A NUMERICAL SIMULATION OF THE GROWTH AND COLLAPSE OF VAPOUR CAVITY NEAR A FREE SURFACE ON DISTRIBUTED COMPUTING THROUGH PVM

U Periyathamby, Khoo B C, Yeo K S and Wang Q X
Department of Mechanical and Production Engineering
National University of Singapore
Kent Ridge Crescent, Singapore 0511.

ABSTRACT

The effect of surface tension on the geometry of the cavity and free-surface jet is studied in detail. The life-cycle and centroid movement of the cavity plus the maximum height to which the free surface jet rises is found to be strongly dependent on the surface tension. The entire computation, based on the *master/slave* model, were done on a distributed computing environment using the Parallel Virtual Machine (*PVM*) software.

INTRODUCTION

Over the past few decades there has been considerable effort put into the numerical study of growth and collapse of cavities (or bubbles) under various restrictions. The behaviour of a single bubble throughout a single cycle of expansion and collapse is of common interest to all the different types of fields : cavitation, underwater signaling, underwater-weapon design, chemical processing, medicine and nuclear physics. Most of the workers (Lundgren & Mansour (1991); Zhang *et al* (1993), Wang *et al* (1994)) have studied the case of single bubble and very few investigations has been done on multiple bubbles. In this paper we present a new approach for solving (axisymmetric) bubble dynamics problems on the state-of-the-art distributed computing environment: *Parallel Virtual Machine* (PVM). The effects of surface tension is also studied in detail.

MATHEMATICAL FORMULATION

The boundary-integral method has become the principal numerical technique used for solving bubble dynamics problems because of the ease with which it can follow the contortions of the bubble shape, and is adopted here. The flow is assumed to be incompressible, irrotational inviscid and axisymmetric (Figure 1); and satisfies the Laplace equation, $\nabla^2\phi = 0$ where ϕ is the velocity potential. The application of the Green's theorem allows one to write the solution of the Laplace equation in the domain Ω as

$$c(p)\,\phi(p) = \int_{\partial\Omega}\left[\frac{\partial\phi(p,q)}{\partial n}G(p,q) - \phi(q)\frac{\partial G(p,q)}{\partial n}\right]dS \quad (1)$$

$$\text{where } c(p) = \begin{cases}2\pi & p \in \partial\Omega = S \cup \Sigma\\ 4\pi & p \in \Omega\end{cases} \quad (2)$$

The point p is somewhere in the flow domain and $\partial/\partial n = n \bullet \nabla$ is the normal derivative at the boundary. $G(p,q) = |p-q|^{-1}$ is the Green's function due to a unit source in an infinite fluid domain.The dynamic boundary conditions comes from the consideration of the Bernoulli equation and the normalised form is given by

$$\frac{d\phi}{dt} = 1 + \frac{1}{2}|\nabla\phi|^2 + \delta^2(z-\gamma) - \frac{1}{\tau}\left(\frac{1}{R_1} + \frac{1}{R_2}\right) \quad \text{for } p \in S \quad (3a)$$

$$\frac{d\phi}{dt} = \frac{1}{2}|\nabla\phi|^2 + \delta^2 z - \frac{1}{\tau}\left(\frac{1}{R_1} + \frac{1}{R_2}\right) \quad \text{for } p \in \Sigma \quad (3b)$$

where $\delta = \sqrt{\rho g R_m / \Delta p}$ is termed the buoyancy force parameter, $\gamma = h / R_m$ (h being the initial inception depth) and τ is defined as $\tau = \frac{R_m \Delta p}{\sigma}$ to measure the effect of the surface tension. $1/R_1$ and $1/R_2$ are the non-dimensional principal curvatures of the surface and R_m is the maximum radius the bubble would reach in an infinite medium (Rayleigh bubble). In terms of potential, the movement of a material on the boundary follows $\frac{dx}{dt} = \nabla\phi$ where x is defined as the spatial vector coordinates of a material point on the surface.

The surface of the bubble and the free surface are discretised as a set of N linear segments S_j, each of which is locally parameterised by ξ in the range [0,1]. The integral equation (1) can be expressed in discretised form as

$$c\,\phi(r_0, z_0, t) = \sum_{j=1}^{N}\int_0^1 \psi(\xi)K_1(r(\xi), z(\xi), r_0, z, t)\,d\xi - \sum_{j=1}^{N}\int_0^1 \phi(\xi)K_2(r(\xi), z(\xi), r_0, z, t)\,d\xi \quad (5)$$

where $\psi(\xi) = \partial\phi/\partial n$, and K_1, K_2 are the kernel functions in terms of the elliptic integrals of the first and second kinds. derived from the integration of the Green's function G(p,q) and its normal derivative $\partial G/\partial n$ in the azimuthal direction.

