The adaptive cross approximation accelerated boundary element method for bubble dynamics

Z. Fu & V. Popov
Wessex Institute of Technology, Environmental and Fluid Mechanics, UK

Abstract

A model based on the adaptive cross approximation (ACA) accelerated boundary element method (BEM) is presented for solving bubble dynamics problems. The computational solution of multiple bubble dynamics problems has high CPU requirements since it involves moving gas–liquid phase interfaces. In order to efficiently solve such problems, a fast algorithm, i.e. adaptive cross approximation, is implemented to compress the induced collocation matrix. An efficient binary-bit key system is applied to build up a hierarchical tree structure for the discretized boundary. With the aid of the key system, the dense matrix is partitioned into blocks which satisfy the condition of admissibility or contain only one row/column. The implemented ACA-BEM numerical technique is verified using the Rayleigh-Plesset equation and it is shown to be of linear complexity.
Keywords: bubble dynamics, boundary element method, adaptive cross approximation.

1 Introduction

Among all the numerical techniques, the Boundary Element Method (BEM) has been particularly favoured for study of bubble dynamics. Analysis of bubble evolution involves a geometrical change of bubble surface and transition of the bubble; hence it can be represented as a moving boundary problem. The use of the BEM allows for discretization of the bubble surface only, which significantly simplifies the re-meshing process and therefore has been widely employed by many researchers for the solution of this problem (e.g. Blake *et al.* [1], Chahine [2], and Khoo *et al.* [3]).

WIT Transactions on Modelling and Simulation, Vol 53, © 2012 WIT Press
www.witpress.com, ISSN 1743-3533 (on-line)
doi:10.2495/BE120191

BEM has shown to be more advantageous than the other numerical techniques e.g. finite element method (FEM), finite difference method (FDM) and finite volume method (FVM) in terms of both accuracy and efficiency for relatively moderate size of problems. As geometry of the problems become complicated, requiring complex meshes, both the computational cost and storage are very high due to the fully populated matrix. Therefore, fast numerical approaches have been developed to combat the drawbacks of the so called Conventional Boundary Element Method (CBEM). A brief review of the fast methods can be found elsewhere [4].

Algorithm called FFTM was presented by Ong *et al.* [5], which evaluates the potential by expressing the translation operator as a series of discrete convolutions of the multipoles. The Fast Fourier Transform (FFT) is applied to approximate the convolutions. The FFTM method has been successfully applied by Bui *et al.* [6] for modelling multiple bubble dynamics, with clustering to fit in with the property of sparseness of the physical problem. It proves computationally effective and requires storage of O(N) (N is the degree of freedom).

The FFTM method involves the procedure of approximating kernels. The form of the kernel varies with different physical problems and approximation of the kernel brings about complication during numerical implementation. Another kind of fast algorithms, referred to as the adaptive cross approximation (ACA), will be discussed in this paper. ACA starts from the point of the algebraic system. The matrix is compressed by applying the algorithm for blocks that satisfy the condition of admissibility.

2 Description of the problem

Bubbles in an ideal (homogeneous, incompressible, irrotational and inviscid) liquid medium are considered. The following governing equation is valid:

$$\Delta\phi = 0, \tag{1}$$

where ϕ denotes the velocity potential and Δ is the Laplacian. The developed model is based on the indirect boundary element method (IBEM):

$$\phi(\mathrm{x}) = \int_{\partial\Omega} \sigma(\xi) \cdot \mathrm{G}(\mathrm{x}, \xi) \cdot \mathrm{dS}(\xi), \tag{2}$$

where σ represents the source density distribution. $\partial\Omega$ is the domain manifold and it represents the discretized surface of the bubble(s). *G* stands for the fundamental solution, taking the form of $1/|x-\xi|$ for the Laplace equation in $\mathbb{R}^3$ space. Taking the derivative of (2) in the x-, y-, and z-direction, respectively, yields:

$$\frac{\partial\phi(\mathrm{x})}{\partial\vec{x_1}} = -\alpha\pi\sigma(\mathrm{x}) + \int_{\partial\Omega} \sigma(\xi) \cdot \vec{x_\iota} \cdot \nabla G(\mathrm{x}, \xi) \cdot \mathrm{dS}(\xi), \tag{3}$$

where α is a parameter in the range of [-4 4], depending on the solid angle of the local surface around x. By solving (2) and (3), one obtains the velocity on the bubble surface. The change of velocity potential on the bubble surface can be described by the Bernoulli equation:

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = -\frac{1}{2}|\nabla\phi|^2 + \frac{1}{\rho_L}(p_b - p_a - p_\infty - \kappa s), \tag{4}$$

where ρ_L is the liquid density. p_b, p_a and p_∞ are the pressure at the bubble interior, the standard atmospheric pressure and the acoustic pressure, respectively. κ represents the sum of the principal curvatures on the bubble surface and *s* denotes surface tension of the liquid.

