Acceleration of Gas Bubble-Free Surface Interaction Computation Using Basis Preconditioners

K. L. Tan, B. C. Khoo and J. K. White

ABSTRACT

The computation of gas bubble-free surface interaction entails a time-stepping algorithm whereby a linear system is solved at each time-iteration. In our investigation, the linear systems are derived from a desingularized boundary integral formulation and are poorly conditioned. This leads to poor convergence rates when Krylov subspace methods are used to solve these systems. The convergence rates may however be improved with proper preconditioning.

We limit our investigation to gas bubbles initiated at depths sufficiently small such that a spike forms on the free surface during the later stages of evolution. Bubble dynamics dictate that for gas bubbles initiated at such depths, the stages through which the gas bubble and free surface evolve are similar. Based on this fact, we propose to perform one computation run for a gas bubble initiated at one particular depth, obtain a judicious set of *a priori* basis preconditioners from this run and thereafter, use this set of preconditioners on computation runs for gas bubble initiated at different depths.

The computation time taken by the proposed method is, in general, 50% and 20% of the time taken by the present method (without preconditioning) with terminating criteria of 1.0e-5 and 1.0e-7 in the infinity-norm respectively using the Bi-conjugate Gradient Stabilized solver. The present method further enables computation to an infinity-norm terminating criterion of 1.0e-10 in a shorter time compared to the present method with a criterion of 1.0e-5.

K. L. Tan is a student with the Singapore-MIT Alliance (SMA), National University of Singapore, 10 Kent Ridge Cresent, Singapore 119260. E-mail: smap9055@nus.edu.sg. B. C. Khoo and J. K. White are SMA fellows.

INTRODUCTION

Bubbles surrounded by a liquid can be found in many applications. Examples include bubble jet printers, boiling water and underwater explosions. The chemical industry uses bubble columns to enhance chemical reactions and mixing.

The study of bubbles can be divided into two main areas. The first area, bubbles that do not undergo much change in volume, is studied extensively. A seemingly simple problem of this kind, the terminal rise velocity of a single bubble in water, is still not yet fully understood. The second area entails bubble dynamics pertaining to oscillating bubbles. Examples of these include cavitation bubbles and underwater explosion bubbles.

Underwater explosion bubbles is an active area of defense research. An explosion or gas bubble is formed when an underwater explosion takes place. Due to inertia, this bubble will overexpand and thereafter collapse. If this collapse takes place near a solid surface, a high-speed jet will be formed, directed towards the solid surface. On the other hand, if it takes place near a free water surface, the high-speed jet will then be directed away from the free surface and a water plume or spike will be formed on the surface. All these problems have been studied using the boundary integral method. This method necessitates the meshing of the surface only and not the entire domain thereby greatly reducing the size of matrices to be solved.

In this paper, we seek to reduce the computation time taken to evolve a single gas bubble interacting with a free surface through use of preconditioning. Limiting our investigation to gas bubbles initiated at depths such that a spike forms on the free surface, the preconditioners are

chosen and computed based on some *a priori* knowledge of evolution of such bubbles initiated at different depths.

MATHEMATICAL FORMULATION

Mathematical Formulation

Assuming the fluid domain bounded by the gas bubble and free surface to be inviscid, incompressible and irrotational, the flow within the domain Ω may then be described by a potential field $\phi(x,y,z)$, where the fluid velocity is given by $\nabla\phi$, that satisfies Laplace's equation

$$\nabla^2\phi = 0 .$$

The kinematic and dynamic boundary conditions governing the motion of the gas bubble with no buoyancy force, and free surface are

$$\frac{\mathrm{D}\mathbf{p}}{\mathrm{D}t} = \nabla\phi ,$$

$$\frac{\mathrm{D}\phi}{\mathrm{D}t} = 1 + \frac{1}{2}\left|\nabla\phi\right|^2 - \varepsilon\left(\frac{V_0}{V}\right)^{\lambda} \qquad \text{for } \partial\Omega \in \mathrm{S},$$

and

$$\frac{\mathrm{D}\phi}{\mathrm{D}t} = 1 + \frac{1}{2}\left|\nabla\phi\right|^2 \qquad \text{for } \partial\Omega \in \Sigma$$

where $\mathbf{p} = (x,y,z)$ are the spatial coordinates of the field point in 3-D, t is the dimensionless time variable, γ the dimensionless depth of initiation, ε the dimensionless strength parameter, V the volume of the gas bubble, V_0 the initial volume of the gas bubble, λ the ratio of specific heats, $\partial\Omega$ the domain boundary, S the portion of $\partial\Omega$ defined by the gas bubble and Σ the portion of $\partial\Omega$ defined by the free surface (Best [1] and Wang *et al.* [2]).

