difference between the results produced by the FFTM Clustering and the former methods.

The tests were conducted on a single IBM p690 Regatta 1.3 GHz CPU speed. While the standard BEM took about 13231 seconds of CPU time to run 2200 steps, the FFTM took 6547 seconds, the FFTM Clustering took about 3388 seconds. In this example, the FFTM Clustering performed faster than the standard BEM by about 4.0 times, while the FFTM performed faster than the standard BEM by only 2.0 times. Comparison of the CPU time taken by FFTM Clustering, FFTM and standard BEM for each time step of calculation is shown in Fig. 4.16. While the CPU time taken by the FFTM varied, the FFTM Clustering did not change much in each time-step during bubbles simulation. Fig. 4.17 shows the comparison of the speed-up factor of the two methods FFTM and FFTM Clustering. It is shown in Fig. 4.17 that the FFTM Clustering method ran faster than the standard BEM by around 4 times in each time step, while the speed-up factor of the FFTM varied from 0.5 to 3 times.

4.2.3 The efficiency of the FFTM Clustering on multiple bubbles

The efficiency of the FFTM Clustering comes from the use of the multipole expansions and local expansions on each group. The standard BEM solves the boundary integral equation by generating a dense matrix system which requires $O(N^2)$ operations to solve with iterative methods such as Gauss Iterative or Generalized Minimal

Residual (GMRES). While the FFTM Clustering requires only $O(N\log N)$ operations.

In the previous sections, the FFTM Clustering has shown its superb performance on the test cases for two or three bubbles. It performs better by around three times to five times faster than the standard BEM in those tests. In this section, further tests for cases with more bubbles are carried out. The FFTM Clustering and the Standard BEM are used in each test on the single IBM p690 Regatta 1.3 GHz CPU speed. The FFTM has not been implemented in this test because the efficiency of the FFTM depends on the distribution of the bubbles and defers very much during bubbles simulation. We consider five examples for two, four, six, eight and twelve bubbles, respectively. The locations of bubbles in the two-bubbles example in dimensionless form are: (0,0,0) and (0,0,5). The locations of bubbles in the four-bubbles example in dimensionless form are: (0,0,0), (0,0,5), (0,5,5), (0,5,0) and (5,0,0). The locations of bubbles in the six-bubbles example in dimensionless form are: (0,0,0), (0,0,5), (0,5,5), (0,5,0), (5,0,0) and (5,0,5). The locations of bubbles in the eight-bubbles example in dimensionless form are: (0,0,0), (0,0,5), (0,5,5), (0,5,0), (5,0,0), (5,0,5), (5,5,5) and (5,5,0). The locations of bubbles in the twelve-bubbles example in dimensionless form are: (0,0,0), (0,0,5), (0,5,5), (0,5,0), (5,0,0), (5,0,5), (5,5,5) , (5,5,0), (10,0,0), (10,0,5), (10,5,5) and (10,5,0). Each bubble was initiated from a charge of 500 kg TNT placed at a depth of 100 m away from the free water surface, and exploded under water at a depth of $H = 100$ m. Other parameter for bubbles are strength parameter $\varepsilon = 97.52$ and ratio of specific heat $\lambda = 1.25$. The maximum bubble radius is 5.6 m corresponding to the dimensionless value of 1.0. All the bubbles

were exploded or initiated from a spheroid with the initial radius $R_0' = 0.149$. In all the tests, each bubble is generated with 642 nodes and 1280 triangle elements on its surface. The CPU time taken by the standard BEM method in the large scale problem (for example: eight bubbles, twelve bubbles) does not permit to undertake 2200 time step as in the previous sections. Therefore, we conducted only 50 time steps in these said tests. It is noted that the time taken by the FFTM Clustering method and the standard BEM does not differ much in each time step during the simulation. The efficiency of the FFTM Clustering is measured in term of speed-up factor which is defined as:

$$\text{speed-up} = \frac{T_{std.BEM}}{T_{FFTM-Clt}}. \tag{4.4}$$

Here $T_{std.BEM}$ and $T_{FFTM-Clt}$ are the average CPU time in one time step taken by the standard BEM and FFTM Clustering, respectively.

Table 4.6 and Fig. 4.18 show the average CPU time taken in one time step of the FFTM Clustering and standard BEM. In the twelve-bubbles example, the FFTM Clustering needed 8.1 seconds of CPU time, while the standard BEM needed 547.1 seconds. To capture the evolution of the bubbles accurately, one must choose small time-step size. With the number time steps taken to be, say 2000, the FFTM Clustering needed only 4.5 hours while the standard BEM would need 254 hours corresponding to 10.5 days. As seen in Fig. 4.19, the speed-up factor increases significantly when the number of bubbles or problem size increases.

It is shown that the FFTM Clustering performs efficiently on the problem which

has large number of nodes or bubbles. The FFTM Clustering method is able to simulate the multiple bubbles with large scale problem size which is difficult to solve using the standard BEM.

Table 4.1: Comparison the performance of FFTM Clustering in term of accuracy and CPU time taken on the two examples: Grouping and Non-grouping.

D	Average Error	CPU time Non-grouping (seconds)	CPU time Grouping (seconds)
2.5 r	0.35	1.54	1.26
3.0 r	0.22	1.37	1.44
4.0 r	0.05	1.16	1.67
5.0 r	0.04	1.16	1.88
6.0 r	0.05	1.16	2.89

Table 4.2: Average error produced by the FFTM Clustering method of the normal velocities on two bubbles with separate distance between centers $d' = 2.6785$.

Dimensionless time	**Average Error (%)**
0.015	0.12
0.045	0.12
0.075	0.13
0.180	0.23
0.285	0.22
0.315	0.22
0.375	0.24
0.465	0.35
0.585	0.36
0.615	0.34
0.765	0.40
0.855	0.58
0.915	0.79
1.015	0.89
1.290	0.99
1.365	0.73
1.440	0.54
1515	0.42
1.665	0.45
1755	0.51
1845	0.45
1.965	0.60

Table 4.3: Average error produced by the FFTM Clustering method of the normal velocities on two bubbles with separate distance between centers $d' = 5.357$.

Dimensionless time	**Average Error (%)**
0.015	0.11
0.045	0.12
0.075	0.15
0.180	0.18
0.285	0.10
0.315	0.25
0.375	0.26
0.465	0.21
0.585	0.19
0.615	0.19
0.765	0.42
0.855	0.63
0.915	0.84
1.015	0.89
1.290	0.45
1.365	0.42
1.440	0.40
1515	0.36
1.665	0.26
1755	0.22
1845	0.29
1.965	0.27

Table 4.4: Average error produced by the FFTM Clustering method of the normal velocities on two bubbles with separate distance between centers $d' = 7.95$.

Dimensionless time	**Average Error (%)**
0.015	0.11
0.045	0.13
0.075	0.17
0.180	0.18
0.285	0.10
0.315	0.13
0.375	0.21
0.465	0.26
0.585	0.19
0.615	0.19
0.765	0.33
0.855	0.62
0.915	0.94
1.025	0.89
1.290	0.34
1.365	0.33
1.440	0.34
1515	0.33
1.665	0.23
1755	0.20
1845	0.30
1.965	0.33

Table 4.5: Average error of the normal velocities on three bubbles produced by the FFTM Clustering method.

Dimensionless time	**Average Error (%)**
0.015	0.13
0.045	0.29
0.075	0.14
0.180	0.16
0.285	0.20
0.315	0.21
0.375	0.21
0.465	0.25
0.585	0.29
0.615	0.31
0.765	0.33
0.855	0.45
0.915	0.50
1.025	0.65
1.290	1.02
1.365	0.77
1.440	0.62
1515	0.49
1.665	0.31
1755	0.30
1845	0.34
1.965	0.33

Table 4.6: The average CPU time taken in each time step by the FFTM Clustering and the standard BEM.

Number of bubbles	$T_{FFTM-Clt}$ (secs)	$T_{std.BEM}$ (secs)	speed-up factor (times)
2	1.7	8.1	4.80
4	3.1	55.9	18.0
6	4.2	119.7	28.5
8	5.4	202.7	37.5
12	8.1	457.1	56.4

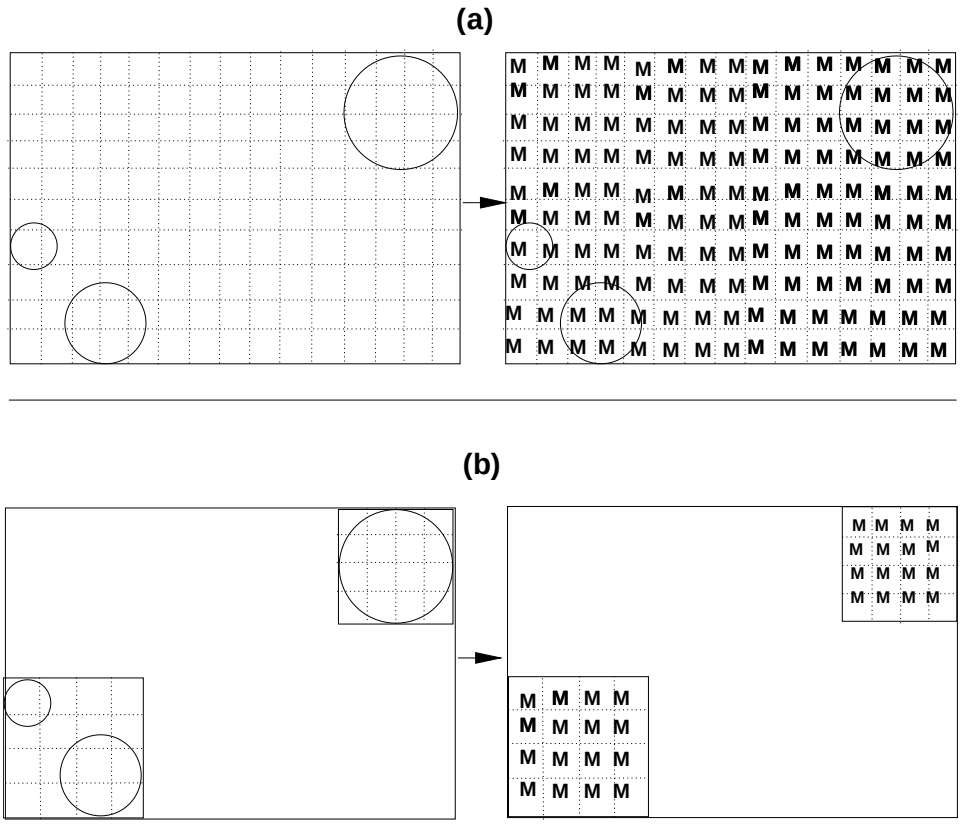

Figure 4.1: Comparison between FFTM (a) and FFTM Clustering (b) in discretizing the problem domain and calculations on each cell.

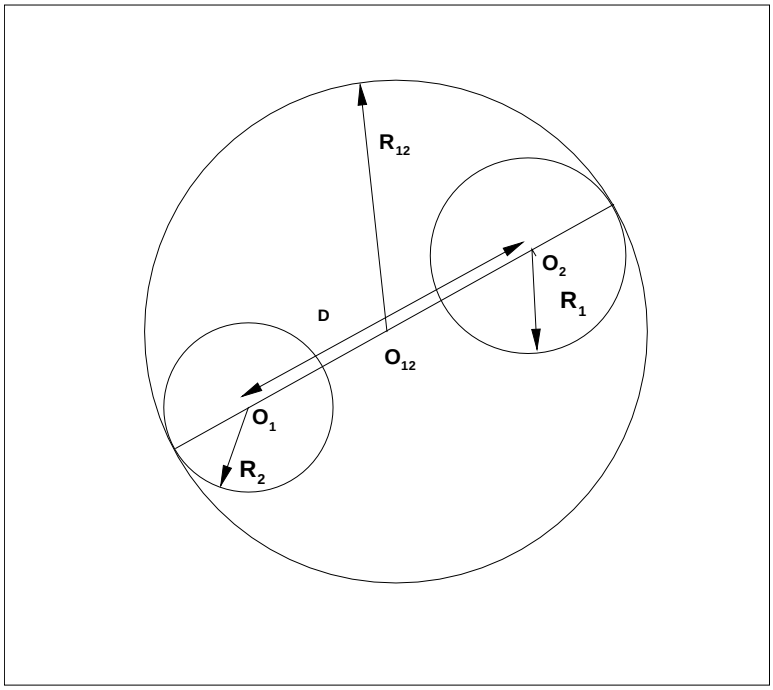

Figure 4.2: Grouping process of two spheroid centered at O_1 and O_2, radii R_1 and R_2.