DISTRIBUTED COMPUTING : PARALLEL VIRTUAL MACHINE (*PVM*)

Parallel processing can be defined as the method of having many small tasks collectively solving one large problem. Parallel processing is basically a consequence of the demand for higher performance, lower cost and sustained productivity. A distributed computing system is different from a parallel computing system in that the processors in the former are physically far apart and connected by a network. As more and more organizations have high-speed local area networks interconnecting many general-purpose workstations (called a workstation farm), the combined computational resources may exceed the power of a single high-performance computer. A key concept in PVM is that it makes a heterogeneous collection of computers appear as one large virtual machine, hence its name. The PVM system is composed of two parts : first part is a daemon (called *pvmd3* or *pvmd*) that resides on all the computers making up the virtual machine; second part is a *library of PVM interface routines* (in C and Fortran77). The PVM supports either or a mixture of programming paradigms, viz. Single Instruction Multiple Data (SIMD), or Multiple Instruction Multiple Data (MIMD). For good efficiency. the task granularity must be fairly large (each granule is a program component that forms a part of the global computational task). In this paper message-passing model is implemented through a *master/slave* programming style (a sample is given in Appendix A), where the master program, residing on one machine, controls the flow of the computation, farms out the calculations to the slave workers, and collect the results from the farm as they are available. The algorithm performance is usually analysed by means of two measures : *speed-up* and *efficiency* defined by $S_p(n) = \dfrac{T_1(n)}{T_p(n)}$ and $E_p(n) = \dfrac{S_p(n)}{p}$ respectively, where p is the number of processors, n is the problem size and T is the total wall time taken.

RESULTS AND DISCUSSION

Surface Tension : It has been shown, both experimentally as well as analytically, that an initially spherical bubble will evolve towards a toroidal shape because of buoyancy forces. As the bubble begins to rise, the vortex sheet which develops at the surface has a sense of rotation which induces motion of a tongue of liquid (penetration jet). The direction and the point on the surface at which this penetration starts to develop strongly depends on its initial distance from the free-surface. When the penetration jet impinges on the opposite surface, the topology changes completely. The final topology also depends quite strongly on the strength of surface tension. The physics of the process beyond the contact is very complex and our present code fails. The contact may be made at a single or multiple points, depending strongly on the two factors mentioned above.

As compared to the case of without surface tension, the basic effect of adding the surface tension will be its strong tendency to eliminate any sharp edges in the geometry of the free-surface and bubble-surface. Consequently, the jet broadens with increasing surface tension value (Figure 2) and gives a more realistic simulation as compared to zero surface tension case. With the addition of surface tension, as shown in Figure 3a, the bubble has a shorter life cycle and the maximum size of the bubble reduces. When the surface tension is increased the bubble rises to a lesser height towards the free-surface and moves away at a much higher rate (Figure 3b). Based on the Kelvin-impulse theory, The direction of the jet and the bubble migration (centroid motion), depends on a particular value of $\gamma\delta$ (see Wang *et al)*. The maximum height to which the free-surface jet shoots from the datum also reduces with increasing surface tension. Similar to the bubble penetration jet, the free-surface jet broadens with increasing surface tension. A typical situation of this is depicted in Figure 4 for two different values of surface tension, keeping all the other parameters same. The surface tension is seen to pull the free-surface jet inwards near its uppermost point producing a ring geometry (necking effect) with high curvature which will eventually result in the breaking off of a drop . The other effect of surface tension is also seen by the concurrent formation of another ring geometry with high curvature at the 'corner' where the horizontal free-surface slowly transforms into the vertical jet. The pathline studies of the fluid particles (Cerone & Blake (1984); Taib (1985)) has showed that, as the initially horizontal free-surface transforms into a vertical jet, the fluid is drawn from the region of low curvature and accelerated into region of high curvature (ring) and then upwards into the vertical jet. This phenomena occurs at such high speed and Boulton-Stone & Blake (1993) proposed that such flows are potentially lethal to the cells in a bioreactor, which may be stretched and ruptured by the high strain rates.