The far field boundary condition is prescribed as

$$\phi(\mathbf{p}) \to 0 \text{ as } |\mathbf{p}| = \sqrt{x^2 + y^2 + z^2} \to \infty .$$

The value of parameters used in our investigation is $\varepsilon = 100$, for explosion bubble, and $\gamma = 1.4$, for diatomic gas (Best [1]).

Time Discretization

The Euler Forward time discretization scheme is used. The time discretized equations would thus be of the form

$$\mathbf{p}^{k+1} = \mathbf{p}^k + \nabla\phi^k \cdot \Delta t$$

for the update of the position of the field points, and

$$\phi^{k+1} = \phi^k + \left[1 + \frac{1}{2}\left|\nabla\phi^k\right|^2 - \varepsilon\left(\frac{V_0}{V^k}\right)^{\lambda}\right]\cdot\Delta t \qquad \text{for } \partial\Omega \in \mathrm{S},$$

and

$$\phi^{k+1} = \phi^k + \left[1 + \frac{1}{2}\left|\nabla\phi^k\right|^2\right]\cdot\Delta t \qquad \text{for } \partial\Omega \in \Sigma$$

for the update of potential at the field points. For simplification, we shall write the above two equations as

$$\phi^{k+1} = \phi^k + F\left(\phi^k, V^k\right)\cdot\Delta t$$

with F representing the appropriate function for $\partial\Omega \in \mathrm{S}$ or $\partial\Omega \in \Sigma$.

$\nabla\phi^k$ is updated by solving Laplace's equation with boundary conditions given by ϕ^k. In our code, $\nabla\phi$ is computed directly through the integral equation by differentiating the Greens function in the three principal directions. An alternative method for computing $\nabla\phi$ would be to obtain the normal component of $\nabla\phi$ using the integral equation and the tangential components using finite differences (Wang *et al.* [2]).

Solution of Laplace's Equation

The Desingularised Indirect Boundary Integral Method (DIBIM) is used to solve Laplace's equation (Zhang *et al.* [3]) in order to derive the updates to $\nabla\phi$. Based on the above method, we have

$$\phi(\mathbf{p}) = \sum_i \frac{\sigma(\mathbf{q}_i)}{|\mathbf{p} - \mathbf{q}_i|}$$

where σ is the unknown source strength, $\mathbf{q}_i = (x,y,z)$ are the spatial coordinates of the source point, and $|\mathbf{p} - \mathbf{q}_i|$ is the

Euclidean distance between the field point and the source point.

Initial Conditions

The initial bubble radius for the chosen value of ε is 0.1651, obtained by solving the Rayleigh equation modified for gas bubble (Best [1]). Assuming an initial quiescent free surface and noting the initial velocity of the gas bubble surface to be negligible, the initial condition of $\phi = 0$ may be prescribed for the boundary.

DISCRETIZATION

The initial geometry of the gas bubble is a discretized sphere of 492 nodes with radius 0.1651 located at $(x,y,z) = (0,0,\gamma)$. The initial quiescent free surface is discretized as a flat rectangular surface comprising 729 nodes spanning $(x,y,z) = (-5{:}5,-5{:}5,0)$. The application of a truncated discretized free surface mesh is an approximation made possible by the far field boundary condition prescribed above.

A point collation scheme is used with the DIBIM described above. The collocation is performed on the nodes, 492 for the gas bubble and 729 for the free surface, defined by the discretization. In order to have the number of unknowns equal to the number of known equations, we set the number of source points to be equal to the number of discretized nodes, with 492 points associated with the gas bubble and 729 points associated with the free surface. This results in a system with N = 1221 degrees-of-freedom. The source points are positioned in accordance with Zhang *et al*. [3].

As mentioned above, the Euler Forward time-stepping scheme is used to evolve the gas bubble and free surface. The time-step is chosen such that the bound on the maximum change in potential $\Delta\phi$ for any node is restricted to 0.04 at each time-iteration,

$$\Delta t = \min\left(\frac{\Delta\phi}{\max\limits_i \left[1 + \frac{1}{2}\left|\nabla\phi\right|^2 + \varepsilon\left(\frac{V_0}{V}\right)^{\lambda} \right]}, \frac{\Delta\phi}{\max\limits_j \left[1 + \frac{1}{2}\left|\nabla\phi\right|^2 \right]} \right)$$

The value of 0.04 is typical of the range of $\Delta\phi$ chosen for such computations (Best [1] and Wang *et al.* [2]). The expected nodal potential values would range from –1 to 3.