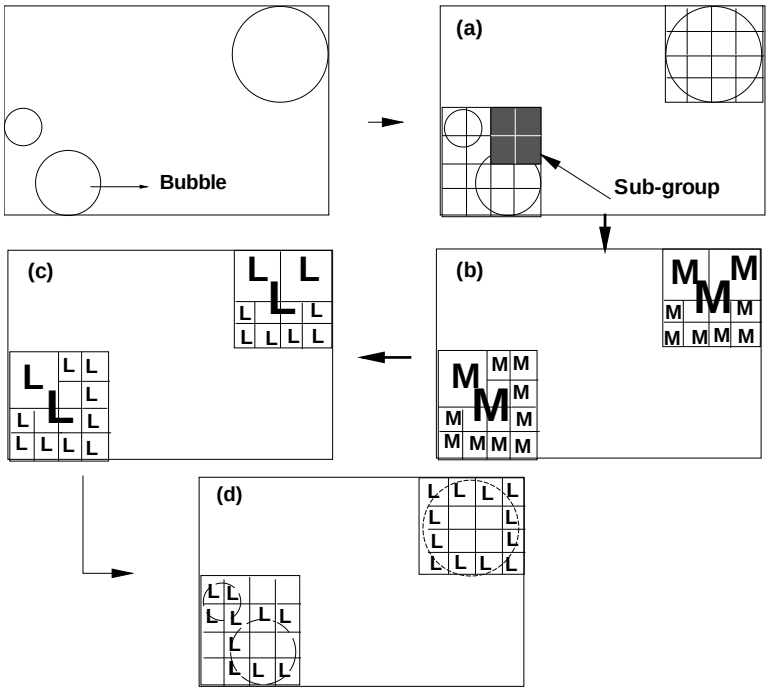

Figure 4.3: Two-dimensional illustration of FFTM Clustering method. (a) Domain Clustering. (b) Computing the multipole interactions among the sub-groups. (c) Computing the local expansion coefficients due to other groups. (d) Computing the local expansions due to contribution from cells inside group and outside group

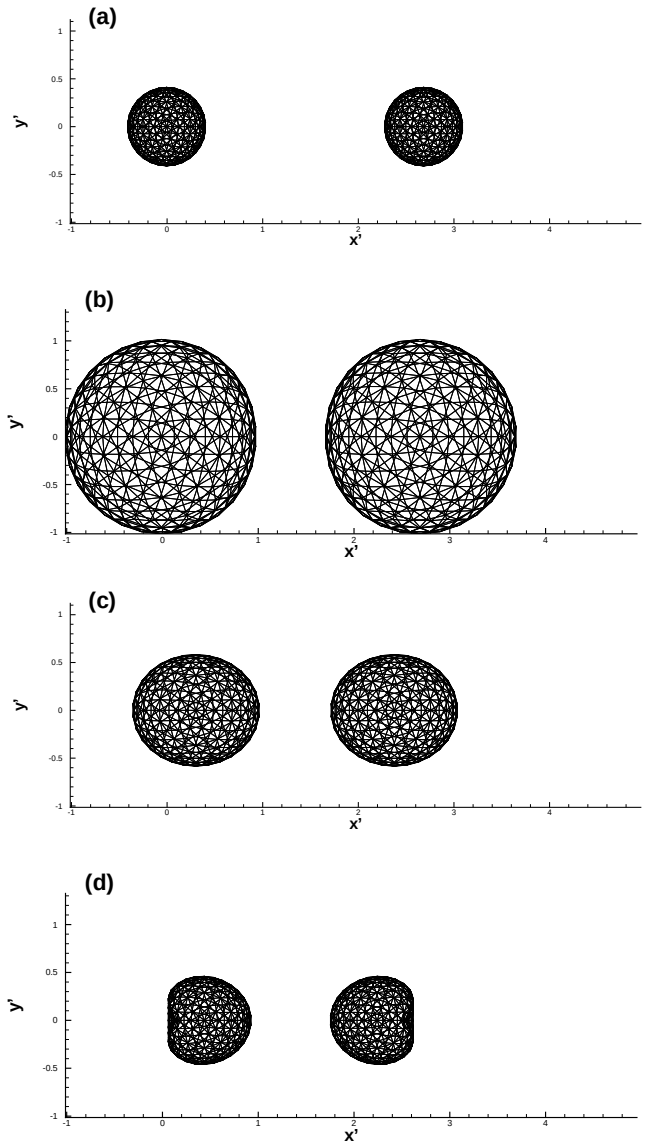

Figure 4.4: Evolution of two bubbles with initial dimensionless distance $d' = 2.6785$. Each bubble has 642 node and 1280 triangle elements on its mesh. Solutions are given by the FFTM Clustering method. (a) and (b) are the bubbles shapes during expansion phase at dimensionless time $t' = 0.080$ and $t' = 0.851$ respectively; (c) and (d) are the bubbles shapes during collapse phase at dimensionless time $t' = 2.010$ and $t' = 2.449$ respectively.

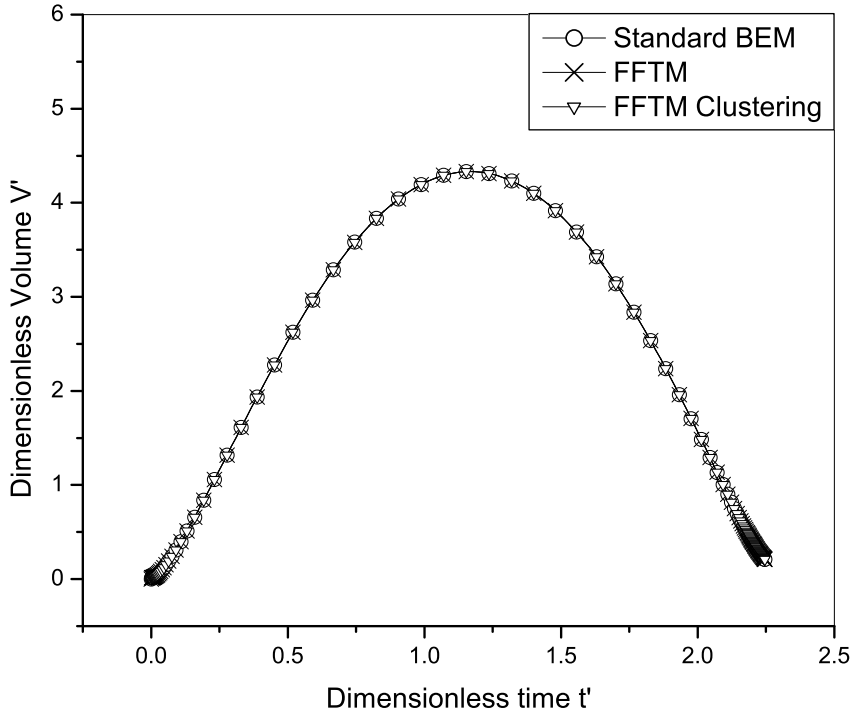

Figure 4.5: Comparison in dimensionless bubble volume produced by the FFTM, FFTM Clustering and standard BEM.

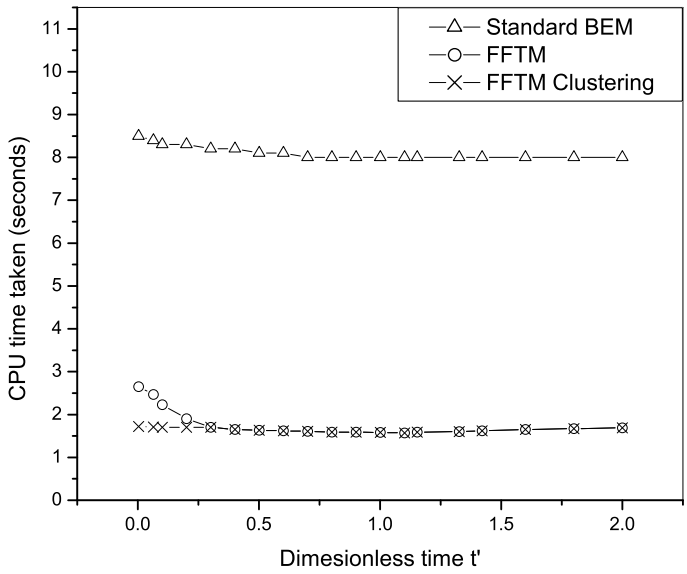

Figure 4.6: Comparison of CPU time for each time step taken by the FFTM, FFTM Clustering and the standard BEM method during the bubbles evolution. Distance between the centers of two initial bubbles is $d' = 2.6785$.

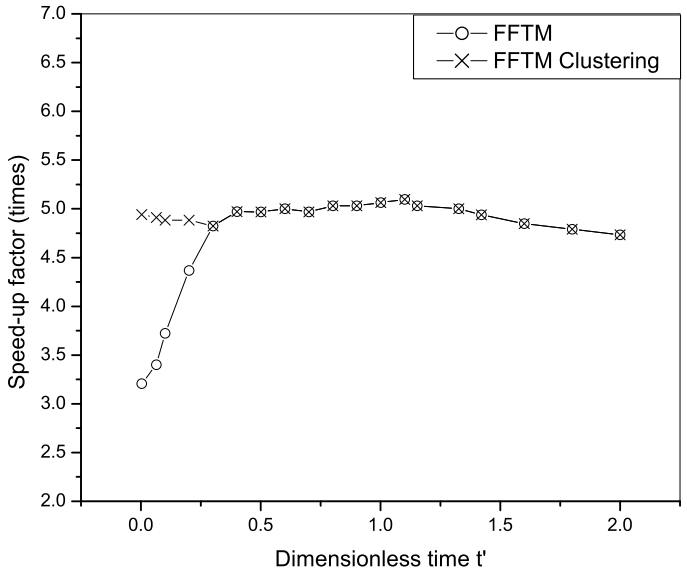

Figure 4.7: The speed-up factor of the FFTM and FFTM Clustering method during bubbles simulation. Distance between the centers of two initial bubbles is $d' = 2.6785$.

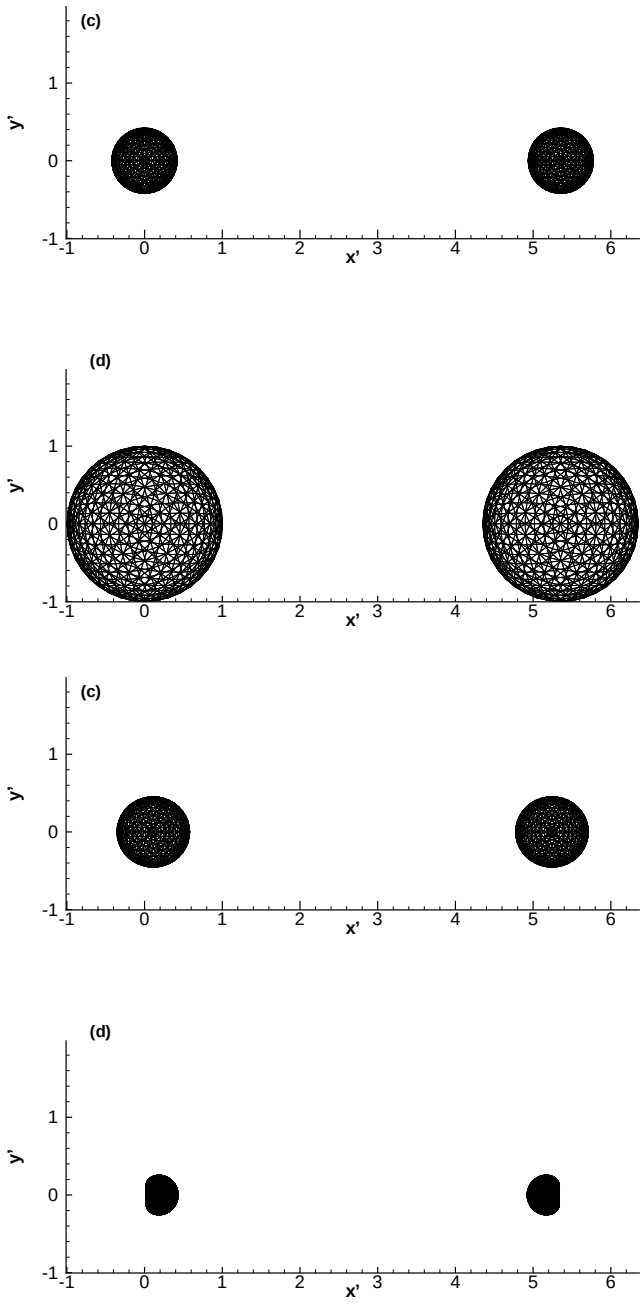

Figure 4.8: Evolution of two bubbles with initial centers distance of $d' = 5.357$. Each bubble has 642 node and 1280 triangle elements on its mesh. Solutions are given by the FFTM Clustering method. (a) and (b) are bubbles shapes during expansion phase at dimensionless time $t' = 0.083$ and $t' = 0.881$ respectively. (c) and (d) are bubbles shapes during collapse phase at dimensionless time $t' = 2.109$ and $t' = 2.138$ respectively.

