Benefits of AMR for Atomization Calculations

C. W. Kuo[1] and M. F. Trujillo[1*]
[1]Department of Mechanical Engineering, University of Wisconsin - Madison, USA

Abstract

Adaptive mesh refinement (AMR) has been introduced as an attractive means of significantly improving computational efficiency for a variety of two-phase flow problems. In the current study, the benefits of AMR are investigated for the case of liquid jet atomization. The evaluation consists of a systematic analysis of results from the `interDymFoam` (AMR octree) and `interFoam` (static octree) codes, both of which form part of the family of solvers distributed within the open source OpenFOAM C++ Toolbox. The two-phase flow treatment is based on an algebraic VoF methodology. As a preliminary set of exercises, cases for pure advection, stationary wave dynamics, and Rayleigh-Plateau breakup of a cylindrical liquid element are considered. The results from these exercises confirm the expected trend of higher numerical efficiency in AMR, while still retaining essentially the same level of accuracy as the fixed embedded mesh solutions. However, for the liquid jet atomization, the behavior is a bit more complicated. First, at lower levels of Weber number, we observe a similar trend as the preliminary exercises. At higher Weber numbers, due to a noticeable increase in interfacial area density, substantial inhomogeneities are formed in the underlying grids yielding slower solutions of pressure Poisson equation, thereby potentially offsetting the benefits of this approach. In fact, at much higher Weber numbers, for instance, those pertaining to Diesel injection, the results suggest that a fixed embedded mesh would provide better computational efficiency. However, this conclusion depends on the target lowest level of numerical resolution, $\Delta x_{\min}$. The current work shows how the efficiency of AMR suffers from increasing interfacial area density, and how this can be alleviated via a decrease in $\Delta x_{\min}$. Various test cases are presented to illustrate this effect.

Keywords: Adaptive Mesh Refinement, Computational Cost, Atomization

1. Introduction

Adaptive mesh refinement (AMR) was introduced by Berger and Oliger [1] and Berger and Colella [2] as a means of dynamically allocating a high level of numerical fidelity in areas requiring it. A common way to achieve this is to dynamically reduce the grid spacing, Δx, in these regions of high-fidelity demand, which can be identified through the use an AMR cost function. One of the cost functions used in the early AMR work is the local truncation error [1][2] based on Richardson extrapolation method. For two-phase problems, the interface is the most obvious choice for the cost function, since it is the location where the highest level of resolution is needed. For instance, in the work of Theodorakakos and Bergeles [3], the tagging of an interface is characterized by having the liquid fraction α between 0.2 and 0.8. In the work of Malik et al. [4], the tagging is instead executed where the normalized curvature, the product of curvature and the cell size, is below 0.2 [5]. Common standard problems that are employed in 2-phase flow investigations to gauge the degree of AMR acceleration include rotation of Zalesak sphere [6][7], droplet deformation in the 3D vortical flow [6][8], Rayleigh–Taylor instability [9][10], and the secondary breakup of a droplet [11]-[14]. Among these, the works of Laurmaa et al. [7] and Zuzio and Estivalezes [9] further demonstrate that the smaller the value of the minimum grid size, Δx_{min}, the better speedup benefit AMR would attain.

Considering the case of spray formation and liquid jet atomization, Fuster et al. [5] reported that by using AMR, a 50% reduction of total cell count could lead to a 38% saving of total CPU time. Within the Lagrangian-Eulerian spray modeling approach, a common procedure for modeling sprays, the work by Tonini et al. [15] showed that the use of AMR could save 13%-29% of CPU time with a minimum mesh size of 0.15 mm. The paper by Xue and Kong [16] reported that a computational savings of up to 67% and 79%, for hollow-cone and solid-cone sprays, could be achieved. A potential issue with AMR may be its poor scaling. As indicated by Li and Soteriou [17], the scalability of AMR was worse than that of fixed embedded mesh in terms of the deviation from the ideal strong scaling slope. This finding motivates a closer look at the AMR benefits for spray problems.

In the present work, the advantage envisioned by AMR is examined for the atomization and spray formation problem. Due to significantly more pronounced interfacial area changes as opposed to the milder two-phase flow problems [6]-[14], it is shown that AMR does not consistently show an improvement over a well-designed static mesh with an embedded high-resolution region. The reasons for this performance are examined leading to the definition of a metric, which is shown to indicate when AMR improvements are, and are not, to be expected. Reasons for this lack of performance are also briefly discussed.