3 Adaptive cross approximation

The kernel G(x, ξ) is asymptotically smooth [7] as a function of ξ if

$$\left|\partial_\xi^m G(x,\xi)\right| \le C_m \cdot |x-\xi|^{g-m}, \tag{5}$$

where C_m depends on the order of derivative m, whilst g is independent of m. The kernel for Laplace equation, G(x, ξ) =1/|x-ξ|, satisfies the above equation when the point x is far from ξ. Asymptotically smooth functions can be decomposed and expressed as:

$$G(x,\xi) = \sum_{j=1}^{m} f_j(x) \cdot g_j(\xi) + R_m(x,\xi), \tag{6}$$

where the remainder $|R_m(x, \xi)| \to 0$, as m→∞. It is notable that G(x, ξ) can be approximated by the product of two vectors of r_m dimensions. Let $X:=\{x_j: j=1,2,\ldots,m\}$ and $\Xi:=\{\xi_j: j=1,2,\ldots,n\}$ and $H_{\text{convex}}(X)$ and $H_{\text{convex}}(\Xi)$ be the convex hulls of the point sets X and Ξ. The induced matrix is decomposed into multiplication of two r_m-rank matrices. This induces the fast algorithm ACA [8]:

```
Let $\mathcal{M}$ be an m-by-n matrix block to be approximated and S and R be of the same size as $\mathcal{M}$. $S_0=0$, $i_1=1$, and for $k=0,1,2,\ldots,r$, compute

    (1) $E_{i_{k+1}}^T R_k = E_{i_{k+1}}^T \mathcal{M} - \sum_{l=1}^{k} (U_l)_{i_{k+1}} V_l^T$,
    (2) $\left|(R_k)_{i_{k+1},j_{k+1}}\right| = max_j |(R_k)_{i_{k+1,j}}| \neq 0 \to j_{k+1}$,
    (3) $V_{k+1} = E_{i_{k+1}}^T R_k / (R_k)_{i_{k+1},j_{k+1}}$,
    (4) $U_{k+1} = \mathcal{M} E_{j_{k+1}} - \sum_{l=1}^{k} (V_l)_{j_{k+1}} U_l$,
    (5) $\left|(U_{k+1})_{i_{k+2}}\right| = max_{i \neq i_{k+1}} |(U_{k+1})_i| \to i_{k+2}$,
    (6) $S_{k+1} = S_k + U_{k+1} V_{k+1}^T$.
```

The fast algorithm applied to $\mathcal{M}_{m\times n}$ requires computation operations of order $O((m+n)\cdot r^2)$ and memory usage of order $O((m+n)\cdot r)$. Here r depends on the prescribed ε. As a stopping criterion the following algorithm is introduced:

$$||U_k||_F \cdot ||V_k||_F \le \varepsilon_a ||S_k||_F, \tag{7}$$

where $||\cdot||_F$ denotes the Frobenius norm, and

$$||S_k||_F^2 = ||S_{k-1}||_F^2 + 2\sum_{i=1}^{k-1}(U_i, U_k)(V_i, V_k) + ||U_{k-1}||_F^2 \cdot ||V_{k-1}||_F^2. \qquad (8)$$

To apply the algorithm, the condition of admissibility should comply with (6) and it has the following form:

$$\mathrm{Diam}\{\, \mathrm{H}_{\mathrm{convex}}(\Xi)\} \leq \eta \cdot \mathrm{Dist}\{H_{\mathrm{convex}}(\mathrm{X}), H_{\mathrm{convex}}(\Xi)\} \qquad (9)$$

where η is a parameter with $0<\eta<1$ for the oscillatory kernel $G(x, \xi) = 1/|x-\xi|$.