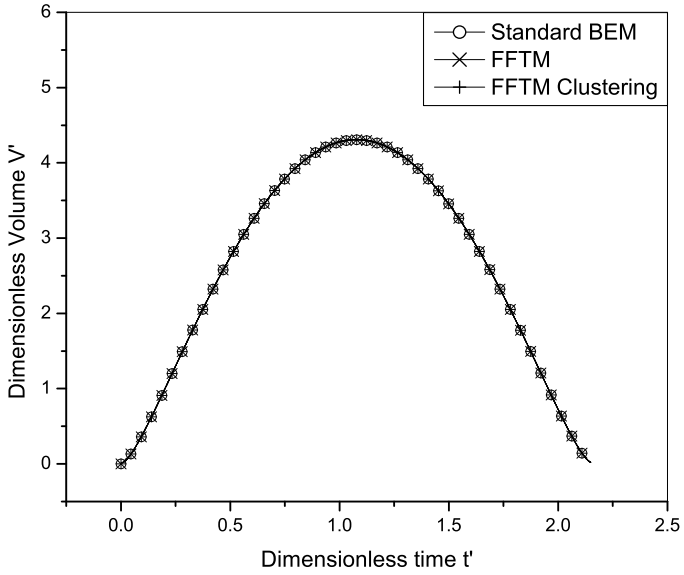

Figure 4.9: Comparison in dimensionless bubble volume produced by the FFTM, FFTM Clustering and standard BEM.

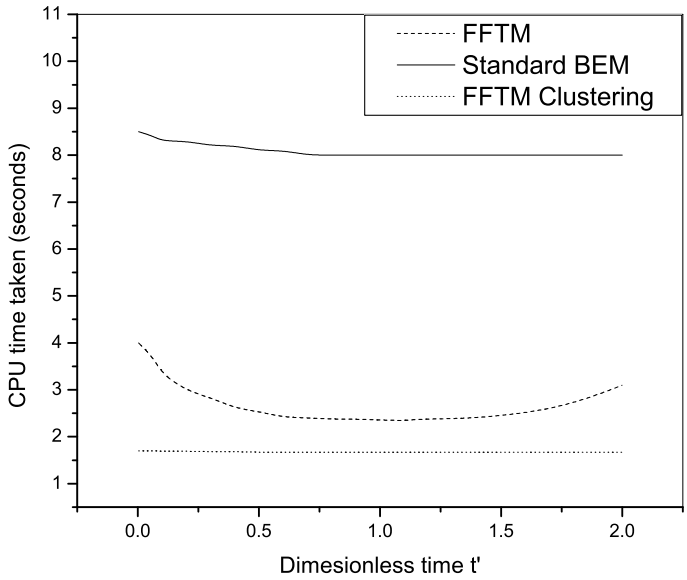

Figure 4.10: Comparison of time during the bubbles evolution for each time step taken by the FFTM, FFTM Clustering method and the standard BEM method. Distance between the centers of two initial bubbles is $d' = 5.357$.

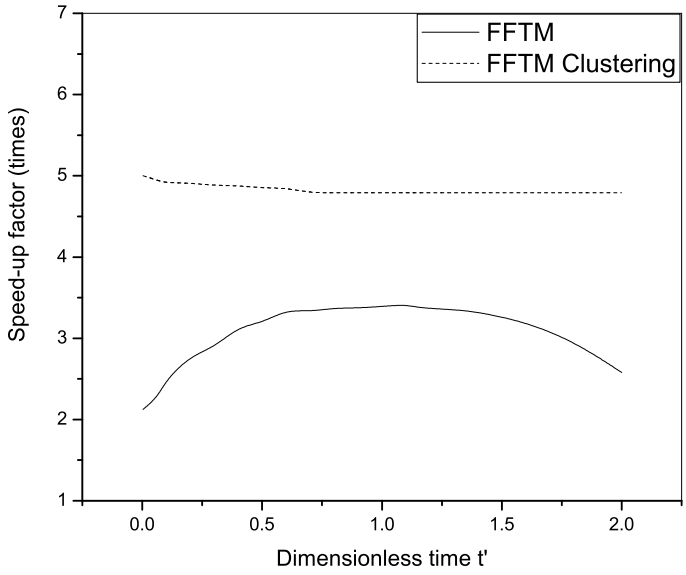

Figure 4.11: The speed-up factor of the FFTM and FFTM Clustering method during bubbles simulation. Distance between the centers of two initial bubbles is $d' = 5.357$.

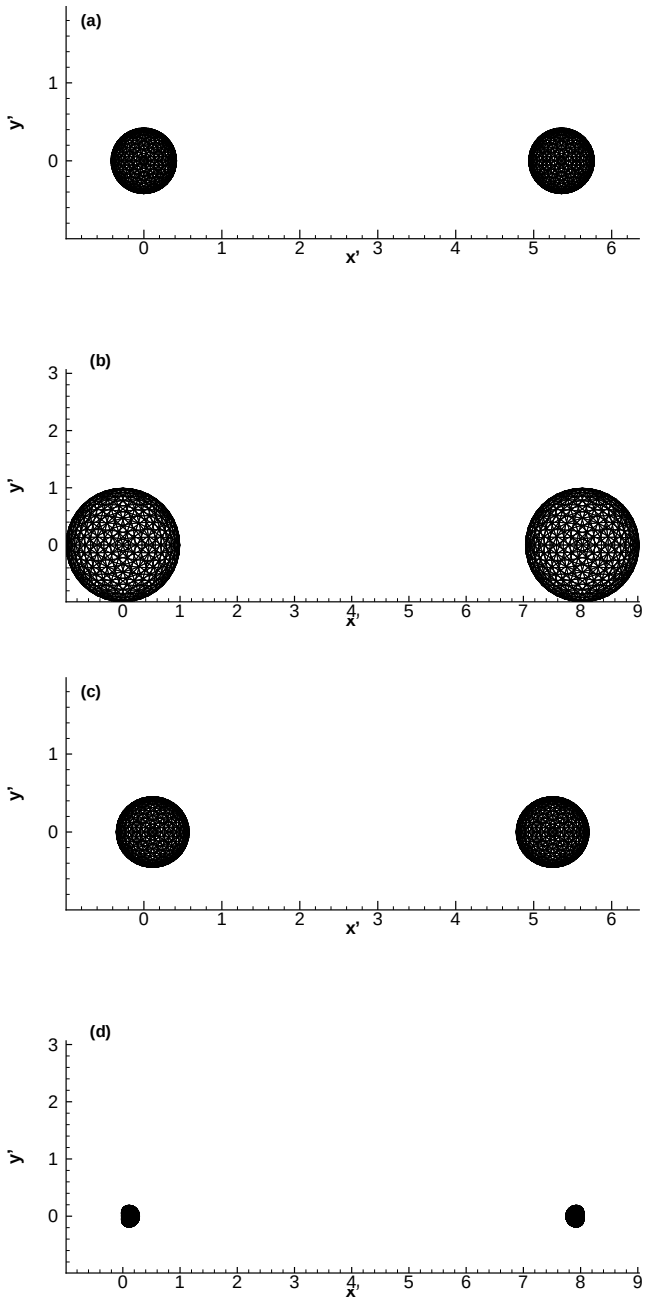

Figure 4.12: Evolution of two bubbles with initial centers distance of $d' = 7.95$. Each bubble has 642 node and 1280 triangle elements on its mesh. Solutions are given by the FFTM Clustering. (a) and (b) are bubbles shapes during expansion phase at dimensionless time $t' = 0.083$ and $t' = 0.887$ respectively. (c) and (d) are bubbles shapes during collapse phase at dimensionless time $t' = 2.109$ and $t' = 2.103$ respectively.

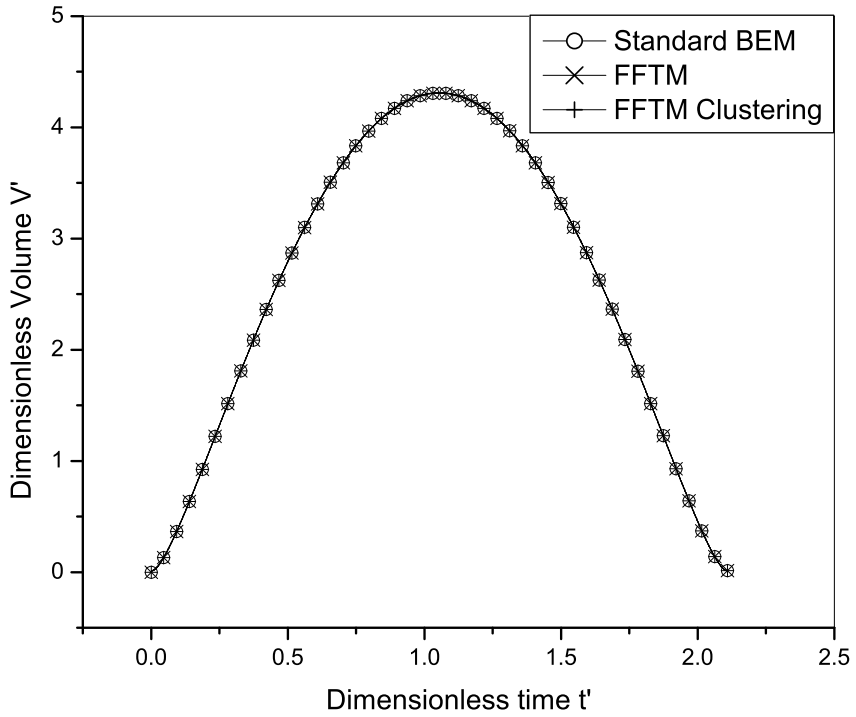

Figure 4.13: Comparison of dimensionless bubble volume produced by the FFTM, FFTM Clustering and standard BEM.

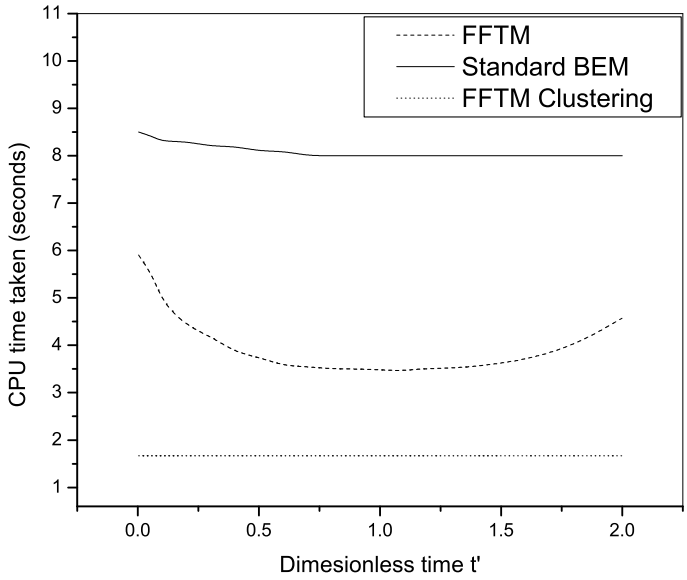

Figure 4.14: Comparison of CPU time taken by the FFTM, FFTM Clustering method and the standard BEM method. .

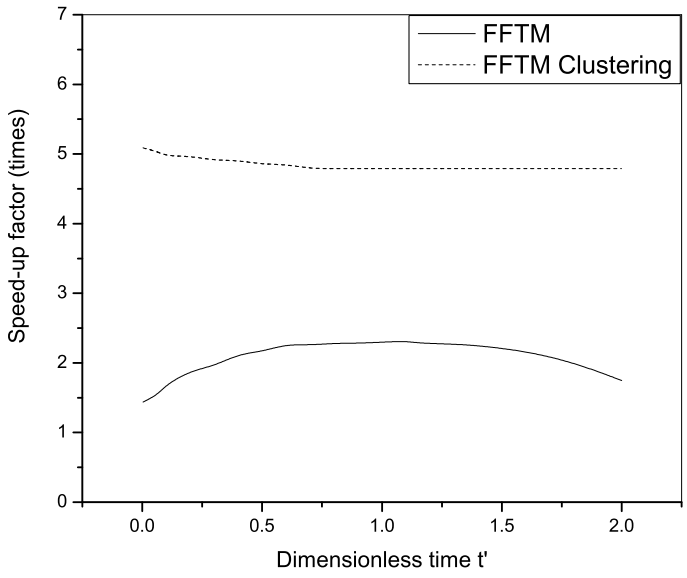

Figure 4.15: The speed-up factor of the FFTM and FFTM Clustering method during bubbles simulation. Distance between the centers of two initial bubbles is $d' = 7.95$.