Computational Aspects : The timings (wall time) for a fixed number of cycles is compared with increasing number of processors. The speed-up, defined by the ratio of the wall time for one processor to *p* processors, and the efficiency of the algorithm is displayed in Table 1.

p	Wall Time	S_p	E_p
1	2500	1.00	1.00
3	1057	2.365	0.788
5	899	2.781	0.556
8	704	3.551	0.444

Table 1 : The efficiency and speed-up with different number of processors. Wall time is in seconds.

The problem is solved on a farm of DEC-Alpha 3000 workstations using the master-slave algorithm for best load-balancing. The efficiency is good for low number of machines and somewhat deviate very much from perfect scaling for higher number, simply due to Amdahl's law which places an upper bound on the available speed-up; in other words, major part of the source-code computes serially (inherent). Nevertheless, the speed-up show that the resources (the available computer power or facility) are utilised more efficiently and shorter computing time is required.

REFERENCES

Boulton-Stone, J.M., and Blake, J.R., 1993," Gas Bubbles Bursting at a Free Surface", *Journal of Fluid Mechanics*, , v.254, pp.437-466.

Cerone, P., and Blake, J.R., 1984,"A Note On The Instantaneous Streamlines, Pathlines and Pressure Contours for a Cavitaion Bubble Near a Boundary", *J.Austral. Math.* B26, , pp.31-44.

Lundgren, T.S. and Mansour, N.N., 1991,"Vortex Ring Bubbles", *Journal of Fluid Mechanics*, , v.224, pp.177-196.

Taib, B.B., 1985,"Boundary Integral Methods Applied to Cavitation Bubble Dynamics", PhD Thesis, The University of Wollongong, Australia,.

Wang, Q.X., Yeo, K.S., Khoo, B.C. and Lam, K.Y., 1994, "Strong Interaction Between Buoyancy Bubble and Free Surface", *Theoretical and Computational Fluid Dynamics*, , Accepted for publication.

Zhang, S., Duncan, J.H. and Chahine, G.L., 1993,"The Final Stages of the Collapse of a Cavitation Bubble Near a Rigid Wall", *Journal of Fluid Mechanics*, , v.257, pp.147-181.

APPENDIX A

Master Program

```
enrol master program in PVM
start-up any number of slave program in the PVM
read input data (r, z, $\phi$, t)
broadcast (send) data to all
        slave programs ⇒INPUTDATA
set node id to zero (nodeid = 0)
do while (slaves are working or nodeid < npoints)
  if (nodeid < npoints) then
    do (for each slave program)
     if (slave is idle) then
       nodeid = nodeid + 1
       send the nodeid to the
        idle slave to compute
        solution for the node
        point (=nodeid) ⇒    ACTIVENODE
      endif
      if (nodeid = npoints) break
    enddo
    wait here for a result from a slave program
  endif
enddo while
terminate all slave programs (when bubble has
collapsed) ⇒ FINISHUP
leave PVM
terminate the program
```

Slave Program

```
enrol slave program in PVM
do (forever)
   wait for a message form the
   master program
   case :
     INPUTDATA
     get the broadcasted data from
     the master program
    break

    ACTIVENODE
     get the nodeid
     do calculations (solving
       equation (5)) for node nodeid.
       return solution of nodeid to
         master program
      break

     FINISHUP
      leave PVM
      leave the program
       break
    endcase
enddo
```

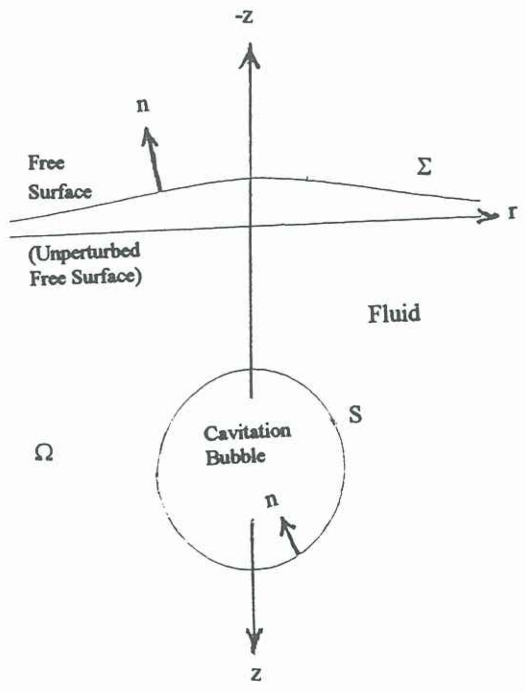

Figure 1 : Geometry and coordinate system for the growth and collapse of a cavitaon bubble near free surface.