4 Tree structure and matrix partitioning

Although it does not involve approximation of the kernel function, the fast ACA algorithm is restricted to the admissible condition in (9). In order to compress the induced matrix by ACA, a hierarchical tree structure is built up to distinguish blocks that satisfy the prescribed condition. The basic idea is to enclose the whole mesh with a root cell (at Level 0). The subdivision is carried out until each node of the mesh occupies a unique cell. An example of the tree structure is shown in Figure 1. The degree of darkness in (a) implies the level of the cell.

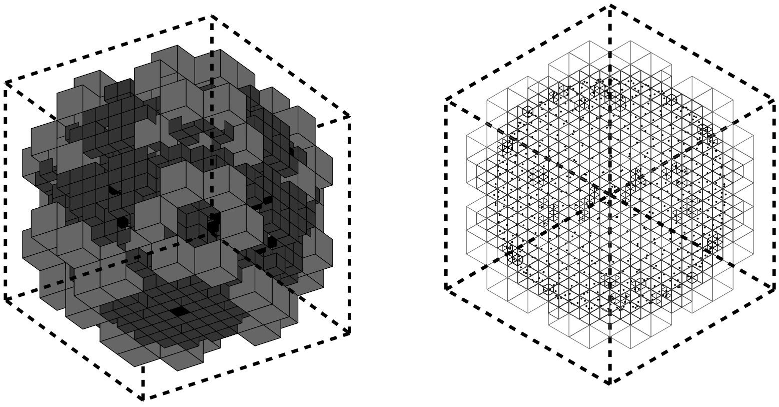

(a) (b)

Figure 1: The tree structure for a mesh of 642 nodes representing a sphere.

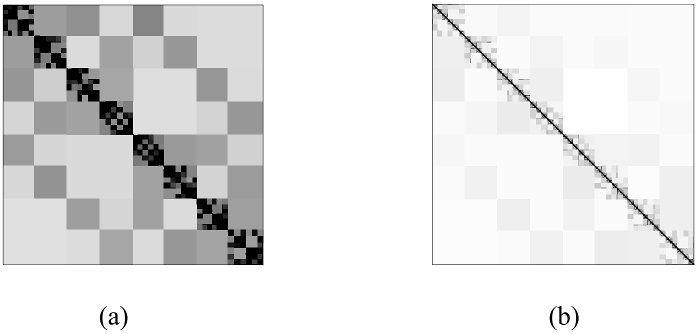
In $\mathbb{R}^3$ space, a cell is subdivided into eight uniform sub-cells, labelled from 0 to 7. These labels can be represented with three binary digits, i.e. 000-111. Consequently, a key system is introduced to represent the oct-tree structure in the $\mathbb{R}^3$ space. Details on the efficient representations can be found in [9]. A complete key is made up of 64 bits, which corresponds to the data type INTEGER(8) in FORTRAN. Each node possesses a unique key. The decimal number of the key indicates the position of the node in the collocation matrix. The row index is identified as a source point and the column index as a collocation point.

Meanwhile, the binary form of the key can be interpreted as the location of the node in the tree. The relative locations between nodes can be obtained as well. Hence, one can now determine the admissibility of the blocks that contain the sets of continuous source points X and continuous collocation points Ξ by using the following steps: (I) determine cells Ψ_s and Ψ_c that cover X and Ξ, respectively; (II) evaluate $H_{\text{convex}}(\Xi)$ by the size of Ψ_c; (III) evaluate the distance between $H_{\text{convex}}(X)$ and $H_{\text{convex}}(\Xi)$ with relative locations between Ψ_s and Ψ_c; (IV) determine whether admissibility is satisfied by using (9) (the block for corresponding to X and Ξ is termed as $B_{X,\Xi}$).

With the aid of the tree represented by the key system, a recursive procedure named BLOCKS($\{\Psi_s\}$, $\{\Psi_c\}$, $\mathcal{B}$) can be carried out as follows to partition the matrix:

```
Let the cell sets $\{\Psi_s\}_l$ and $\{\Psi_c\}_l$ be at the same level l and the top level is denoted as $l_t$, then call BLOCKS($\{\Psi_s\}_l$, $\{\Psi_c\}_l$, $\mathcal{B}$):
For each $\Psi_s \in \{\Psi_s\}_l$ and $\Psi_c \in \{\Psi_c\}_l$,
IF (X in $\Psi_s$ or Ξ in $\Psi_c$ contains only one node) THEN
        $\mathcal{B}=\mathcal{B} \cup B_{X,\Xi}$
ELSEIF (admissible) THEN
        $\mathcal{B}=\mathcal{B} \cup B_{X,\Xi}$
ELSE
        $\{\Psi_s\}_{l+1}$= son(s) of $\Psi_s$, $\{\Psi_c\}_{l+1}$=son(s) of $\Psi_c$;
        CALL BLOCKS($\{\Psi_s\}_{l+1}$, $\{\Psi_c\}_{l+1}$, $\mathcal{B}$):
ENDIF
```

