MPI PARALLELIZATION FOR TWO-WAY COUPLED EULER-LAGRANGE SIMULATION OF MICROBUBBLE ENHANCED HIFU

Jingsen Ma¹, Aswin Gnanaskandan, Chao-Tsung Hsiao, Georges L. Chahine
DYNAFLOW, INC.
www.dynaflow-inc.com
10621-J Iron Bridge Road, Jessup, MD, USA
¹Contact author: jingsen@dynaflow-inc.com

ABSTRACT
Microbubble enhanced High Intensity Focused Ultrasound (HIFU) is of great interest to tissue ablation for tumor treatment such as in liver and brain cancers, in which ultrasonic contrast agent microbubbles are injected to the targeted region to promote local heating while reducing pre-focal damage. To accurately characterize the acoustic and thermal fields during this process, a compressible Euler-Lagrange model is used. The non-linear ultrasound field is modeled using compressible N-S equations on an Eulerian grid, while the microbubbles are tracked as discrete singularities in a Lagrangian fashion with their dynamics computed. Their intimate coupling is realized through the local void fraction, which is computed from the instantaneous bubble volumes and locations, and then fed to the fluid continuum model.

Owing to demanding computational cost in real applications, schemes for significant speedup are highly desirable. We present here a MPI parallelization scheme based on domain decomposition for both the continuum fluid and the discrete bubbles. The Eulerian computational domain is subdivided into several subdomains having the same number of grids, while the bubbles are subdivided based on their locations corresponding to each subdomain. During each computation time step, MPI processors, each handling one subdomain, are 1) first used to execute the fluid computation, and 2) then to execute the bubble computations, 3) followed by the coupling procedure, which maps the void fraction from the Lagrangian bubble solutions into the Eulerian grids.

Steps 1) and 2) are relatively straightforward by routinely following regular MPI procedures. However, step 3) becomes challenging as the effect of the bubbles through the void fraction at an Eulerian point near a subdomain border will require information from bubbles located in different subdomains. Similarly, a bubble near a border between subdomains will spread its contribution to the void fraction of different subdomains. This is addressed by a special utilization of ghost cells surrounding each fluid subdomain, which allows bubbles to spread their void fraction effects across subdomain edges without the need of exchanging directly bubble information between subdomains and significantly increasing overhead. This void fraction implementation is verified by gas volume conservation before and after spreading the bubble effects. Other bubble effects such as thermal effects are handled in a similar way.

This parallelization scheme is validated and illustrated on a typical microbubble enhanced HIFU problem, followed by parallelization scaling tests and efficiency analysis.

INTRODUCTION
High Intensity Focused Ultrasound (HIFU) is applied in therapeutic and surgical medical interventions such as for tissue ablation in cancer treatment, particularly, of deep-seated tumors such as in the liver and brain [1,2], because HIFU is a truly noninvasive form of localized ablative therapy. To widen the clinical adoption of HIFU treatment, one promising technique is to inject ultrasonic contrast agents (UCA) in the target region [3,4]. The interaction between the UCA and HIFU increases target tissue heating via bubble dynamics. This allows use of lower intensity HIFU in order to lower the risk of undesirable pre-focal damage. Further understanding of microbubble assisted HIFU acoustic and thermal fields is essential to predicting the bio-effects of HIFU on the target tissue and its
surroundings and to developing standards to ensure the safety and efficacy of the treatments.

It is highly desirable to use a computational tool, which could enable accurate characterization of the acoustic and thermal fields for bubble enhanced HIFU in clinically relevant environments. For this purpose, we have developed a fully compressible Eulerian-Lagrangian approach which couples the acoustic field with bubble dynamics [5,6]. This method has shown intrinsic advantages owing to its capability to include the complex physics of nonlinear interactions between the oscillating bubbles and the ultrasound in the computation of heat deposition. However, owing to high computational cost in real applications, efficient parallelization schemes for significant speedup are critical.