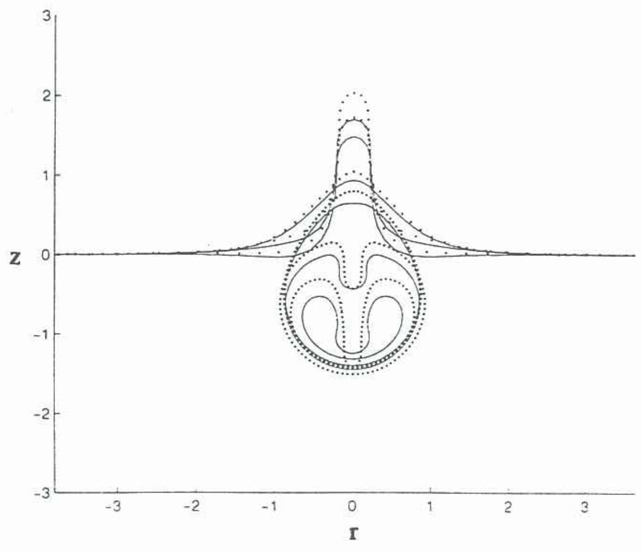

Figure 2 : The broadening of the free-surface jet and bubble penetration jet due to surface tension. (————) with surface tension (· · · · · ·) without surface tension

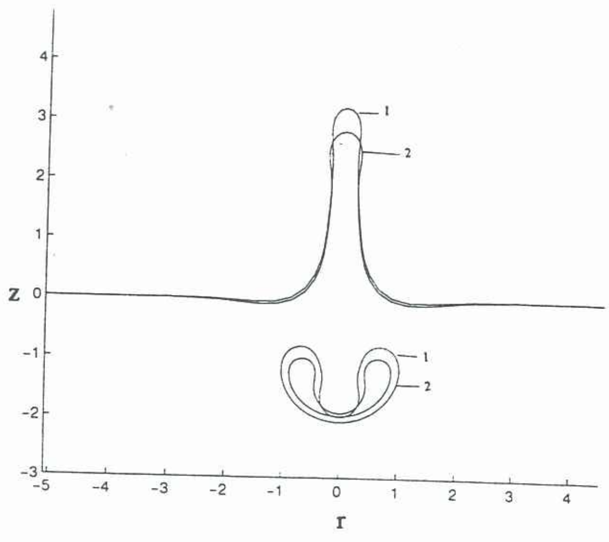

Figure 4 : The effect of surface tension on the maximum height of free-surface jet and geometry of both the free-surface and bubble surface during the final stage of collapse. 1) $1/\tau = 0.1$ 2) $1/\tau = 0.15$

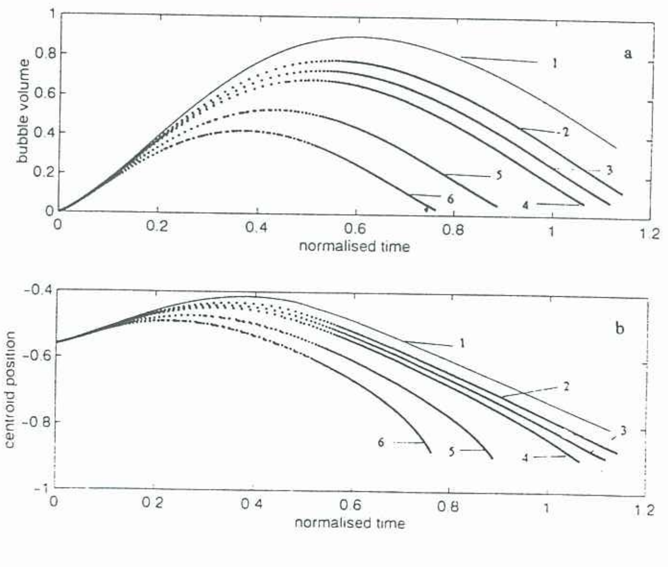

Figure 3 : (a) The life cycle, and (b) the centroid motion of the bubble with increasing surface tension. 1) $1/\tau = 0.0$ 2) $1/\tau = 0.05$ 3) $1/\tau = 0.075$ 4) $1/\tau = 0.1$ 5) $1/\tau = 0.2$ 6) $1/\tau = 0.3$