PRESENT IMPLEMENTATION

Let $\mathbf{p}_j = (x, y, z)$ be the position of field point j, $\boldsymbol{\varphi} \in \mathbb{R}^N$ be the vector of potentials at the N field points, $\boldsymbol{\sigma} \in \mathbb{R}^N$ be the vector of source strengths at the N source points and K be the number of time-iterations to be taken.

The present implementation is as follows:

Set initial conditions: $\mathbf{p}_j = \mathbf{p}_j^0 \ \forall\, j = 1:N$, $\boldsymbol{\varphi} = \mathbf{0}$ *where* $\mathbf{p}_j^0$ *is the initial position of the field point j.*

Do $k = 1:K$

1. *Form matrix* $\mathbf{G}$ *where* $\mathbf{G}_{ij} = \dfrac{-1}{\left|\mathbf{p}_j - \mathbf{q}_i\right|}$
2. *Solve* $\mathbf{G}\boldsymbol{\sigma} = \boldsymbol{\varphi}$ *for* $\boldsymbol{\sigma}$ *using iterative solver with set terminating criterion*
3. *Compute material derivative* $\nabla\boldsymbol{\varphi} = \mathbf{H}\boldsymbol{\sigma}$

   where $\mathbf{H}_{ij} = \nabla\left(\dfrac{-1}{\left|\mathbf{p}_j - \mathbf{q}_i\right|} \right)$
4. *Compute time-step* Δt
5. *Update position of field points*

   $\mathbf{p}_j^{k+1} = \mathbf{p}_j^k + \left(\nabla\boldsymbol{\varphi}\right)_j \cdot \Delta t \ \forall\, j = 1:N$
6. *Update potential of field points*

   $\boldsymbol{\varphi}_j^{k+1} = \boldsymbol{\varphi}_j^k + F\left(\boldsymbol{\varphi}, z_j, V\right) \cdot \Delta t \ \forall\, j = 1:N$

   where F is the kinematic boundary condition function associated with node j
7. *Compute new volume V*

End

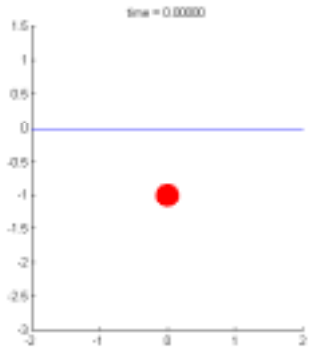
The iterative solver used is the Bi-conjugate Gradient Stabilized (BICGSTAB) solver. The $\mathbf{G}$ matrices generated

based on the DIBIM above have been found to be poorly conditioned. A terminating criterion of

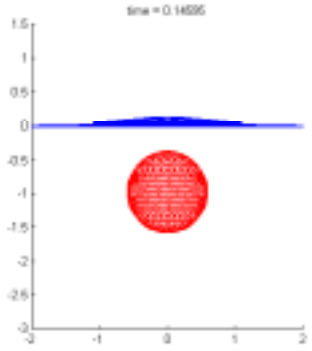
$$\|\mathbf{r}\|_{\infty} < 1.0\text{e} - 5$$

where $\mathbf{r} \in \mathbb{R}^N$ is the vector of residuals, is used. This liberal terminating criterion is chosen because of the poor convergence characteristics associated with using a Krylov Subspace solver on ill-conditioned matrices. The DIBIM is also limited in that it starts to fail as opposite sections of the bubble surface approach each other. In our investigation, we set K = 150 so as to end our computation before this scenario arises.

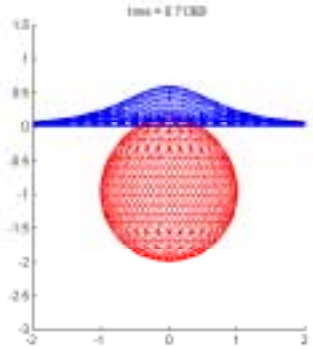
The output based on the above implementation for the evolution of free surface interaction with a single gas bubble initiated at $\gamma = -1.0$ is shown in Figure 1.