In this paper, we present an MPI parallelization scheme based on domain decomposition for both the continuum fluid and the discrete bubbles to take advantage of MPI scalability due to distributed memory features. The Eulerian computational domain is subdivided into several subdomains each having the same number of grids, and subsequently the Lagrangian bubbles are subdivided based on their locations in each subdomain. The MPI realization for the fluid and bubble computations is relatively straightforward by following routine MPI procedures. However, the key step of two-way coupling, which requires mapping the Lagrangian bubble solutions into the fixed Eulerian grids is more challenging. This is due to the fact that we need to spread smoothly the effects of each bubble to all its adjacent computational cells and some might belong to other subdomains. To overcome this difficulty, we explored a novel introduction of ghost cells surrounding each fluid subdomain.

The paper is organized as follows. The main features of the two-way coupled, fully compressible Euler-Lagrange model are first summarized. This is followed by a description of the MPI parallelization scheme, particularly focusing on Euler-Lagrange Mapping. After a brief discussion of simulation results for a bubble enhanced HIFU problem [5,6], scaling tests and efficiency analysis of the parallelization scheme are conducted and discussed. Finally, a summary of the findings is presented.

**Eulerian-Lagrangian Model**

The 3D Eulerian-Lagrangian fully compressible model employed in this study has been documented and validated in our previous studies [5,6]. The present paper aims at accelerating the computations through MPI parallelization and does not consider additional model development. Thus, below we only provide a brief summary of the key features of the model and concentrate on the parallelization aspects. Readers interested in more details on modeling aspects may refer to the references provided above.

**Eulerian Compressible Mixture Flow Solver**

The unsteady compressible flow solver for acoustic wave propagation through a two-phase medium and other associated physics includes mainly the governing equations for conservation of mass, momentum, and energy as follows:

\[
\frac{\partial \rho_m}{\partial t} + \nabla \cdot (\rho_m \mathbf{u}) = 0,
\]

\[
\frac{\partial \rho_m \mathbf{u}}{\partial t} + \nabla \cdot (\rho_m \mathbf{u} \mathbf{u} + p_m \mathbf{I}) = 0,
\]

\[
\frac{\partial \rho_m E_m}{\partial t} + \nabla \cdot ((\rho_m E_m + p_m) \mathbf{u}) = \sigma \mathbf{u} + q = 0.
\]

where \( \rho_m \) is the mixture density, \( E_m \) its total energy, \( \mathbf{u} \) its velocity, and \( p_m \) its pressure. The mixture density can be related to the volume fraction \( \alpha \) of each component (such as tissue/phantom medium and the gas inside the bubbles) by:

\[
\rho_m = \sum_i \rho_i \alpha_i, \text{ where } \sum_i \alpha_i = 1.  
\]

Also, \( \rho_m \) and \( E_m \) are related to \( \rho_m \) with EOS which considers that the background medium obeys a stiffened equation of state [7].

The numerical scheme uses a fully conservative higher order scheme based on MUSCL method and an approximate Riemann solver [8,9].

**Lagrangian Modeling of Dispersed Phase**

For the bubbles dispersed in the medium, we use a Lagrangian bubble tracking method. All bubbles are treated as point sources to account for volume change and dipoles to account for translation. The bubble volume variations are obtained using a modified Rayleigh-Plesset-Keller-Herring equation [10]

\[
(1 - \frac{R}{c_m})RR + \frac{3}{2} (1 - \frac{R}{3c_m})R^2 = \left(\frac{\mathbf{u} - \mathbf{u}_b}{4}\right)^2 + \\
\frac{1}{\rho_m} (1 + \frac{R}{c_m}) \left(\frac{\mathbf{u} - \mathbf{u}_b}{4}\right) \left[\frac{p_v + p_g - p_m - \frac{2\gamma}{R}}{4\mu_m} \cdot \left(\frac{R}{3} - G \left[1 - \left(\frac{R_0}{R}\right)^3\right]\right)\right].
\]

In the above equation \( R \) is the bubble radius and the dots represent time differentiation. \( \mu_m \) is the mixture dynamic viscosity, \( c_m \) the local mixture sound speed, \( \mathbf{u}_b \) is the bubble velocity, \( p_v \) the gas pressure inside the bubble, and \( p_i \) the vapor pressure. \( \gamma \) is the surface tension, and \( G \) is the shear elasticity of the medium surrounding the bubble.