Set $\mathcal{B} = \emptyset$ and call BLOCKS($\{\Psi_s\}_0$, $\{\Psi_c\}_0$, $\mathcal{B}$), and the whole matrix is partitioned into a certain number of blocks, which are stored in $\mathcal{B}$. Figure 2 presents two examples of partitioning the N×N matrix. The degree of whiteness stands for the efficiency of the ACA algorithm in the block.



(a)

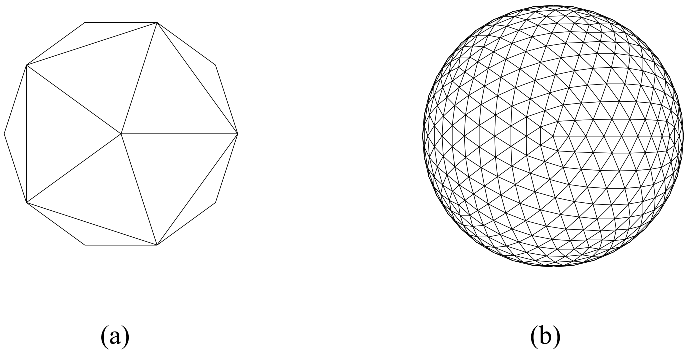
(b)

Figure 2: Matrix partitioning: (a) N=2562; N=10242.

5 Verification and analysis

The ACA-BEM model is developed based on the abovementioned theories. It has been pointed out in [8] that the overall complexity of the ACA method is $O(N^{1+\alpha}\varepsilon^{\alpha})$, with $\alpha>0$ arbitrarily small. The ACA-BEM model is verified by applying it to the solution of bubble dynamics. A case study is analysed with three bubbles in line.

The bubbles are assumed initially to be spherical. An idea presented in [10] is used to approximate a sphere by a refined icosahedron. Each equilateral triangle on the icosahedron is divided into a number of triangles and then they are projected to the surface of the sphere. Hence, the sphere can be represented by a mesh of quasi-lateral triangles. The numbers of nodes and elements are $10\cdot D^2+2$ and $20\cdot D^2$, respectively (D is a positive integer, termed as the degree of refinement). An example of approximating a spherical bubble is presented in Figure 3.



(a) (b)

Figure 3: A sphere approximated by an icosahedron: (a) D=1; (b) D=8.

5.1 Verification of the ACA-BEM model

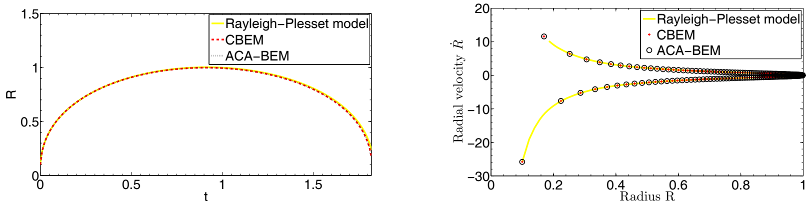
The Rayleigh-Plesset equation was developed by Rayleigh [11] and Plesset [12]. It has been applied for analytical solution of bubble cavitation and spherical bubble oscillation [13, 14]. The simplified Rayleigh-Plesset equation takes the form:

$$R\ddot{R}+\frac{3}{2}\dot{R}^2=\frac{1}{\rho_L}(p_b-p_\infty), \tag{10}$$

where R is the radius of the spherical bubble. Starting from an initial bubble radius R_0 with the prescribed initial condition [15],

$$\phi_0=R_0\left\{\frac{2}{3}\left(\frac{p_\infty-p_b}{\rho_L}\right)\left[\left(\frac{R_m}{R_0}\right)^3-1\right]\right\}^{\frac{1}{2}}, \tag{11}$$