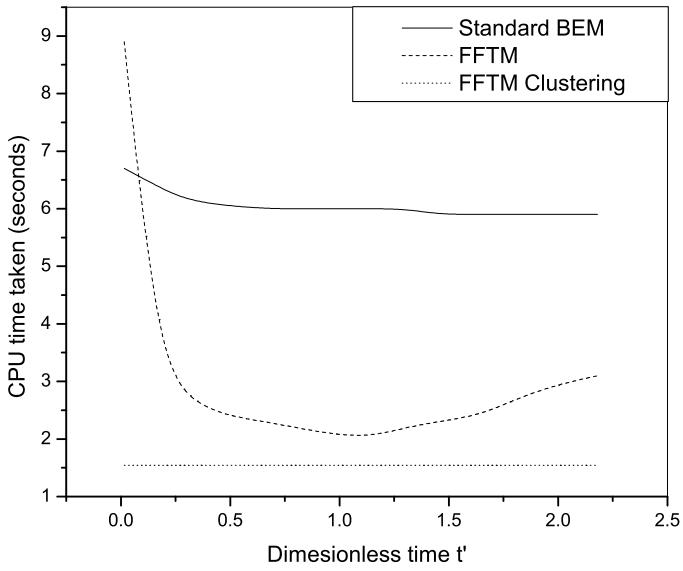

Figure 4.16: Comparison of time taken by using the FFTM, FFTM Clustering and standard BEM in each step during evolution of three bubbles.

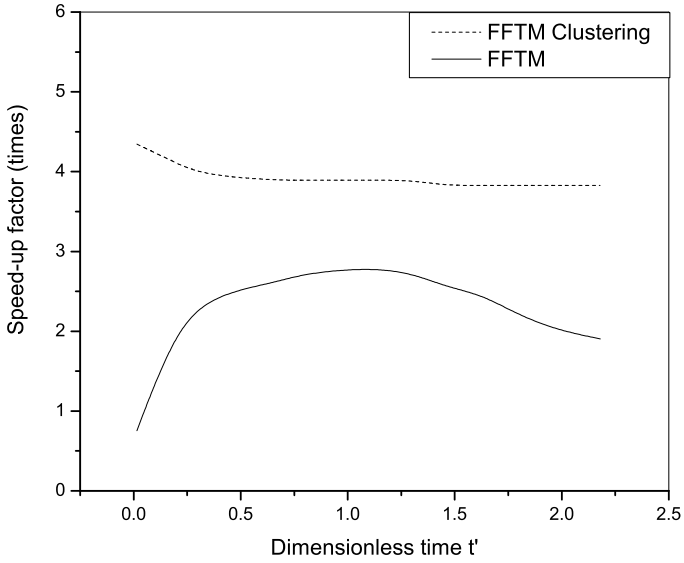

Figure 4.17: The speed-up factor of the FFTM and FFTM Clustering method during simulation of three bubbles.

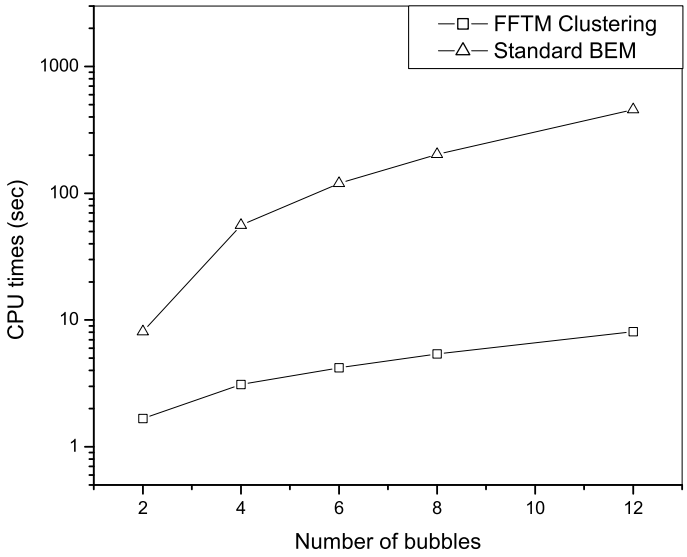

Figure 4.18: Comparison of CPU time taken in one step taken by the FFTM Clustering and the standard BEM.

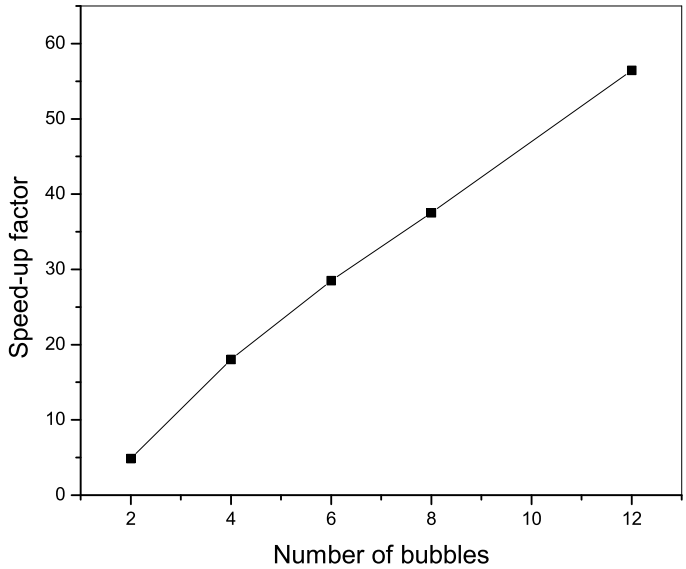

Figure 4.19: The speed-up factor of the FFTM Clustering.

Chapter 5

Multiple Bubbles Simulation

The finding on the direction of bubble-induced jet in the midst of multiple bubbles dynamics has important implication in various engineering applications. Take, for example, in ultrasonic cleaning of electronic or medical equipments and components, multiple bubbles of various sizes are created with accompanying bubble-induced jets directed towards the solid rigid surfaces for the removal of foreign bodies or otherwise. An optimum or closed-to optimum timing of bubbles initiation and distribution has yet to be worked out. Similarly, too, in the bio-medical application of shock lithotripsy treatment or the removal of kidney stone. It is suggested that the impact pressure of induced jets due to bubbles collapse near the solid wall is the main mechanism for the destruction of kidney stone. Therefore, a study of multiple bubble behavior is critical to our better understanding of the subject.

Simulation of multiple bubbles with large number of nodes have not been presented before due to the immense computational time and memory requirement of standard BEM. FFTM Clustering is an accurate and efficient method which allows

the simulation of multiple bubbles with high number of nodes on each bubble. In this section, we will present several case examples to study the physics of multiple explosion bubbles.

5.1 Three bubbles

Consider three explosion bubbles far away from any boundaries without gravitational effect. Centers of three bubbles are initially placed on the z-plane and at the three apexes of an equilateral triangle of dimensionless length $a' = 3.57$. Three bubble exploded at the same time. Each bubble is initiated from a charge of 500 kg of TNT placed at a depth of 100 m away from the free water surface. The maximum bubble radius will be 5.6 m for the 500 kg TNT charge. The parameter for all the bubbles are strength parameter $\varepsilon = 97.52$ and ratio of specific heat $\lambda = 1.25$. Each bubble starts to expand from the initial radius $R'_0 = 0.1499$. Each bubble is generated with 362 nodes and 720 triangle elements on its surface.

Figures 5.1(a)-(b) show the bubbles shapes during the expansion phase and Figures 5.2(a)-(b) show the bubbles shapes during the collapse phase. Three bubbles initially start to expand from very small volume until reaching the maximum volume at $V' = 4.2711$ (corresponding $R'_m = 1.011$) at $t' = 1.225$. Bubbles have spherical shape during the expansion phase then they collapse and form jetting directed towards the geometrical center.

In the next test, we consider three explosion bubbles far away from any boundaries without gravitational effect. The three bubbles were initiated from three identical

charges of 500 kg of TNT placed at a depth of 100 m away from the free water surface. The maximum bubble radius of each bubble will be 5.6 m. Three bubble exploded at the same time. Each bubble starts to expand from the initial bubble radius $R'_0 = 0.1499$. Other parameters for all the bubbles are strength parameter $\varepsilon = 97.52$ and ratio of specific heat $\lambda = 1.25$. Each bubble is generated with 362 nodes and 720 triangle elements on its surface. The initial centers location of each bubbles are $(0, 0, 0)$, $(2.68, 0, 0)$, $(5.36, 0, 0)$. These parameter have been chosen, in such a way, that the bubbles would not contact each other during their evolution.

Fig. 5.3 shows the bubbles shapes at the dimensionless time $t' = 0.459$. At this time bubbles are in the expansion phase and each of them keeps to their spherical shape. Fig. 5.4 shows the bubbles shapes at the dimensionless time $t' = 2.221$. It is obvious from Fig. 5.4 that three bubbles contract but the center bubble is stretched along the x-axis due to the presence of the outer bubbles. Figure 5.5 shows the bubbles shapes at the dimensionless time $t' = 2.303$. At the time $t' = 2.303$, the outer bubbles is in the collapse phase and these have the jets directed towards the center of the system.

Fig. 5.6 shows the bubbles volume during the simulation. The outer bubble reaches its maximum of $V' = 4.246$ at dimensionless time $t' = 1.182$, while the inner bubble reaches its maximum of $V' = 4.494$ at dimensionless time $t' = 1.509$.

5.2 Four bubbles

We consider four explosion bubbles placed under water far away from any boundaries without gravitational effect. Four bubbles was initiated at the same time from five charges of 500 kg of weight. Each bubble starts to expand from the initial bubble radius $R'_0 = 0.1499$. Other bubble parameters are strength parameter $\varepsilon = 97.52$ and ratio of specific heat $\lambda = 1.25$. Each bubble is generated with 362 nodes and 720 triangle elements on its surface. The initial centers location of each bubbles are $(2.68, 0, 0)$, $(-2.68, 0, 0)$, $(0, 2.68, 0)$, $(0, -2.68, 0)$. The reason for placing the bubbles so far away from each other is to add an additional fifth bubble in the next section, without the bubbles touching each other. Figs. 5.7 - 5.10 show the evolution of four bubbles from beginning until some part of the collapse phase. It is noted that the four bubbles expand with the fairly symmetrically shape until they reach about their maximum radii. Then each bubble begins its phase with the bubble surface deforming and rising a jet directed towards the center of the system.

5.3 Five bubbles

We consider five charges placed under water far away from any boundaries without gravitational effect. Five bubbles was initiated from five charges of 500 kg of weight. Each bubble starts to expand from the initial bubble radius $R'_0 = 0.1499$. Other bubble parameters are strength parameter $\varepsilon = 97.52$ and ratio of specific heat $\lambda = 1.25$. Each bubble is generated with 362 nodes and 720 triangle elements on its surface. The initial centers location of each bubbles are $(0, 0, 0)$, $(2.68, 0, 0)$, $(-2.68, 0, 0)$,

$(0, 2.68, 0)$, $(0, -2.68, 0)$. The location of five initial bubbles is shown in Fig. 5.11. In this figure, there is one bubble located at the center and surrounded by four other bubbles. Figures 5.11 - 5.14 show the evolution of five bubbles from beginning until some part of the collapse phase. It is noted that the five bubbles expand until they reach about their maximum radii. Then the outer four bubbles begin their collapse phase with the bubble surface deforming and rasing a jet directed towards the center of the system. At the meantime, the center bubble still keep to its relatively symmetrical shape.

Fig. 5.15 shows the bubbles volumes during the simulation. The outer bubbles reach its maximum of $V' = 3.893$ at dimensionless time $t' = 1.276$, while the inner bubble reaches its maximum of $V' = 5.0344$ at dimensionless time $t' = 1.775$.

In the next test, we simulate for five bubbles but do not explode or initiated all at the same time. The initial centers location of the four bubbles are $(2.68, 0, 0)$, $(-2.68, 0, 0)$, $(0, 2.68, 0)$, $(0, -2.68, 0)$. The outer four bubbles explode at the same time first, then at the dimensionless time of $t' = 1$ when the four bubbles nearly reach their maximum, the fifth bubble is initiated at the center of the system. The initial center location of the fifth bubble is $(0, 0, 0)$. Fig. 5.16 shows the shapes of the four bubbles at the dimensionless time $t' = 0.198$. The four bubbles expand symmetrically with spherical shapes. Then the fifth bubble initiated at dimensionless time $t' = 1$, and it also expanded symmetrically with spherical shape. Fig. 5.16 shows the shapes of the five bubbles at the dimensionless time $t' = 1.042$. Previous test of five bubble exploded at the same time demonstrated that the jetting of four outer bubble directed into the system. In this test, when the inner bubble exploded latter,

the jetting outer bubbles direct outward of the system. Figures 5.18 and 5.19 show the bubbles shapes at the dimensionless time $t' = 2.022$ and $t' = 2.194$, respectively. Figure 5.20 shows the shape of the outer bubble at time $t' = 2.194$ in different views.

Fig. 5.21 shows the dimensionless volumes of the inner bubble and the outer bubble. The outer bubble reaches its maximum of $V' = 4.182$ at dimensionless time $t' = 1.080$, while the inner bubble can reach the dimensionless volume of $V' = 9.795$. The maximum of the inner bubble is large than the outer bubble while they initiated from the same weight of charges. By initiating the fifth bubble at different time, it is possible to create jets directed away from the geometrical center. This is opposed to the center-directed jets as observed in Fig. 5.14. The implication for simulating multiple bubbles is, needles to say, tremendous.