The bubble trajectory is obtained using the motion equation of Johnson, & Hsieh [11]:

Copyright © 2020 by ASME
\[
\frac{d\mathbf{u}_p}{dt} = -\frac{3}{\rho_p} \nabla p + \frac{3}{4} \frac{C_D}{R} (\mathbf{u}_p - \mathbf{u}_b) \cdot (\mathbf{u}_p - \mathbf{u}_b) \\
+ \frac{3}{2\pi} \frac{C_L}{R} \sqrt{\frac{\mu_m}{\rho}} (\mathbf{u}_p - \mathbf{u}_b) \times \mathbf{\omega} + \frac{3}{R} (\mathbf{u}_p - \mathbf{u}_b) R .
\]

where \( C_D \) is the drag coefficient, \( C_L \) the lift coefficient, and \( \omega \) the local vorticity.

**Eulerian-Lagrangian Coupling**

The two-way coupling between the Eulerian continuum-based model and the Lagrangian discrete bubble model is realized as follows [12-15]:

- The mixture flow field has an evolving mixture density in space and time which satisfies mass, momentum and energy conservation.
- The bubble dynamics and motion of the individual bubbles in the flow field are controlled by the two-phase medium’s local properties and gradients.
- The local properties of the mixture are determined by the bubble size and distribution.

**Bubble “gas volume” contribution range**

![Bubble “gas volume” contribution range](Image 85x462 to 258x474)

**Fig. 1. Sketch of the void fraction computation using the Gaussian distribution scheme [12].**

The key of this coupling procedure is to map to the fixed Eulerian grids the Lagrangian bubbles solutions, specifically, to derive the void fraction in each computational cell from the instantaneous bubble sizes and locations. As illustrated in Fig. 1, a Gaussian distribution scheme is employed to smoothly “spread” each bubble volume effect across neighboring cells [12-14]. The void fraction is computed using the following formulation:

\[
\alpha_i = \frac{\sum_{j=1}^{N_i} f_{i,j} V_{j}^b} {\sum_k f_{k,j} V_{k}^{cell}} ,
\]

where \( V_{j}^b \) is the volumes of bubble, \( j \), and \( V_{k}^{cell} \) is the volume of cell, \( k \), respectively. \( N_i \) is the number of bubbles which are in the “influence range” of a cell \( i \). \( N_{cell} \) is the number of cells “influenced” by a bubble \( j \). And \( f_{i,j} \) is a weight factor (Gaussian distribution) of bubble \( j \) contributing to a cell \( i \), which gradually decays to zero as the distance between the bubble and the cell increases.

**Modeling of Heat Deposition**

Once the multiphase flow field solution is obtained from the above-described approach, heat source terms, \( q_{US,AC} \) due to the ultrasound absorption and acoustic emission, and and \( q_{VIS} \) due to viscous damping of the bubbles are derived.

\[
q_{US,AC} = \mu_b \varepsilon_{kk}^2 + 2 \mu \left( \varepsilon_{ij} - \frac{1}{3} \varepsilon_{kk} \delta_{ij} \right)^2 ,
\]

where \( \mu \) and \( \mu_b \) are the shear and bulk viscosities of the fluid and \( \varepsilon \) is the strain rate tensor [6]. \( q_{VIS} \) is computed through a procedure very similar to that of void fraction computation described in Eq. (5), which sums up the smoothly-spread contributions from each bubble within the “influence range”:

\[
q_{VIS} = \sum_{k=1}^{N_{cells}} \frac{\sum_{j=1}^{N_i} f_{i,j} V_{j}^b V_{k}^{cell}} {V_{k}^{cell}} ,
\]

\[
q_{vis} = \left( 4 \pi R^2 \right) \left( 4 \mu \frac{R^2} {R} \right) .
\]

A heat transfer equation is then resolved to compute the heat deposition effect using the heat sources \( q_{US,AC} \) and \( q_{VIS} \).

\[
\rho C_p \frac{dT}{d\tau} = K \nabla^2 T + q_{US,AC} + q_{VIS} ,
\]

where \( \rho \), \( C_p \), and \( K \) are the density, specific heat, and thermal conductivity of the medium, \( T \) the temperature and \( \tau \) the time.