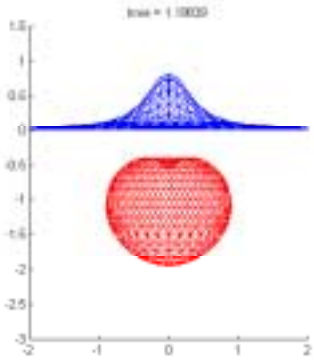
(a)



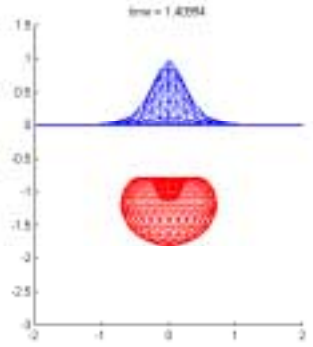
(b)



(c)



(d)



(e)

Figure 1: Evolution of Free Surface Interaction with Gas Bubble Initiated at $\gamma = -1.0$

PROPOSED METHOD

The bound on convergence rate of a Krylov Subspace solver for the solution of a general linear system

$$\mathbf{Ax} = \mathbf{b}$$

is a function of the condition number of the matrix $\mathbf{A}$, $\kappa(\mathbf{A})$. In general, the better the condition of $\mathbf{A}$, the higher is the convergence rate and vice-versa.

Block Diagonal Preconditioning

The use of preconditioning to achieve better convergence properties for iterative solvers is not new. There are also many forms of preconditioners (Saad [5]). In our investigation, we choose to use block diagonal preconditioners. The rationale for such a choice is explained below.

As stated in *Step 1* of the implementation shown before, the entries in the $\mathbf{G}$ matrix are

$$\mathbf{G}_{ij} = \frac{-1}{\left|\mathbf{p}_j - \mathbf{q}_i\right|}.$$

This means that the magnitude of entry $\mathbf{G}_{ij}$ is inversely proportional to the distance between field point *j* and source point *i*.

Let the 492 nodes associated with the gas bubble be indexed $j = 1:492$ and the 792 nodes associated with the free surface be indexed $j = 493:1221$. The resulting $\mathbf{G}$ matrix generated would be of the form

$$\mathbf{G} = \begin{bmatrix} \begin{bmatrix} \mathbf{G1} \end{bmatrix} & \begin{bmatrix} \mathbf{G3} \end{bmatrix} \\ \begin{bmatrix} \mathbf{G4} \end{bmatrix} & \begin{bmatrix} \mathbf{G2} \end{bmatrix} \end{bmatrix}$$

where $\mathbf{G1} \in \mathbb{R}^{492\times492}$ is the sub-matrix associated with the gas bubble nodes only, $\mathbf{G2} \in \mathbb{R}^{729\times729}$ is the sub-matrix associated with the free surface nodes only, and $\mathbf{G3} \in \mathbb{R}^{492\times729}$ and $\mathbf{G4} \in \mathbb{R}^{729\times492}$ are the "cross-coupled terms".

It is observed from Figure 1a that during the initial stages of evolution, the distance between two bubble nodes is much smaller compared to the distance between a bubble node and a free surface node. Because of the positioning of the source points in accordance with Zhang *et al.* [3], it is expected that the value of all entries in matrices **G3** and **G4** to be much smaller in magnitude as compared to the value of the entries in matrices **G1** and **G2**. As the gas bubble evolves, it is observed that only the top section of the bubble surface comes into close proximity with the free surface. This implies that most of the "coupling" occurs within matrices **G1** and **G2**, and not within **G3** and **G4**.

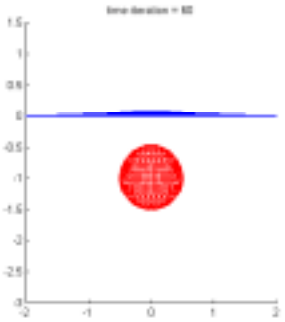
Based on the above observation, if we are able to construct a preconditioner $\mathbf{M}$ for the $\mathbf{G}$ generated at every time-iteration such that

$$\mathbf{M} = \begin{bmatrix} \begin{bmatrix} \mathbf{G1} \end{bmatrix}^{-1} & \begin{bmatrix} \mathbf{0} \end{bmatrix} \\ \begin{bmatrix} \mathbf{0} \end{bmatrix} & \begin{bmatrix} \mathbf{G2} \end{bmatrix}^{-1} \end{bmatrix}$$