the bubble expands to the maximum size R_m, and then undergoes collapse. p_b-p_∞ equals the standard atmospheric pressure p_a. The dimensionless results are shown in Figure 4. Figure 4(a) presents variation of the bubble radius, increasing from 0.1 to the maximum size 1.0 and subsequently decreasing to around 0.34 at t=1.80. The three curves come from the Rayleigh-Plesset model, the classic BEM (CBEM) model and the ACA-BEM model, respectively. It demonstrates a good agreement between the three models. Particularly, curves from CBEM and ACA-BEM are completely overlapped, which indicates that the accuracy of the fast algorithm is satisfactory. The radial velocity $\dot{R}$ of the bubble in response to the radius R is depicted in Figure 4(b). The negative value of $\dot{R}$ implies bubble expansion and the positive corresponds to bubble collapse. Both the CBEM and ACA-BEM results lie close to the curve representing the analytical solution.



(a) (b)

Figure 4: Bubble expansion and collapse: (a) radius variation in time; (b) radial velocity versus radius.

5.2 Efficiency of the ACA-BEM model

The efficiency of ACA-BEM model is discussed by comparison with the CBEM model. Meshes with different degrees of refinement (D=2, 3, …, 32) are introduced to simulate the evolution of a single spherical bubble. Consequently, the degree of freedom ranges from 42 to 10242 by $10\cdot D^2+2$. Simulations are carried out on a workstation with Intel Xeon W5580 @3.20 GHz processor and 24.0 GB RAM.

The relationship between the degree of freedom N and the required memory is shown in Figure 5. Both the CBEM and ACA-BEM curves grow as N increases. The slope of the CBEM curve is increasing exponentially, whilst the other remains linear. As a result, the gap between the two curves increases. This shows that the memory usage is linearly related to the degree of freedom in the ACA-BEM model. The results show that ACA-BEM provides reduction in required memory by over 90% compared to CBEM, when $N \approx 10^4$.

The computational time is divided into two parts: time to construct the algebraic system of equations and time to solve the system. Figure 6 presents the computational cost versus the degree of freedom N. As N increases, both models require more CPU time for system construction. The curves for CBEM and ACA-BEM in Figure 6(a) demonstrate a distinctive gap for large N, which is

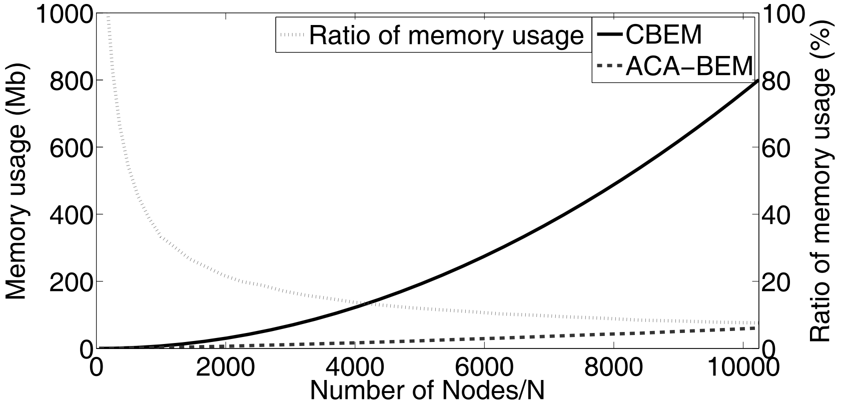


Figure 5: Memory usage versus degree of freedom.

similar as that for memory usage. The curve for the CPU time ratio shows the ACA-BEM model spends less than 10 per cent of the time required by the CBEM model when N≈10^4. In case of small N, the ACA-BEM model might require more CPU time for system construction (up to 150 per cent of the time with the CBEM model). This is due to the fact that the ACA-BEM model spends extra time building up the tree structure and partitioning the matrix.

In the CBEM model, a direct solver based on LU decomposition is employed to solve the asymmetric and fully populated system. This indicates computation operations of O(N^3). By contrast, the iterative GMRes solver is applied to solve the compressed system in the ACA-BEM model. Results are presented in Figure 6(b). The curve for the ratio of CPU time between the ACA-BEM and CBEM models decreases as N increases, although a slight fluctuation can be observed due to the unpredictable iteration number of the GMRes solver. Less than 2 per cent of the CPU time required by the CBEM is needed to solve the system in the ACA-BEM when N≈10^4.

One can find that the CPU time required for system construction is higher than that for solving the system, by comparing the time scales of Figure 6(a) and (b).