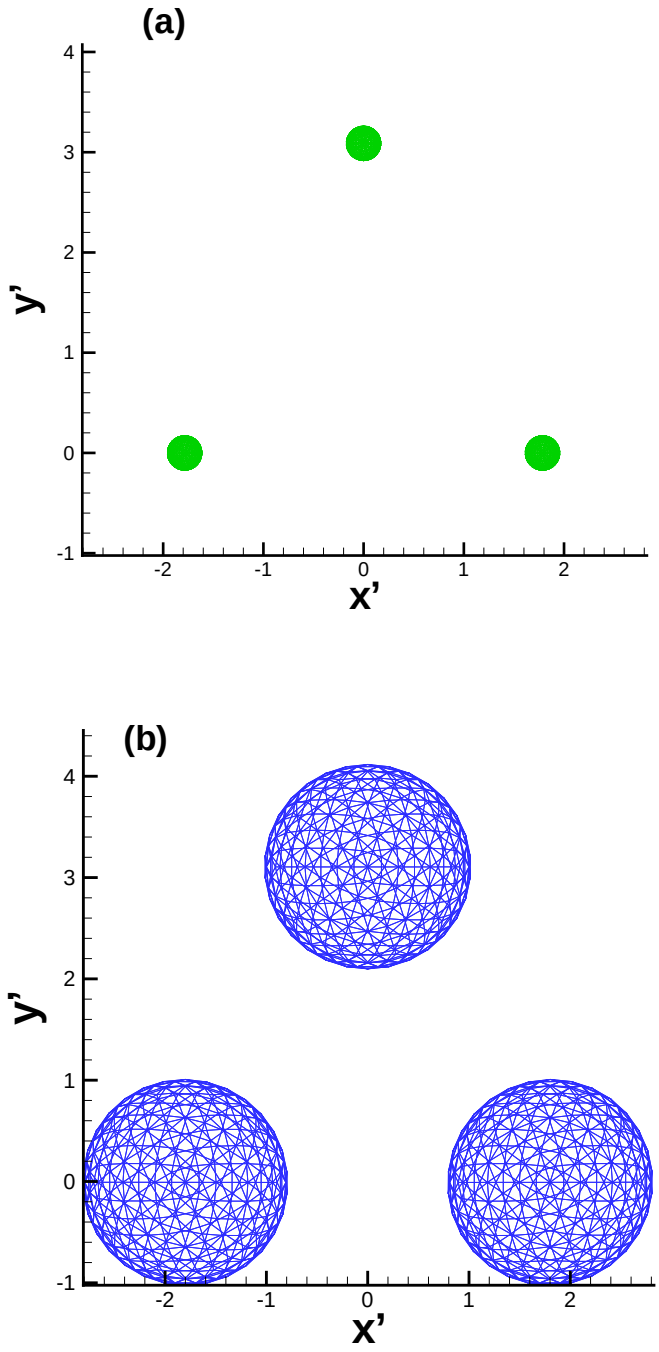

Figure 5.1: Evolution of three bubbles. Each bubble has 362 node and 720 triangle elements on its mesh. Solutions are given by FFTM. (a) and (b) are the bubbles shapes during the expansion phase at dimensionless time $t' = 0.011$ and $t' = 0.840$, respectively.

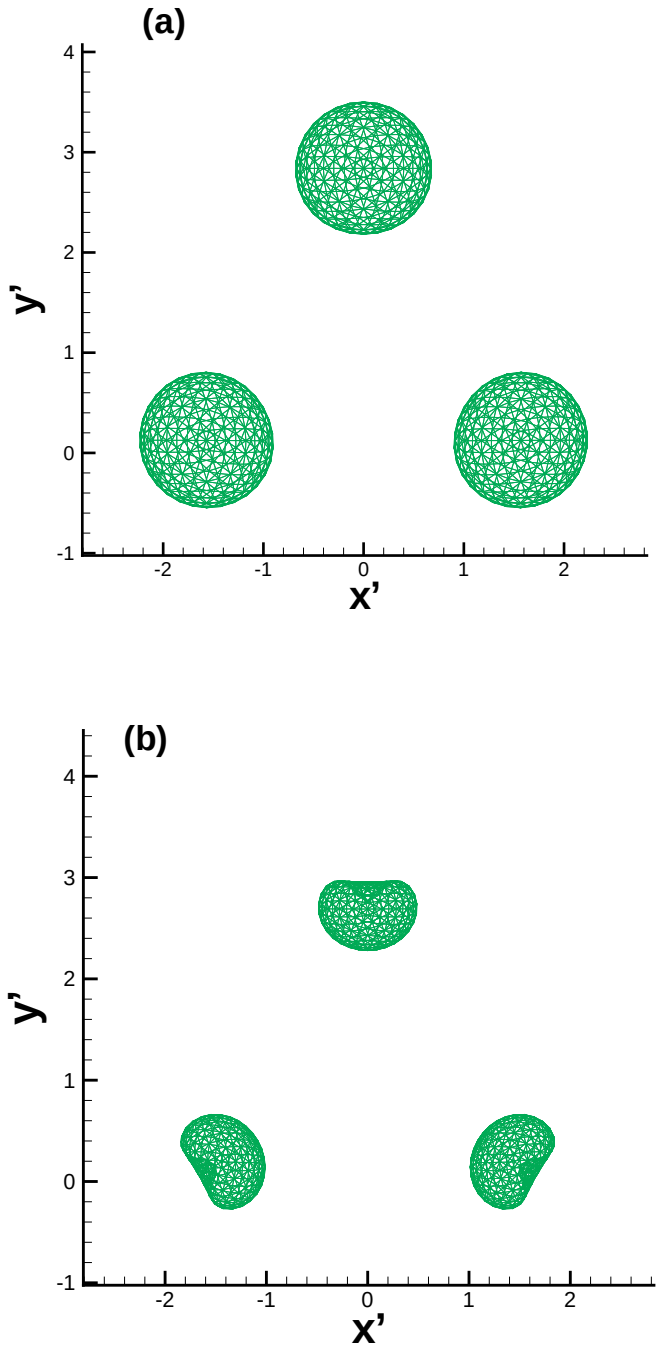

Figure 5.2: Evolution of three bubbles. Each bubble has 362 node and 720 triangle elements on its mesh. Solutions are given by FFTM. (a) and (b) are the bubbles shapes during the expansion phase at dimensionless time $t' = 2.176$ and $t' = 2.243$, respectively.

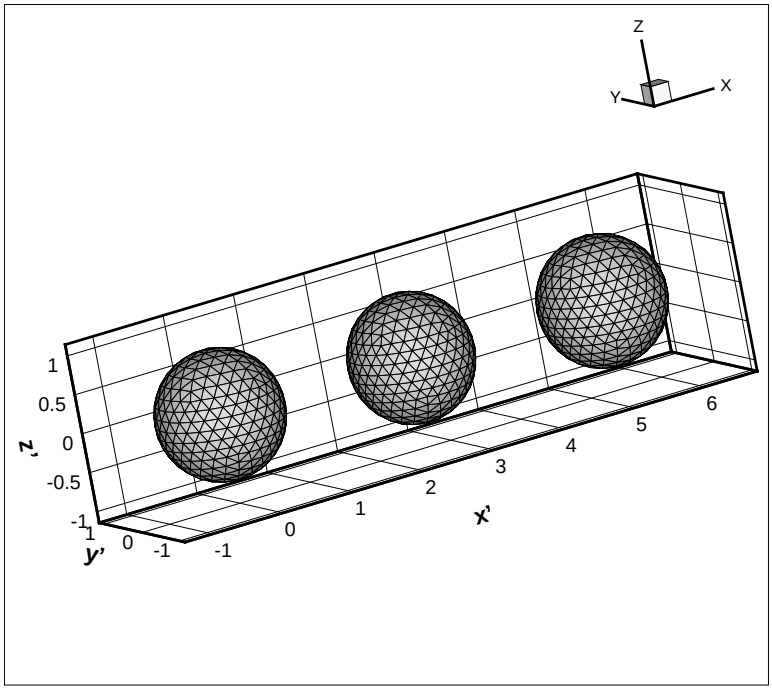

Figure 5.3: Bubbles shapes at dimensionless time t' = 0.459.

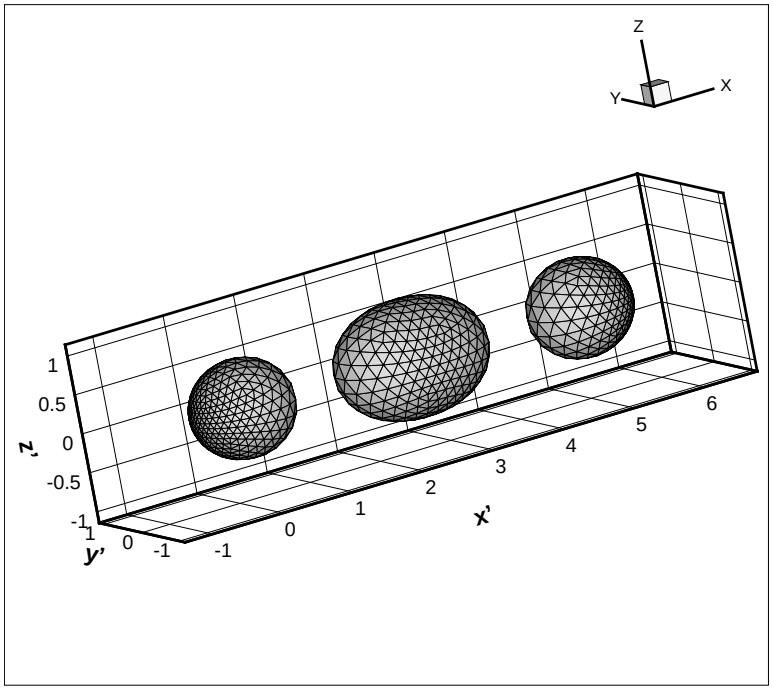

Figure 5.4: Bubbles shapes at dimensionless time t' = 2.139.

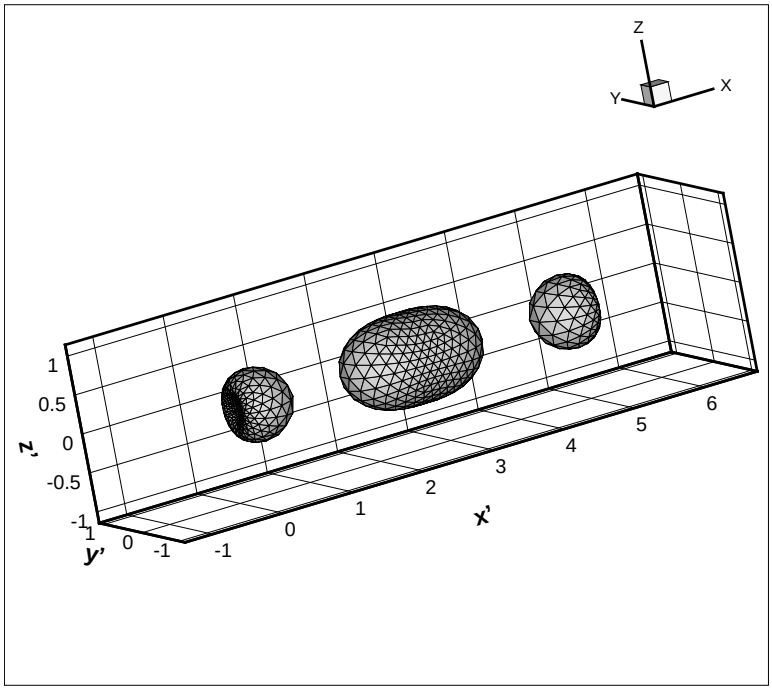

Figure 5.5: Bubbles shapes at dimensionless time t' = 2.303.

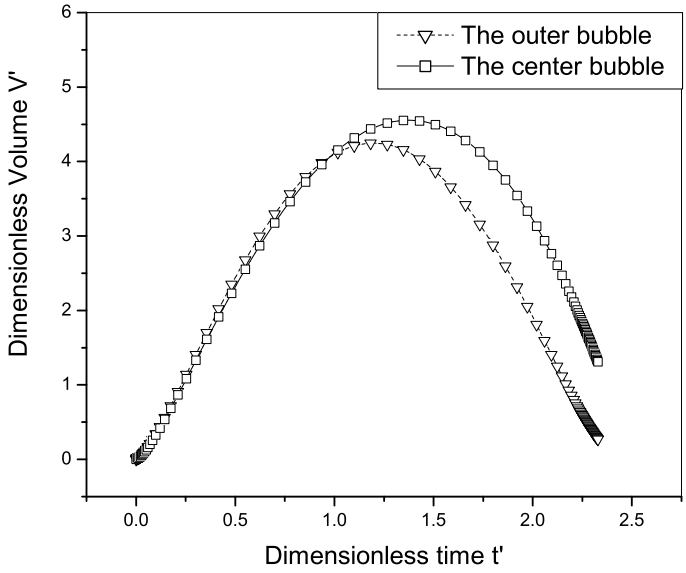

Figure 5.6: Dimensionless volume of bubbles during their evolution.