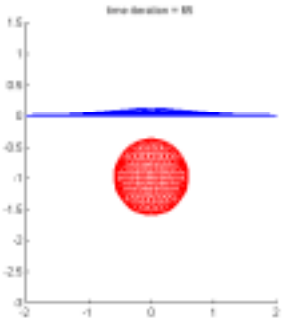
then (right-)preconditioning the matrix $\mathbf{G}$ with $\mathbf{M}$ would yield

$$\mathbf{GM} = \begin{bmatrix} \begin{bmatrix} \mathbf{I} \end{bmatrix} & \begin{bmatrix} \mathbf{G3}\cdot\mathbf{G2}^{-1} \end{bmatrix} \\ \begin{bmatrix} \mathbf{G4}\cdot\mathbf{G1}^{-1} \end{bmatrix} & \begin{bmatrix} \mathbf{I} \end{bmatrix} \end{bmatrix},$$

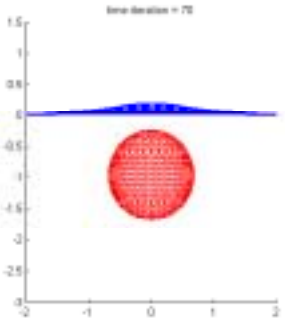
resulting in a better conditioned matrix. It would however be computationally expensive to invert sub-matrices **G1** and **G2** at every time-iteration for use as preconditioners.



(a)



(b)



(c)

Figure 2: Gas Bubble and Free Surface Geometry at Time-iterations 60 (a), 65 (b) and 70 (c) for $\gamma = -1.0$.

The need to compute preconditioners at every time-iteration can be eliminated if we relax the requirement that the block preconditioner sub-matrices be exact inverses of sub-matrices **G1** and **G2**. In fact, our aim is just to precondition the **G** matrix such that it has better convergence properties and not to convert the diagonal sub-matrices into identity sub-matrices. Figures 2a, 2b and 2c show the gas bubble and free surface for $\gamma = -1.0$ at time-iterations 60, 65 and 70 respectively. It is observed from Figure 2 that the change in geometry over several time-iterations is limited. This implies that the change in value of entries in the **G** matrix from one time-iteration to the next is also limited. In this case, we can compute the preconditioner for the **G** matrix generated at a particular time-iteration and make use of this preconditioner to precondition the **G** matrices generated at time-iterations about that particular time-iteration from which the preconditioner is obtained. In this case, what we obtain is

$$\mathbf{M}' = \begin{bmatrix} \begin{bmatrix} \sim \mathbf{G1} \end{bmatrix}^{-1} & \begin{bmatrix} \mathbf{0} \end{bmatrix} \\ \begin{bmatrix} \mathbf{0} \end{bmatrix} & \begin{bmatrix} \sim \mathbf{G2} \end{bmatrix}^{-1} \end{bmatrix}$$

and after (right-)preconditioning

$$\mathbf{GM}' = \begin{bmatrix} \begin{bmatrix} \sim \mathbf{I} \end{bmatrix} & \begin{bmatrix} \mathbf{G3} \cdot \sim \mathbf{G2}^{-1} \end{bmatrix} \\ \begin{bmatrix} \mathbf{G4} \cdot \sim \mathbf{G1}^{-1} \end{bmatrix} & \begin{bmatrix} \sim \mathbf{I} \end{bmatrix} \end{bmatrix}$$

which would still result in a relatively well-conditioned matrix.

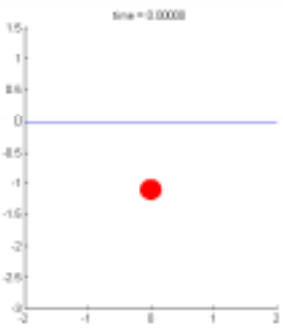
By judiciously choosing a handful of **G** matrices throughout the entire computation run to generate the preconditioners from, effective preconditioning may be achieved for the entire computation run.

The above method may improve the condition of the matrix and therefore reduce the time taken to solve the linear system at each time-iteration. However, this method would require *a priori* knowledge of the **G** matrices so as to compute the preconditioners. This would imply that, in order to accelerate the computation, we would have to perform the said computation first, which is somewhat inefficient. An alternative method is to decide *a priori* at which time-

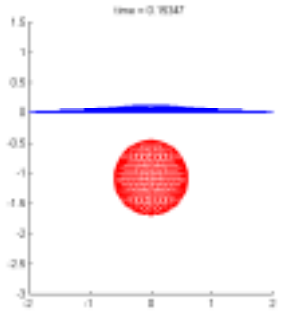
iteration to compute the preconditioners. This would however necessitate the inversion of matrices during the computation run thus making the computation run expensive. In fact, there may not even be any savings in computation time in using this method over the present method.