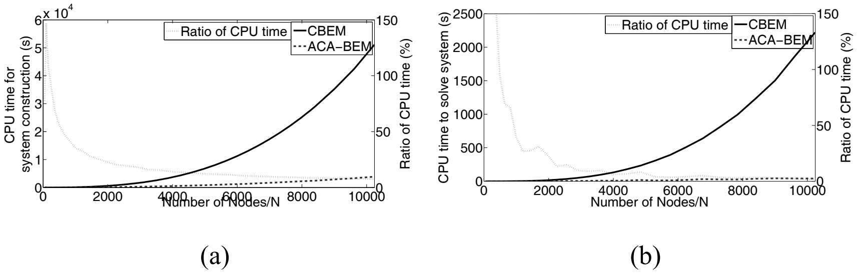


Figure 6: Computational time versus degree of freedom: (a) CPU time for system construction; (b) CPU time for solving the system.

5.3 Case study

The ACA-BEM model is applied to study three bubbles in line. The bubbles are located in large liquid medium of ethanol, assumed to be in an initially equilibrium state. Acoustic forcing is introduced as follows:

$$p_\infty = \bar{p} + p_{\mathrm{amp}} \cdot \mathrm{Re}\left\{e^{i(\omega t - \overrightarrow{\vec{k}\cdot\Delta x})}\right\}, \tag{12}$$

where ω and *k* is the angular frequency and wave number of the regular wave, p_{amp} is the amplitude of the wave pressure and $\bar{p}$ is constant. The bubble in the middle has an initial size R_m=10 μm with a distance from neighbouring bubbles d=10 μm. Two cases are examined, see Figure 7, with the initial size of the neighbouring bubbles $R_n=R_m$ and $R_n=2R_m$, respectively.

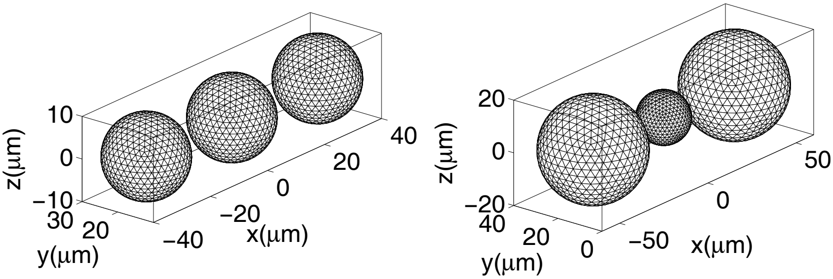
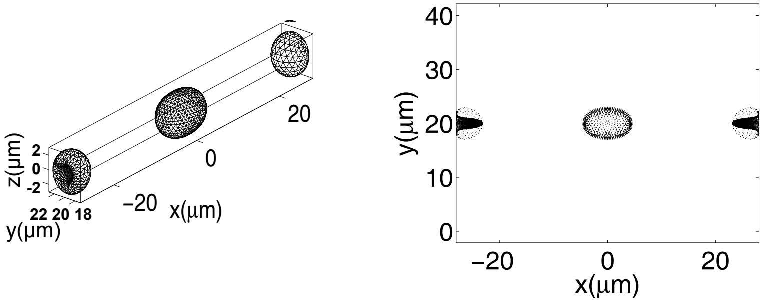


Figure 7: Three bubbles in line: (a) $R_n=R_m$; (b) $R_n=2R_m$.

The last stage of bubble collapse for the case of $R_n=R_m$ is shown in Figure 8. One can see that neighbouring bubbles on both sides collapse with jets towards the bubble in the middle. Due to a symmetric drag of neighbours, the bubble in the middle displays an 'egg-like' shape. The maximum speed is observed at the tip of jets with 366.5 $m\cdot s^{-1}$.



(a) (b)

Figure 8: The last stage of bubble collapse ($R_n=R_m$): (a) geometry; (b) velocity distribution.

Results for the case of $R_n=2R_m$ are presented in Figure 9. Neighbouring bubbles maintain a roughly spherical shape, though a jet formation can be recognized from the velocity distribution. Due to the symmetric drag of the big neighbouring bubbles, the small bubble in the centre shrinks from the middle, and finally displays a 'peanut- like' shape. Presumably, the target will split into two smaller bubbles during later stages. A maximum speed of 63.6 $m\cdot s^{-1}$ is observed in the concave area of the small bubble in the middle.