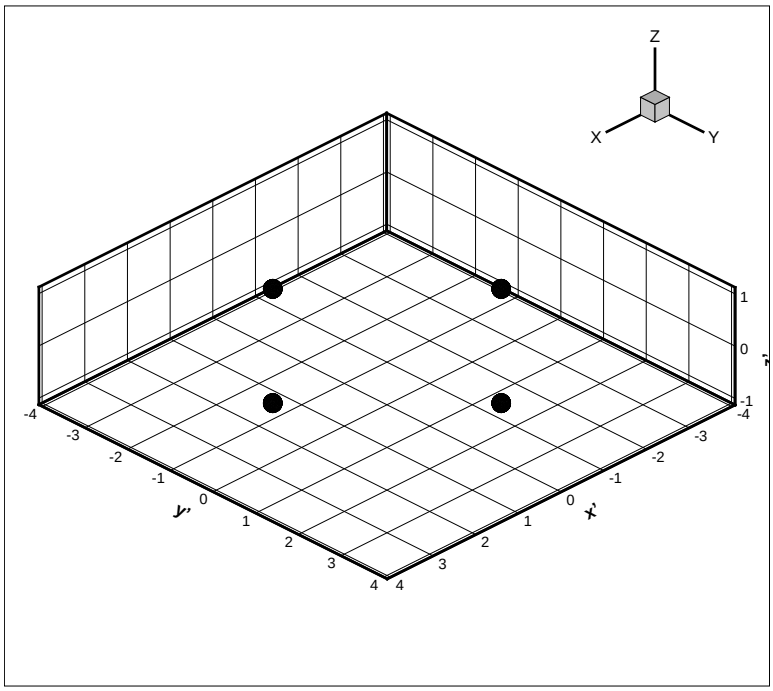

Figure 5.7: The location of four initial explosion bubbles under water at the same time and no gravitational effect.

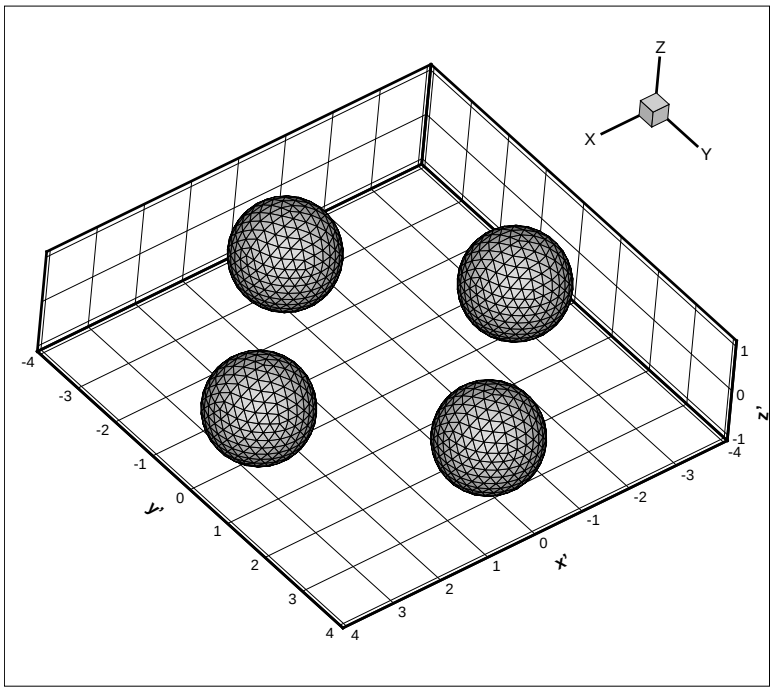

Figure 5.8: The shapes of five explosion bubbles under water at the same time and no gravitational effect at dimensionless time t' = 0.830.

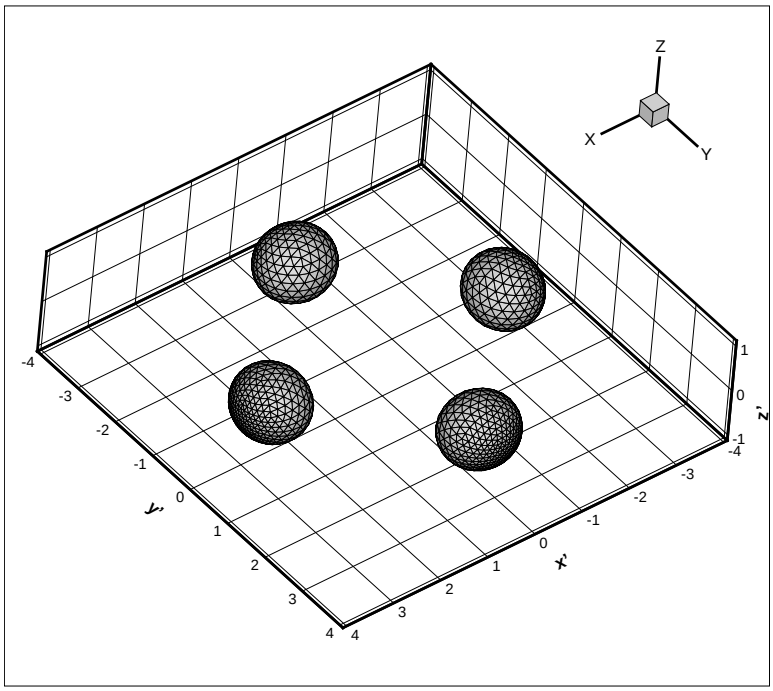

Figure 5.9: The shapes of five explosion bubbles under water at the same time and no gravitational effect at dimensionless time t' = 2.231.

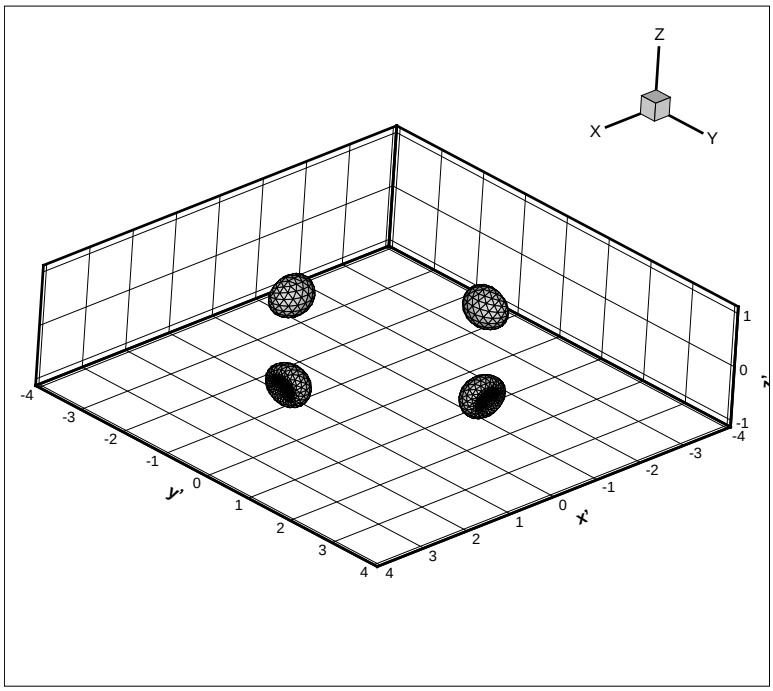

Figure 5.10: The shapes of five explosion bubbles under water at the same time and no gravitational effect at dimensionless time t' = 2.479.

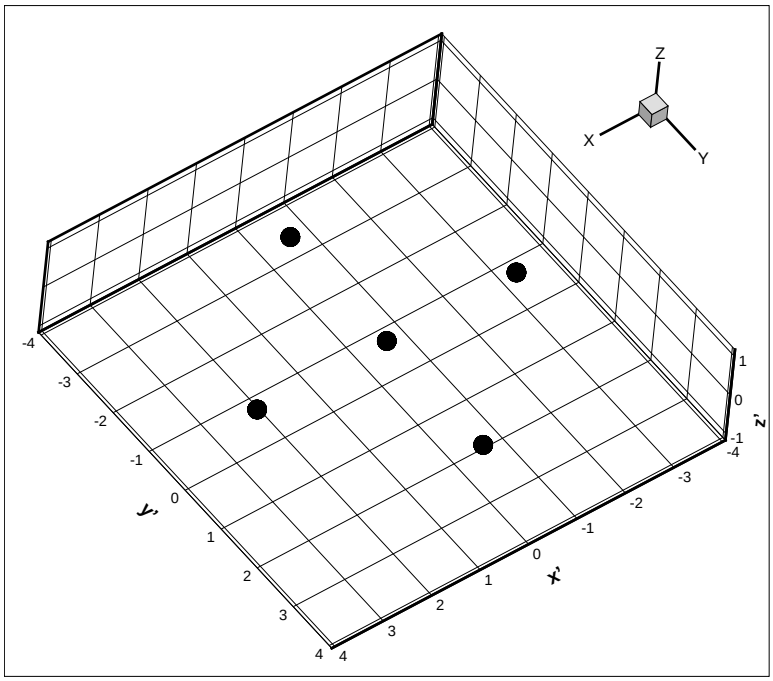

Figure 5.11: The location of five initial explosion bubbles under water at the same time and no gravitational effect.

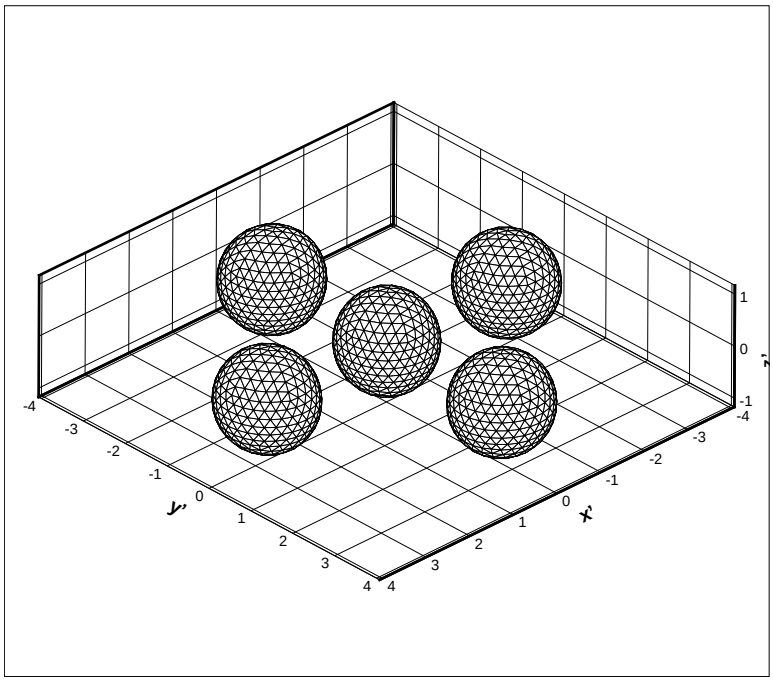

Figure 5.12: The shapes of five explosion bubbles under water at the same time and no gravitational effect at dimensionless time t' = 0.7683.

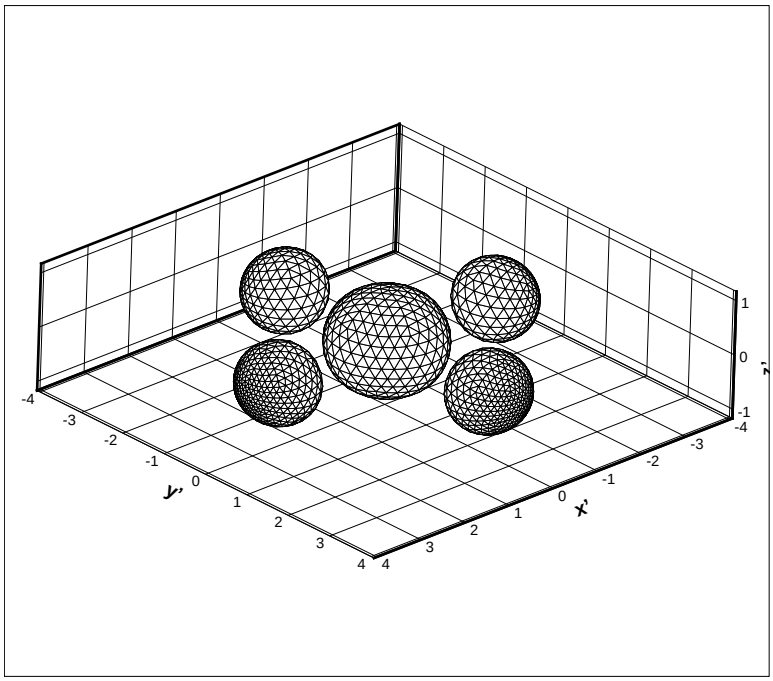

Figure 5.13: The shapes of five explosion bubbles under water at the same time and no gravitational effect at dimensionless time t' = 2.191.