A Priori Computed Basis Preconditioners

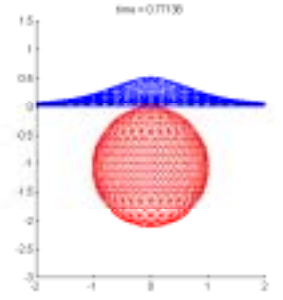
Given in Figure 3 is the evolution of free surface interaction with a single gas bubble initiated at $\gamma = -1.1$. Comparing Figures 3 and 1, it is observed that the gas bubble in both cases, though initiated at different depths, goes through the same sequence of evolution with small differences in geometry.



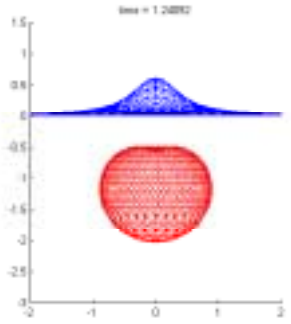
(a)



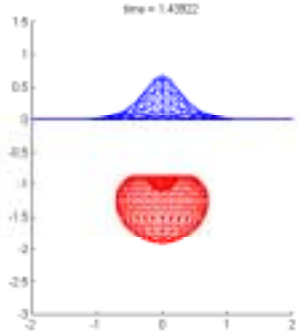
(b)



(c)



(d)



(e)

Figure 3: Evolution of Free Surface Interaction with Gas Bubble Initiated at $\gamma = -1.1$

As mentioned before, our aim of preconditioning is not to generate identity sub-matrices but simply to improve the condition of the $\mathbf{G}$ matrices generated such that they have better convergence properties. It is therefore plausible to make use of preconditioners generated from the computation run of $\gamma = -1.0$ for the computation run of $\gamma = -1.1$. This claim is substantiated by the observation above that for a single gas bubble initiated at various depths such that a spike forms on the free surface, the gas bubble and free surface would go through the same sequence of evolution with marginal differences in geometry

In this case, the need to have *a priori* knowledge of the $\mathbf{G}$ matrices for each new computation run of a different γ is eliminated. In fact, we need only perform one computation run of one value of γ using the present method, compute the preconditioners from the $\mathbf{G}$ matrices obtained at judiciously chosen time-iterations and then make use of these preconditioners for other computation runs of any value of γ that results in spike formation on the free surface.

Proposed Implementation

The proposed implementation is given below.

Perform only once to obtain the basis preconditioners:

Judiciously select the time-iterations to compute the basis preconditioners from. Let the number of basis preconditioners selected be $\tilde{n}$.

For a selected value of $\gamma = \tilde{\gamma}$, *perform computation run using the present method.*

If time-iteration k = one of the $\tilde{n}$ *pre-selected time-iterations above,*

1. *Extract sub-matrices* $\mathbf{G1}$ *and* $\mathbf{G2}$, *and compute* $\mathbf{G1}^{-1}$ *and* $\mathbf{G2}^{-1}$
2. *Form* $\mathbf{M}$
3. *Write* $\mathbf{M}$ *to file*

At the end of the computation run, there should be $\tilde{n}$ *a priori* computed basis preconditioners.

To obtain the solution to any value of γ:

Set initial conditions.

Do $k = 1 : K$

1. *Form matrix* $\mathbf{G}$
2. *Precondition* $\mathbf{G}$ *with the most appropriate* $\mathbf{M}$ *from the set of* $\tilde{n}$ *basis preconditioners*
3. *Solve* $\mathbf{G\sigma} = \boldsymbol{\varphi}$ *for* $\boldsymbol{\sigma}$ *using iterative solver with set terminating criterion*
4. *Compute material derivative* $\nabla\boldsymbol{\varphi} = \mathbf{H\sigma}$
5. *Compute time-step* Δt
6. *Update position of field point*

$$\mathbf{p}_j^{k+1} = \mathbf{p}_j^k + \left(\nabla\boldsymbol{\varphi}\right)_j \cdot \Delta t \;\; \forall \; j = 1 : N$$

7. *Update potential of field points*

$$\boldsymbol{\varphi}_j^{k+1} = \boldsymbol{\varphi}_j^k + F\left(\boldsymbol{\varphi}, V\right) \cdot \Delta t \;\; \forall \; j = 1 : N$$

where F is the kinematic boundary condition function associated with node j

8. *Compute new volume V*

End

In *Step 2* above, the "most appropriate $\mathbf{M}$" would be the preconditioner whose geometry which it is computed from, most closely matches the geometry at time-iteration *k*, which matrix $\mathbf{G}$ is computed from. Right-preconditioning is used since this preserves the residual of the original linear system.