**PARALLELIZATION ALGORITHM & IMPLEMENTATION**

**Fig. 2. Illustration of the computation domain decomposition.**

**Algorithm and Overall Procedure**

As mentioned previously, the Eulerian computational domain is subdivided into several subdomains, as illustrated in Fig. 2. The number of subdomains is matched with the number of processors in the available distributed-memory system, or as required by the user. During every computation time step, all
involved MPI processors handle each one subdomain and operate simultaneously to:
1. Solve (1) and (2) for fluid computation,
2. Solve (3) and (4) for all bubbles,
3. Map the Lagrangian bubble solutions into the Eulerian grids of each subdomain using (5).
These operations proceed for each physical time step until the required final time is reached.

![Fig. 3. Definition of N layers of “ghost cells” surrounding a subdomain (here N=2). The red lines indicate the border of subdomain. The white cells inside subdomain are “real” cells. Those in blue along each border line are “ghost” cells.](image)

Note that in MPI parallelization, each processor operates independently except when communication with other processors is needed. Therefore, steps 1 and 2 are relatively straightforward and are very similar to serial computations. Inter-processor communication is necessary where cells near the subdomain edge require information from neighboring subdomains handled by different processors. This can be realized through the addition of ghost cells along the borders of each subdomain. As illustrated in Fig. 3, each subdomain is surrounded by “N” layers of ghost cells along all its edges, where N depends on the order of the selected numerical scheme.

Fig. 4 gives further details on how ghost cells are used to realize communication between Subdomain A and its neighbor B. As seen in the figure, the boundary data in the real cells of B are copied to fill the corresponding A ghost cells, which share the same geometric location. At the same time, the parallel operation is performed in the opposite direction, i.e., the boundary data from Subdomain A is used to fill the corresponding ghost cells data in B. This two-way data movement between neighboring subdomains could be easily realized by routinely calling the standard MPI library functions (e.g., MPI_SENDRECV).

However, step 3 is more challenging as the effect of the bubbles through the void fraction at an Eulerian point near a subdomain border will require information from bubbles located in different subdomains. Similarly, a bubble near subdomain borders has to spread its’ void fraction contribution to cells belonging to different subdomains. Therefore, below we mainly focus on addressing how to map Lagrangian bubble solutions into the Eulerian framework in the MPI environment.

![Fig. 4. Sketch of data copying from real cells of Subdomain B to the ghost cells of Subdomain A and vice versa from real cells of Subdomain A to the ghost cells of Subdomain B. (Note that the edges of the white real parts of A and B actually are superposed but have been separated here to illustrate the procedure)](image)

**Parallelization of Euler-Lagrange Mapping**
As known from above, each subdomain is surrounded by “N” layers of ghost cells along its edges. Traditionally, ghost cells do not participate in the computations, but are merely used to create a transfer zone to load data from the “real” cells along the edge of a neighboring subdomain. This means the data movement direction is: **Real → Ghost**, as indicated by the direction of the orange arrows in Fig. 4. However, to enable mapping in the Eulerian-Lagrangian scheme, we explored a different way of using the ghost cells, where they participate in the computations and then copy their values to the “real” cells along the edge of the neighboring subdomain (i.e., **Ghost → Real**).