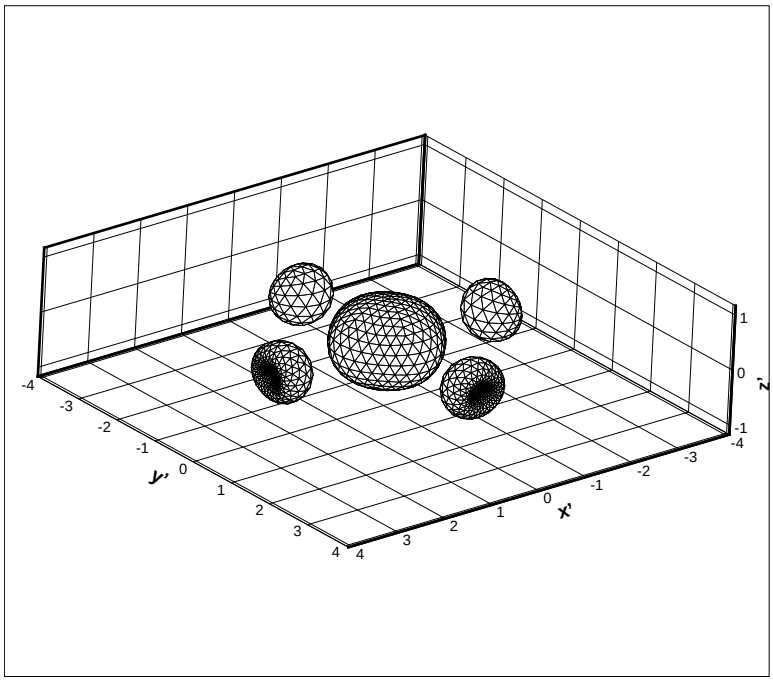

Figure 5.14: The shapes of five explosion bubbles under water at the same time and no gravitational effect at dimensionless time t' = 2.4736.

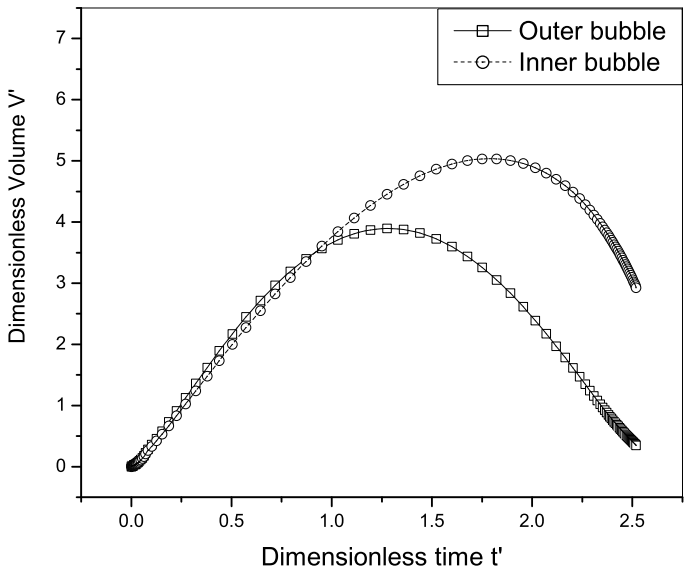

Figure 5.15: Dimensionless volume of bubbles during their evolution.

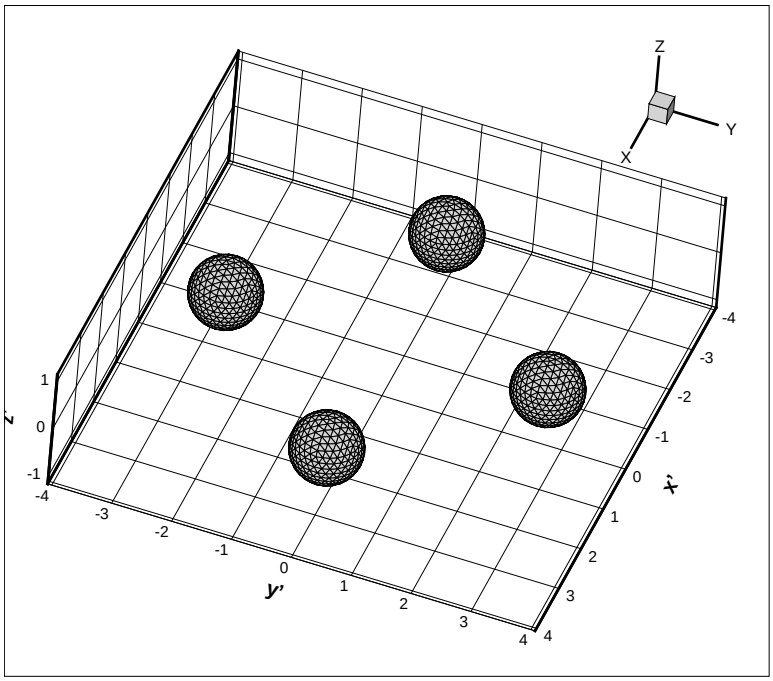

Figure 5.16: The shapes of four explosion bubbles under water without gravitational effect at dimensionless time t' = 0.198.

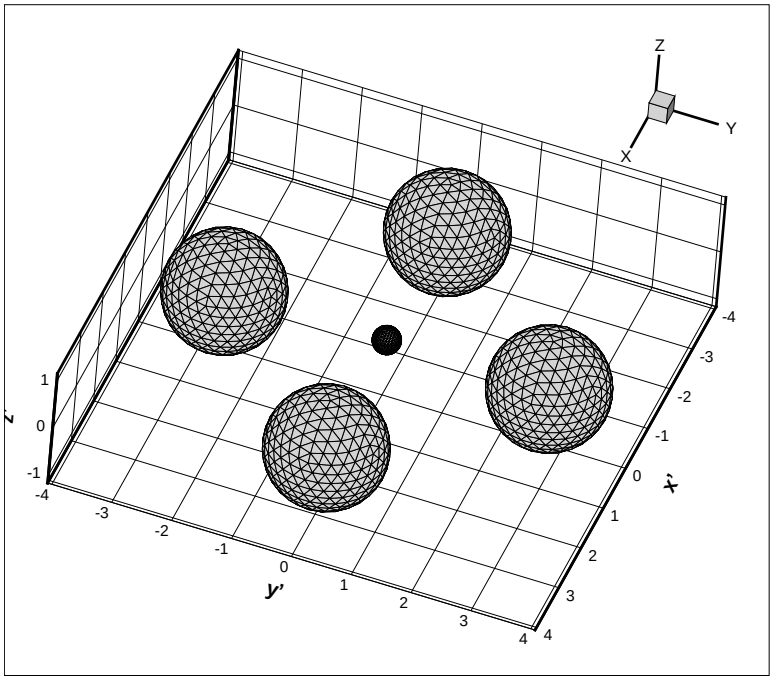

Figure 5.17: The shapes of five explosion bubbles under water without gravitational effect at dimensionless time t' = 1.042.

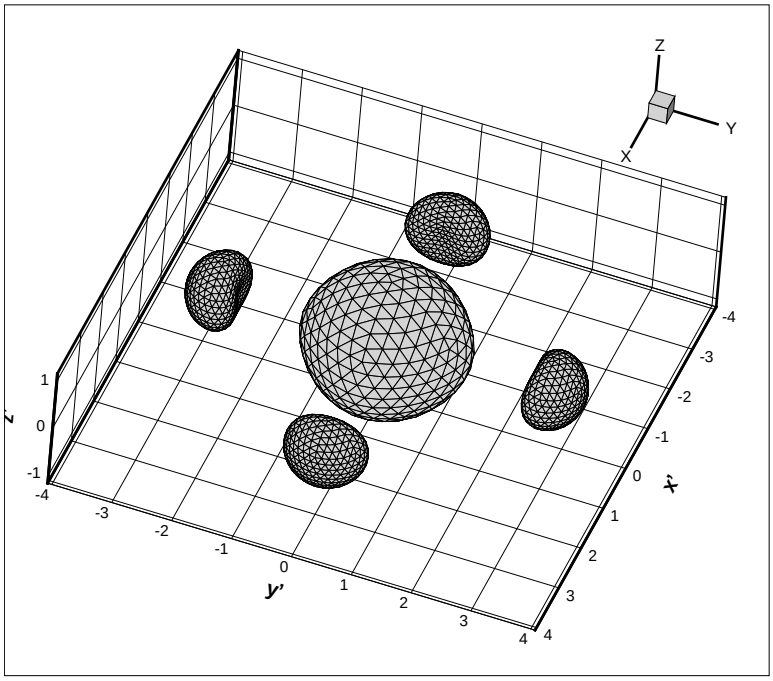

Figure 5.18: The shapes of five explosion bubbles under water without gravitational effect at dimensionless time t' = 2.022.

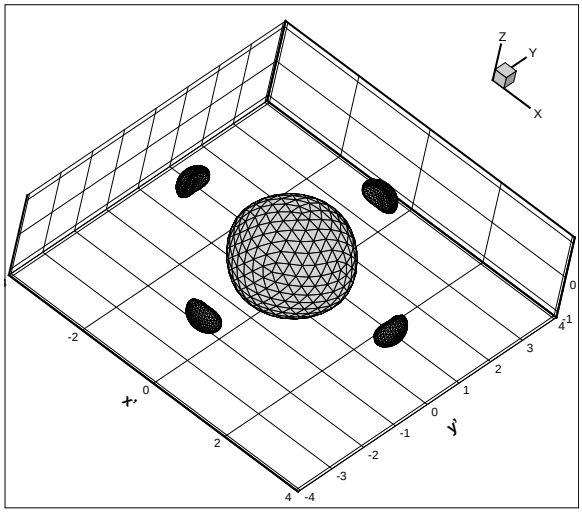

Figure 5.19: The shapes of five explosion bubbles under water without gravitational effect at dimensionless time t' = 2.194.

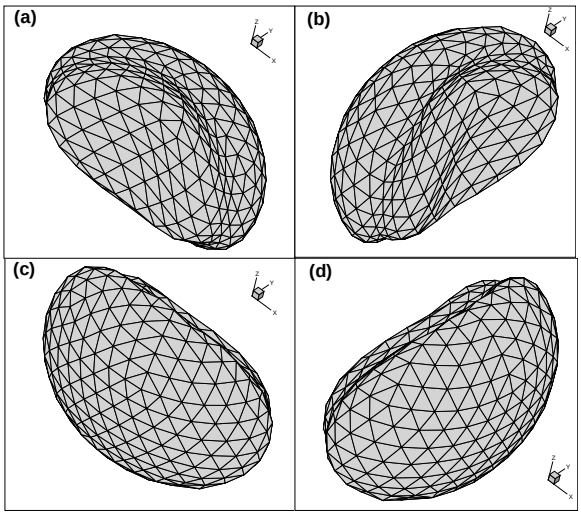

Figure 5.20: The shapes of outer bubbles at dimensionless time t' = 2.194 from different views.

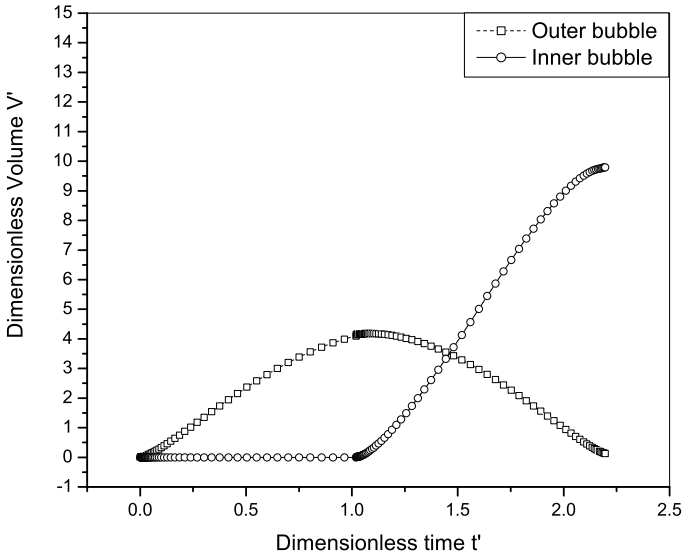

Figure 5.21: Dimensionless volume of bubbles during their evolution.

Chapter 6

Conclusions

The objective of the thesis is to provide an efficient simulation tool to study the physics of multiple bubbles. The major contributions of this thesis are briefly summarized below, and possible future work will be suggested, where appropriate.