Selection of Preconditioners

In the selection of preconditioners, we would not want to select too large a number since this would degrade the performance of the proposed method through excessive reading of files. On the other hand, we would not want to select too low a number of preconditioners since this would not efficiently precondition the $\mathbf{G}$ matrices and also degrade the performance of the proposed method.

For the computation of the basis preconditioners, we use $\gamma = -1.0$. This value of γ is used because it is approximately the mean of the range of γ values that we are interested in investigating.

The selection criterion we choose is based on the amount of change in geometry within a range of time-iterations, rather than simply a uniform distribution based on the number of time-iterations or geometry of bubble. The rationale for the above criterion is that if the geometry changes minimally over a wide range of time-iterations, there is no need for many preconditioners over that range of time-iterations on the basis of the limited change in geometry. On the other hand, if the change in geometry is tremendous over a limited range of time-iterations, there is no need for many preconditioners over that range of geometry on the basis of the limited number of time-iterations taken to evolve over that range of geometry.

Since the volume of the gas bubble is solely a function of the geometry of the bubble, we can replace "geometry" in the above criterion with "volume of the gas bubble". We can also use the volume to establish the "most appropriate" preconditioner to use. The alternative view is that we can determine the range of volumes that we use a particular preconditioner for.

Based on the above criterion, we look to the volume-iteration plot given in Figure 4 to select the volume, and thus the time-iteration to compute the preconditioners from.

It is observed from Figure 4 that the increase in volume during the first 40 time-iterations is minimal. The rate of increase picks up dramatically at about time-iteration 60, tapering off upon nearing maximum volume. The rate of change in volume during the collapse phase is slightly less than that during the rapid expansion phase. Based on the selection criterion detailed above, the volumes and the corresponding time-iterations we choose to compute the preconditioners for various ranges of volume are given in Table 1.

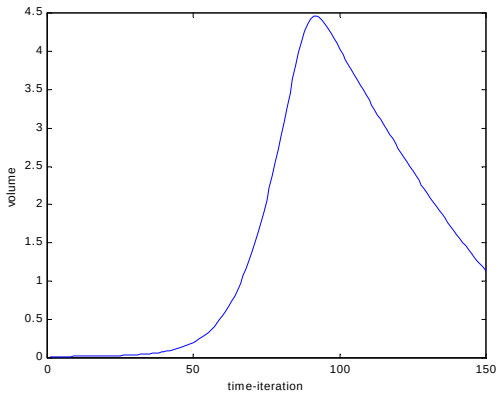


Figure 4: Gas Bubble Volume against Time-iteration Plot for $\gamma = -1.0$.

Desired Volume	Selected Time-iteration	Range of Volume Selected For
~ 0.04	30	Initial – 0.08
~ 0.75	63	0.08 – 1.5
~ 2.25	76	1.5 – 3.0
~ 3.75	85	3.0 – 4.5
~ 4.0	100	4.5 – 3.5
~ 3.0	116	3.5 – 2.5
~ 2.0	133	2.5 – 1.5

Table 1: Volumes Chosen for Computation of Preconditioners

Having determined the range of volume associated with each preconditioner, a new preconditioner is read into memory from file as and when necessary during the computation.

RESULTS AND DISCUSSION

The cpu-time (CPU) and wall-clock (WC) time taken to complete a computation run of 150 time-iterations for various γ's is given below. The iterative solution method used is the BICGSTAB method with the terminating criterion being the infinity norm of the residual as is with the present method. Tables 2, 3 and 4 show the times taken with terminating criteria of 1.0e-5, 1.0e-7 and 1.0e-10 respectively. Table 5 shows a comparison of the total number of iterations taken by the iterative solver during the entire computation run of 150 time-iterations.

It is observed from the results above that with a terminating criterion of 1.0e-5, the proposed method reduces the time taken by almost 50% over the present method. Upon tightening the terminating criterion to 1.0e-7, the proposed method reduces the time taken by almost 80% over the present method. There is no timing recorded for the present method with a terminating criterion of 1e-10. However, an alternative way of interpreting the results would be that with the proposed method, we can achieve an improvement of 5 orders in the infinity norm of residual over the present method while taking less time to complete the computation run.

Further evidence of the positive effect of the proposed preconditioning method on the convergence characteristics of the $\mathbf{G}$ matrices formed may be observed in Table 5, which shows the total number of iterations taken by the iterative solver for an entire computation run of 150 time-iterations at various γ's.

Figure 5 shows the plots of number of iterations taken by the BICGSTAB solver at each time-iteration for $\gamma = -0.8$ using the present method and the proposed method with a terminating criterion of 1.0e-7.