![Fig. 5. Illustration of the void fraction contribution for a bubble (red sphere) near the edge of a subdomain A spreading its contribution into its neighboring subdomain B. The blue cells are ghost cells of A. The orange arrows indicate the “void” effects are copied from “ghost” cells of A and added into “real” cells along the left edge of B.](image)
As exampled in Fig. 5, a bubble is located close to the right edge of Subdomain A and needs to spread its void effects into neighboring Subdomain B, immediately to its right. Firstly, we have the bubble smoothly spreads its void effects into all adjacent cells regardless whether they are real or ghost cells. Then, using MPI communications, the void values in the A ghost cells are transferred to the neighboring B subdomain and added into the corresponding real edge cells (Note now the direction of orange arrows in Fig. 5 is opposite to those in Fig. 4). By doing so, the bubbles near the edge are able to spread void effects across subdomain edges without the need of sending bubble information to other subdomains. Vice versa the opposite condition, i.e. the bubble is near the left edge of subdomain B, can be handled in the same way.

The next example in Fig. 6 is for a more complex condition, where a bubble is located near the corner of a subdomain A, which has neighboring subdomains in two directions (e.g., top and right). Here again, the bubble first smoothly spreads its void effects into all adjacent cells no matter if they are real (in white), ghost (in blue), or doubly ghost (in green). Next, MPI communication is conducted direction by direction. As illustrated in the figure,

1) First, void fraction values are copied from the doubly ghost cells of A and added into the ghost cells along the top edge of the right subdomain B.

2) Then the updated values in those ghost cells of B are further added into the real cells along the bottom edge of subdomain C.

The same logic applies to three dimensional configurations.

For further validation purposes, we also consider the case of a spherical bubble cloud located at the center of a 3D cubic fluid domain, similar to our previous study in [15]. Fig. 8 shows the decomposition of the 3D domain into subdomains in the three directions. As seen in Fig. 9-Left, the resulting void fraction contours are smoothly continuous across all subdomain edges. More importantly, further very good quantitative validation is obtained in Fig. 9-Right by comparing the total gas volume in the cloud versus time obtained by summing at each time the volume of all bubbles in the cloud and by integrating at each time step the void fraction of all the fluid cells.

As mentioned earlier, other bubble effects such as thermal effects are handled in the same way as the void fraction computations.
TEST SIMULATION & RESULTS

In this section we apply the two-way coupled Eulerian-Lagrangian approach to simulate HIFU enactment using microbubbles, corresponding to in vitro experiments in [4]. In these tests, a 2.2 MHz transducer was used to insonate a phantom tissue made of Polyacrylamide gel in the presence of bubbles. The phantom tissue volume was cylindrical, with a radius of 25 mm and a height of 60 mm. In our computations, it is discretized in the cylindrical \( x, \theta, z \) domain using \( N_x=608 \times N_o=720 \times N_z=1415 \) cells. A cloud of bubbles of radius 1.3 µm are uniformly distributed in a smaller cylindrical volume of diameter 10 mm and length 10 mm and centered at the geometric focus of the excitation transducer \( (z = 40 \text{ mm}) \).

To take advantage of the axisymmetry of the problem, we used only 1 cell in the \( \theta \)-direction, i.e., reduced \( N_o \) to 1 and extended the solution to the full 360° domain. The bubble cloud was excited by an HIFU pressure wave emitted from the plane \( z = 0 \text{ mm} \), with a frequency of 2.2 MHz and an amplitude of 3.0 atm. The wave travels through the phantom tissue and reaches the bottom of the cloud. As seen in Fig. 10-Left, the simulation captures the scattering of the pressure waves by the bubbles and their attenuation before they reach the focus. The obtained temperature history near the focus is shown in Fig. 10-Right for three bubble cloud void fractions, \( \alpha=0, \alpha=10^{-3} \) and \( \alpha=10^{-4} \). The figure shows a clear trend of the temperature rise due to bubble effects, reasonably matching the experiments. More simulation details and result analysis can be found in [5,6]. Below we focus on the major objective of this paper, i.e., parallelization efficiency tests and analysis.