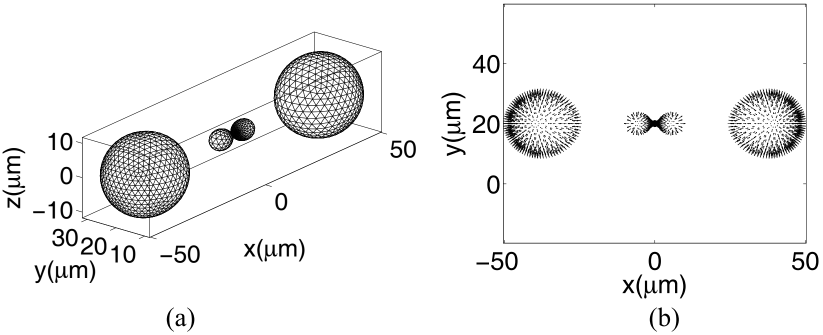


Figure 9: The last stage of bubble collapse ($R_n=2R_m$): (a) geometry; (b) velocity distribution.

6 Conclusions

The ACA accelerated BEM technique has been developed for investigating bubble dynamics. It demonstrates that both numerical operations and memory usage are of linear complexity. The developed model is verified by using analytical solutions and the CBEM model. A case of three bubbles in line is analysed as an application of the ACA-BEM model to bubble dynamics.

Acknowledgement

The present study was supported by the SONO project, contract number: 228730, as part of the Seventh Framework Programme (FP7-NMP-2008-Large-2).

References

[1] J. R. Blake, M. C. Hooton, P. B. Robinson, and Tong R. P., "Collpasing cavities, toroidal bubbles and jet impact," *Phil. Trans. R. Soc. London*, vol. 355, no. 1724, pp. 537-550, 1997.

[2] G.L. Chahine, "Strong interactions bubble/bubble and bubble/flow.," in *IUTAM conference on bubble dynamics and interfacial phenomena (ed. J. R. Blake),* Kluwer, 1994.

[3] B.C. Khoo, D. Adikhari, S.W. Fong, and E. Klaseboer, "Multiple Spark-Generated Bubble Interactions," *Modern Physics Letters B*, vol. 23, no. 3, pp. 229-232, 2009.

[4] I. Benedetti, M.H. Aliabadi, and G. Davi, "A fast 3D dual boundary element method based on hierarchical matrices," *International Journal of Solids and Structures*, vol. 45, pp. 2355–2376, 2008.

[5] 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," *Journal of Computational Physics*, vol. 192, no. 1, pp. 244–261, 2003.

[6] T. T. Bui, E. T. Ong, B. C. Khoo, E. Klaseboer, and K. C. Hung, "A fast algorithm for modeling multiple bubble dynamics," *Journal of Computational Physics*, vol. 216, pp. 430–453, 2006.

[7] A. Brandt, "Multilevel computations of integral transformtions and particle interactions with oscillatory kernels," *Computer Physics Communications*, vol. 65, pp. 24-38, 1991.

[8] M. Bebendorf, "Approximation of boundary element matrices," *Numer. Math.*, vol. 86, pp. 565-589, 2000.

[9] M.S. Warren and J.K. Salmon, "A Parallel Hashed Oct-Tree N-Body Algorithm," in *Supercomputing '93 Proceedings of the 1993 ACM/IEEE conference on Supercomputing*, New York, 1993.

[10] Q. X. Wang, "The evolution of a gas bubble near an inclined wall," *Theoret. Comput. Fluid Dynamics*, vol. 12, pp. 29–51, 1998.

[11] L. Rayleigh, "On the pressure developed in a liquid during the collapse of a spherical void," *Philos. Mag.*, vol. 34, pp. 94-98, 1917.

[12] M. S. Plesset, "The dynamics of cavitation bubbles," *ASME J. Appl. Mech.*, vol. 16, pp. 228-231, 1949.

[13] M. S. Plesset and A. Prosperetti, "Bubble Dynamics and cavitation," *Ann. Rev. Fluid Mech.*, vol. 9, pp. 145-185, 1977.

[14] Z. C. Feng and L. G. Leal, "Nonlinear bubble dynamics," *Annu. Rev. Fluid. Mech.*, vol. 29, pp. 201-243, 1997.

[15] J. R. Blake and D. C. Gibson, "Growth and collapse of a vapour cavity near free surface," *J. Fluid Mech.*, vol. 111, pp. 124-140, 1981.