Boundary Element Method provides a model with a high level of accuracy, but its computation is very intensive, and therefore not efficient and cost effective when simulating multiple bubbles with large number of nodes to approximate the bubbles surface. The numerical cost of the BEM is significantly reduced by coupling FFTM with standard BEM. Instead of setting up the fully populated BEM matrices, the matrix vector product is evaluated via multipole expansion and local expansion. In Chapter 3, the implementation of the FFTM algorithm to solve the boundary integral equation is presented. The comparison between the theoretical results and the numerical results produced by the FFTM on the Rayleigh bubble shows that FFTM is an accurate and efficient method. Further tests on two and three bubbles are conducted to compare the performance of the FFTM and the standard BEM. It is

shown that FFTM is an accurate and efficient method to simulate multiple bubbles dynamics. However, several tests on two bubbles placed at different distances apart show the drawback of the FFTM method. The efficiency of the FFTM deteriorates quite significantly when the problem is full of empty spaces if the multiple bubbles are well-separated. To overcome this limitation, a new version of FFTM Clustering is proposed in Chapter 4. FFTM Clustering is a generalized method which encompasses the original FFTM method. When the problem at hand has only one group, the FFTM Clustering is simply the original FFTM. The improvement of the FFTM Clustering arises from the clustering process and then compute the potential contributions from the bubbles within its own group, while the contributions from the other separated groups are evaluated via the multipole to local expansions translations operations directly. Comparisons between results produced by the FFTM Clustering method and the standard BEM on multiple bubbles show that FFTM Clustering method is an accurate method and has the same order of accuracy as the original FFTM method. The FFTM Clustering run faster than the original version and its efficiency is not reduced when there are plenty of blank spaces in the problem domain. The significant feature of the FFTM Clustering is to simulate the multiple bubbles in large scale problem in which the standard BEM is hardly employed due to the tremendous CPU time. In chapter 5, the physics of the multiple explosion bubbles under water is explored. It is shown that the jet direction depends on the bubbles configuration such as the weight of charges, and time of initiation of bubble explosion.

The present work is seen as a contribution to promote a fast BEM for industrial

applications. It is shown that the use of advanced numerical algorithms can significantly improve the efficiency of multiple bubble simulations. With the efficiency of the FFTM Clustering method, it is feasible to study the dynamics of multiple bubbles like the jet directions in future works.

References

[1] M. K. Bevir and P. J. Fielding , Numerical Solution of Incompressible Bubble Collapse With Jetting, Clerendon Press, 1974.

[2] J. R. Blake and D. C. Gibson, Growth and Collapse of a Vapour Cavity Near a Free Surface, J. Fluid Mech., Vol. 111, pages 123–140, 1981.

[3] J. R. Blake and D. C. Gibson, Cavitation Bubbles Near Boundaries, Ann. Rev. Fluid Mech. Vol. 19, pages 99–123, 1987.

[4] J. R. Blake, G. S. Keen, R. P. Tong and M. Wilson, Acounstic Cavitation: The Fluid Dynamics of Non-Sherical Bubbles. Phil. Trans. Roy. Soc. A, Vol. 375, pages 251–267, 1987.

[5] S. A. Wilkerson, A Boundary Integral Approach to Three-Dimensional Underwater Explosion Bubble Dynamics, Ph.D dissertation, Johns Hopkins University, 1992.

[6] R.H. Cole, Underwater Explosions, Princeton, NJ: Princeton University Press, 1948.

[7] C. E. Brennen, Cavitation and bubble dynamics, New York: Oxford University Press , 1995.

[8] L. Guerri and G. Lucca and A. Prosperetti, A Numerical Method for Dynamics of Non-Spherical Cavitation Bubbles, Proc. of the 2nd International Colloquium on Drops and Bubbles, pages 175-181, 1981.

[9] Q. X. Wang, K. S. Yeo, B. C. Khoo and K. Y. Lam, Strong Interaction Between a Buoyancy Bubble and a Free Surface, Theoretical Computational Fluid Dynamics, Vol. 8, pages 73–88, 1996.

[10] Q. X.Wang, K. S. Yeo, B. C. Khoo and K. Y. Lam, Nonlinear Interaction Between Gas Bubble and Free Surface, Computers & Fluids, Vol. 25, No. 7, pages 607–628, 1996.

[11] Y.L. Zhang , K.S. Yeo, B.C. Khoo and W.K. Chong, Simulation of Three-dimensional Bubbles Using Desingularized Boundary Integral Method, International Journal for Numerical Methods in Fluids, Vol. 31, No. 8, pages 1311–1320, 1999.

[12] Y.L. Zhang, K.S. Yeo, B.C. Khoo and W.K. Chong, Three-dimensional Computation of Bubbles Near a Free Surface, Journal of Computational Physics, Vol. 146, pages 105–123, 1998.

[13] S. Rungsiyaphornrat, E. Klaseboer , B.C. Khoo and K.S. Yeo, The Merging of Two Gaseous Bubbles with an Application to Underwater Explosion, Computer & Fluid, Vol. 32, pages 1049–1074, 2003.

[14] K. Evert, K. C. Hung, C. Wang, B. C. Khoo, P. Boyce, S. Debono, H. Charlier, Experimental and numerical investigation of the dynamics of an underwater explosion bubble near a resilent/ rigid structure, Journal of Fluid Mechanics (inpress).

[15] C. Wang, S. W. Fong, B. C. Khoo, K. C. Hung, Multiple bubbles dynamics using level set indirect boundary element method, ICTAM 2004, Abstracts book and CD–ROM, Proceedings, Gutkowski W. and Kowalewsky T. A. (eds.), 2004.

[16] K. C. Hung, C. Wang, K. Evert, B. C. Khoo, Modelling and simulation of explosion bubble dynamics and its effect on submerged strutures, 74 shock and vibration symposium, San Diego, USA, 2003

[17] J.P. Best and A. Kucera, A Numerical Investigation of Nonspherical Rebounding Bubbles, J. Fluid Mech., Vol. 245, pages 137–154, 1992.

[18] Rayleigh, On the Pressure Developed in a Liquid during the Collapse of a Spherical Cavity, Phil. Mag., Vol. 34, pages 94–98, 1917.

[19] T. B. Benjamin and A. T. Ellis, Cavitation: The collapse of cavitation bubble and pressures thereby produced against solid boundaries, Philosophical Transactions of Royal Society of London, Vol. 260, pages 221–240, 1966.

[20] W. Lauterborn and H. Bolle, Experimential Investigations of Cavitation Bubble Collapse in the Neighbourhood of a Solid Boundary, J. Fluild Mech., Vol. 72, pages 391–399, 1975.

[21] C. L. Kling and F. G. Hammitt, A Photographic Study of Spark-Induced Cavitation Bubble Collapse, J. Basic Eng., Vol. 94, pages 825–833, 1972.

[22] R. B. Chapman and M. S. Plesset, Nonlinear Effects in the Collapse of a nearly Spherical Cavity in a Liquid, J. Basic Eng., Vol. 94, pages 142–146, 1972.

[23] P. Holmes, J. L. Lumley and G. Berkooz, Turbulence Coherent structures, Dynamical Systems and Symmetry, Cambridge University Press, 1996.

[24] L. Sirovich, Turbulence and the Dynamics of Coherent Structures, Part 1: Coherent Structures, Quarterly of Applied Mathematics, Vol. 45, No. 3, pages 561–571, October 1987.

[25] Y. Saad and M.H. Schultz, GMRES: A Linear Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, SIAM J. Sci. Stat. Comput., Vol. 7, pages 856–869 , 1986.

[26] A. W. Appel, An Efficient Program for Many-Body Simulations, SIAM J. Sci. Statist. Comput., Vol. 6, no. 85, pages 85–103, 1985.

[27] J. Barnes and P. Hut, A Hierarchical $o(nlogn)$ Force Calculation Algorithm, Nature 324, Vol. 446, 1986.

[28] L. Greengard and V. Rokhlin, A Fast Algorithm for Particle Simulations, J. Comput. Phys., Vol. 73, 1987.

[29] L. Greengard and V. Rokhlin, A New Version of the Fast Multipole Method for the Laplace Equation in three Dimension, Acta Numerica, Vol. 6, 1997.

[30] K. Nabors, F.T. Korsmeyer, F.T. Leighton, J. White, Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory, SIAM J Sci Stat Comput, Vol. 15, 1994.

[31] S.K. Nabors and J. White, Fast Capacitance Extraction of General Three-Dimensional Structures, IEEE Trans. Microwave Theory Tech., Vol. 6, 1496-1506, Nov. 1992.

[32] C. T. Kelley, Iterative Methods for Linear and Nonlinear Equations. SIAM Frontiers in Applied Mathematics. SIAM, Philadelphia, no. 16, 1995.

[33] Y. Saad, Iterative Methods for Sparse Linear Systems, Boston:, PWS Pub Co; 1996.

[34] E.O. Brigham, The fast Fourier Transform and its Applications, Englewood Cliffs, NJ: Prentice-Hall; 1988.

[35] B.A. Luty, W.F. Gunstersun, Calculating Electrostatics Interactions Using Particle-Particle-Particle-Mesh Method with non Periodic Long-Range Interactions, J. Chem. Phys., Vol. 100, 1996.

[36] J.R. Phillips, J. White, A Precorrected-Fft Method for Electrostatics Analysis of Complicated 3-D Structures, IEEE Trans Comput Aided Design, Vol. 16, 1059–72, 1997.

[37] S. Kapur, D. E. Long, IES3: Efficient Electrostatic and Electromagnetic Simulation, IEEE Comput. Sci. Eng., 60–67, 1998.

[38] P. L. Levin, M. Spasojevic, R. Schenider, Creation of Sparse Boundary Element Matrices for 2-D and Axis-Symmetric Electrostatics Problems using the Bi-Orthogonal Haar Wavelet, IEEE Trans. Dielectric Elect. Insulation, Vol. 5, 469–484,1998

[39] A. Brandt, M. Spasojevic, R. Schenider, Multilevel Matrix Multiplication and Fast Solution of Integral Equations, J. Comput. Phys., Vol. 90, 348–370,1990

[40] E.T. Ong, K.M. Lim and H.P. Lee, A Parallel Fast Fourier Transform on Multipole (Fftm) Algorithm for Electrostatics Analysis of Three-Dimensional Structure, IEEE Transactions on Computer-Aided Design of integrated circuits and systems, Vol. 7, 1063–1072, 2004.

[41] E.T. Ong, K.M. Lim, K.H. Lee and H. P. Lee, A Fast Algorithm for Three-Dimensional Potential Fields Calculation: Fast Fourier Transform on Multipoles (FFTM), J. Comput. Phys., Vol. 192, 244–261, 2003.

[42] T.J Rudolphi and Y.J. Liu, New Identities for Fundamental Solutions and Their Applications to Non-Singular Boundary Element Formulations, Computat. Mech., Vol. 24, 286-292, 1999.

[43] M. Epton and B. Dembart, Multipole translation theory for the three-dimensional Laplace and Helmholtz equations, SIAM J. Sci, Vol.16, 865, 1995.

[44] A. W. Appel, Handbook of Mathematical Functions, Dover Publications, New York, 1974.

[45] Frigo M., Johnson S.G., FFTW, C subroutines library for computing discrete Fourier transform (DFT), [Online] Available: http://www.fftw.org.

Appendix A

Recurrence formulas

Given $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$ and $y = ||\mathbf{x}||$, the recurrence formulas for the inner and outer functions can be derived from those of the associated Legendre functions [44].

The inner function, $I_n^m(x)$ is written in the form:

$$I_n^m = \frac{1}{2m}(-x_1 + ix_2)I_{n-1}^{m-1}, \tag{A.1}$$

$$I_n^m = \frac{1}{(n-m)(n+m)}(-x_1 + ix_2)[(2n-1)x_2 I_{n-1}^m - r^2 I_{n-2}^m]. \tag{A.2}$$

The Outer function, $O_n^m(x)$ is written in the form:

$$O_n^m = \frac{(2m-1)}{r^2}(-x_1 - ix_2)O_{n-1}^{m-1}, \tag{A.3}$$

$$O_n^m = \frac{1}{r^2}\left[(2n-1)x_2 O_{n-1}^m - (n-m-1)(n+m-1)O_{n-2}^m\right]. \tag{A.4}$$

The derivative of inner function $I_n^m(\mathbf{x})$, is written as:

$$\frac{\partial}{\partial x_1} I_n^n = \frac{1}{x_1^2 + x_2^2} \left[(n x_1 + i m x_2) I_m^n - x_1 x_3 I_{n-1}^n \right], \tag{A.5}$$

$$\frac{\partial}{\partial x_2} I_n^n = \frac{1}{x_1^2 + x_2^2} \left[(n x_2 + i m x_1) I_m^n - x_2 x_3 I_{n-1}^n \right], \tag{A.6}$$

$$\frac{\partial}{\partial x_3} I_n^n = I_{n-1}^n. \tag{A.7}$$