Scaling Tests and Speedup Analysis

To examine the scalability of the developed scheme, we tested the code for bubble enhanced HIFU simulations on DYNAB-HIVE, one of DYNAFLOW’s multi-nodes, multi-processors Linux clusters. DYNAB-HIVE consists of Ten nodes, with each equipped with twelve 3G Xeon CPUs and 27 GB memory. Before conducting the scaling tests, the accuracy of the parallelization is examined in Fig. 11. The pressure history is monitored at a selected field point near the focal region inside the bubble cloud for different selections of the number of processors. Also, the histories of the radius of a selected bubble also near the focus are compared for different numbers of processors. As seen in Fig. 11, for both comparisons, the results are very close for the different number of processors used.

Fig. 9. Left: Void fraction contour on the central plane of a spherical bubble cloud. Right: Comparison of time history of the total gas volume computed by summing bubble volume and integrating void fraction through all cells.

Fig. 10. Left: Instantaneous pressure contours from HIFU simulations in phantom tissue with bubbles in a 10x10mm cylindrical domain. Right: Comparison of the temperature history with the experimental data of [4] near the focus [5,6].

Fig. 11: Comparison of the time history of the pressure at a fixed point for different MPI processors being used (Top) and bubble size for a selected bubble inside the bubble cloud also for different number of processors (Bottom).
Fig. 12 shows the wall time per time step spent by the three main modules in the coupled computations. Fig. 13 shows the corresponding speed-up factors for increasing number of MPI processors. Here each line corresponds to one main part of the computation, i.e., Eulerian, Lagrangian and coupling through the Eulerian-Lagrangian mapping. It is seen that the wall time per time step for the Eulerian module (the most time-consuming part) drops from 15s to less than 0.15s as the processor number increases from 1 to 120, indicating a two-order of magnitude speedup. Through all the way of increasing processor numbers, the speed-up factor is consistently linear except negligible deviation for a few points Fig. 13. This is understandable and expected as the cells are evenly distributed when the fluid domain is decomposed into multiple sub-domains. Similar results are seen for the Lagrangian part of computation.

Fig. 12. Wall time per step for bubble enhanced HIFU simulation with the increase of the number of MPI processors.

However, surprisingly the speed-up of the Eulerian-Lagrangian coupling through mapping performs even better; a nearly 240 times speedup is achieved when 120 processors are used. This indicates that the newly introduced ghost cells procedure described above involves minimal overhead since it short-circuits sending bubbles across sub-domains as in traditional parallelization approaches. The new ghost cells approach transforms the MPI communication to an operation similar to the parallelization of the Eulerian part. In addition, since the mapping involves the interaction between both cells and bubbles, the numbers of both are reduced when the number of decomposed subdomains increases. This also explains why the parallelization of Eulerian-Lagrangian mapping part is better than a linear speedup.

Fig. 14 displays the speedup factor in terms of the total computational time per step summing the cost for all three parts. Owing to the good performance of each of these three parts presented above, an excellent speedup effect of our MPI parallelization is observed. In another word, this indicates at least a two-order of magnitude of speedup can be achieved just using our small in-house cluster. This is particularly helpful to carry out parametric studies of bubble enhancement of HIFU to optimize experimental designs or to plan clinical research.

CONCLUSION

An MPI parallelized 3D Eulerian-Lagrangian model was developed for microbubble enhanced HIFU problems. This parallelization scheme adopts domain decomposition for both the continuum Eulerian domain and the Lagrangian discrete bubbles. An innovative utilization of ghost cells surrounding each subdomain is designed to overcome the difficulties encountered in mapping the Eulerian and Lagrangian solutions. This new scheme avoids the need to sending bubble
information across all processors and enables the bubbles to smoothly spread their effects into all neighboring subdomains.

After careful verifications, efficiency tests were conducted for a bubble enhanced HIFU problem. More than two-orders of magnitude of speedup could be achieved using 120 processors on a Linux cluster consisting of ten 12-core Intel Xeon Nodes.

ACKNOWLEDGEMENT

We gratefully acknowledge the support from NIH under SBIR Phase II Grant No. 2R44CA213866-02

REFERENCES