It is observed from Figure 5 that the convergence rate is improved throughout the entire computation run.

	Proposed Method		Present Method		Ratio	
	CPU (s)	**WC (min:s)**	**CPU (s)**	**WC (min:s)**	**CPU**	**WC**
γ= -0.8	252.82	4:12	483.67	8:03	0.523	0.522
γ= -0.9	224.37	3:45	433.31	7:13	0.518	0.520
γ= -1.0	213.28	3:33	411.77	6:51	0.518	0.519
γ= -1.1	207.45	3:27	410.07	6:50	0.506	0.505
γ= -1.2	211.40	3:32	423.31	7:04	0.500	0.500

Table 2: Time taken for 150 Time-iterations with Iterative Solver Terminating Criterion of 1.0e-5

	Proposed Method		Present Method		Ratio	
	CPU (s)	**WC (min:s)**	**CPU (s)**	**WC (min:s)**	**CPU**	**WC**
γ = -0.8	294.12	4:54	1635.52	27:15	0.180	0.180
γ = -0.9	258.43	4:19	1502.76	25:03	0.172	0.173
γ = -1.0	245.30	4:06	1453.27	24:13	0.169	0.170
γ = -1.1	239.97	4:00	1439.76	24:00	0.167	0.167
γ = -1.2	245.85	4:06	1554.17	25:54	0.159	0.159

Table 3: Time taken for 150 Time-iterations with Iterative Solver Terminating Criterion of 1.0e-7

	Proposed Method	
	CPU (s)	**WC (min:s)**
γ= -0.8	368.55	6:08
γ= -0.9	321.92	5:22
γ= -1.0	295.61	4:55
γ= -1.1	295.66	4:56
γ= -1.2	308.84	5:09

Table 4: Time taken for 150 Time-iterations with Iterative Solver Terminating Criterion of 1.0e-10

Tolerance	**1.0e-5**			**1.0e-7**			**1.0e-10**
Method	**Proposed**	**Present**	**Ratio**	**Proposed**	**Present**	**Ratio**	**Proposed**
γ= -0.8	799	3583	0.223	1060	14555	0.073	1536
γ= -0.9	599	3070	0.196	822	13315	0.062	1223
γ= -1.0	524	2870	0.183	738	12729	0.058	1057
γ= -1.1	487	2873	0.170	705	12609	0.056	1060
γ= -1.2	507	2987	0.170	746	13641	0.055	1145

Table 5: Total Number of Iterations taken by Iterative Solver for 150 Time-iterations

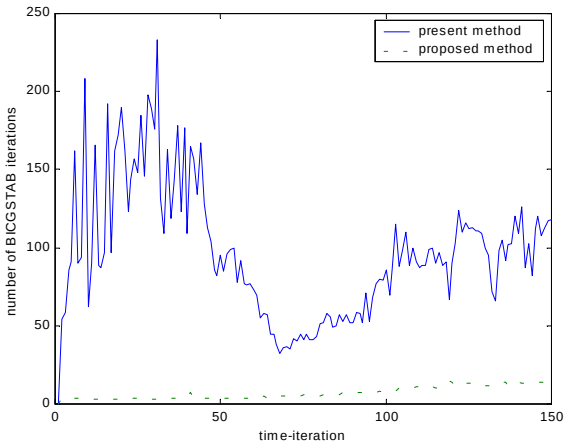


Figure 5: Comparison of Number of Iterations taken by Iterative Solver at each Time-iteration between Present Method and Proposed Method

CONCLUSIONS

Based on some *a priori* knowledge of the evolution of free surface interacting with a single gas bubble initiated at the range of depths under investigation, a set of basis preconditioners is computed from the computation run made for one particular depth. This set of basis preconditioners is then used to precondition the matrices generated at each time-iteration for computation runs made for different depths.

Using this proposed method, the time taken to complete 150 time-iterations of evolution of free surface interacting with a single bubble is 50% and 20% of the time taken by the present method with terminating criteria of 1.0e-5 and 1.0e-7 respectively for the iterative solver used. Furthermore, the proposed method allows computation to a tolerance of 1.0e-10 in a time less than that taken by the present method to compute to a tolerance of 1.0e-5.

REFERENCES

1. Best, J. P., 1991, "The Dynamics of Underwater Explosions", Ph.D. Thesis, The University of Wollongong, Australia.

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

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

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

5. Saad, Y., 1996, "Iterative Methods for Sparse Linear Systems", PWS Publishing Company.