The evaluations originate from codes `interDymFoam` (octree-based AMR) and the `interFoam` (octree-based fixed embedded mesh), both of which form the family of algebraic VoF (Volume of Fluid) solvers in the open source C++ CFD platform (version 2.1.1) [18]. With the addition of adaptive mesh capability, `interDymFoam` is an extended version of `interFoam`, where the two codes employ the same discretization and time marching procedure. The explanation of the algebraic VoF approach employed in both codes, as well as its verification/validation can be found in Ref. [19]. The outline of the rest of paper is as follows. In section 2, we would illustrate the AMR implementation in `interDymFoam`. In section 3, standard exercise tests commonly used in interface capturing codes are performed. We then proceed to systematically evaluate the benefits of AMR for spray cases in sections 4 and 5.

2. AMR implementation and Performance Evaluation

The implementation procedure of AMR is initiated with the evaluation of a cost function for identifying cells in need of further refinement or coarsening. The criteria employed is

$$10^2 > |\nabla\alpha|\times\Delta x > 10^{-4} \quad \text{for mesh refinement} \tag{1}$$

$$|\nabla\alpha|\times\Delta x < 0.9\times10^{-4} \quad \text{for mesh coarsening,} \tag{2}$$

where $|\nabla\alpha|\times\Delta x$ is the product of the magnitude of the liquid fraction gradient, $|\nabla\alpha|$, and the local mesh size Δx. If the cell-centered value of a cell satisfies Eq. (1) (parent cell), it would be recursively split into eight smaller child cells following the octree structure (see Fig. 1(a)). The sizes of the child cells are given by

$$\Delta x_{refined} = 0.5^N\times\Delta x_{original} \text{ for } N-\text{level refinement.} \tag{3}$$

For the sake of narrowing the scope of the present investigation, we are only considering a two-level AMR (N=2), although a three-level or higher AMR is also available in `interDymFoam`. Once a refinement has taken place, the flow field is mapped from the parent cell to the child cells by assigning them the same value [20],

$$\emptyset_{c,i} = \emptyset_P \quad i = 1,2,3,\dots,8, \tag{4}$$

where $\emptyset_{c,i}$ is the cell-centered value of the i^{th} child cell, and $\emptyset_P$ is the corresponding value of the parent cell.

In the case of mesh coarsening, i.e. if any of the eight child cells belonging to the same node satisfies Eq. (2), the cluster of these child cells would be inversely coarsened to be a single parent cell (see Fig. 1(b)). The cell-centered value of the recovered parent cell would be the average of its child cells [20],

$$\emptyset_P = \frac{1}{8}\left(\sum_{i=1}^{8}\emptyset_{c,i}\right). \tag{5}$$

This applies to the liquid fraction, velocity field, and any other cell-centered quantity. Other schemes for mapping parent-child or child-parent have been presented in the literature [3][16], which applied to more general grid structures. For our case, the present method offers fast performance and is sufficiently accurate for the type of overall mesh structure employed. In the computations that we have performed, we have observed that the ratio of the time span during refinement to that of coarsening is around 0.8 for a spray configuration.

Figure 1 Mesh updating steps of AMR: (a) refinement (b) coarsening.

The jet atomization computations using the AMR grid are compared to a static mesh with an embedded highly-refined region as illustrated in Fig. 2. The global domain extent in all computations is 1260μmx1260μmx9150μm along the respective (x, y, z) directions, where z-direction is the spray axis. For the sake of performing consistent comparisons, the minimum mesh size of the fixed embedded mesh (Fig. 2a) and AMR (Fig. 2b) is the same, namely Δx_{min}=3.21μm.

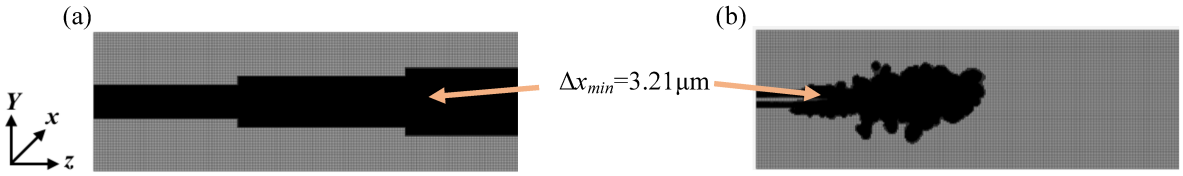

Figure 2 Spatial mesh distribution shown at the centerline plane of the spray configuration: (a) fixed embedded mesh (b) AMR.

For the performance evaluation, the computations are performed up to a physical end time, t_f. The following metrics quantify the AMR speedup:

$$\text{speedup}_{(n)} = \frac{t_{IB}}{t_{AMR}} \tag{6}$$

$$\text{speedup}_{(\text{Cell})} = \frac{t_{IB}/N_{cell,IB}}{t_{AMR}/N_{cell,AMR}} \tag{7}$$

$$\text{speedup}_{(\alpha,\text{U,P})} = \frac{t_{IB}(\alpha,\text{U,P})/N_{cell,IB}}{t_{AMR}(\alpha,\text{U,P})/N_{cell,IB}}, \tag{8}$$

where t_{IB} and t_{AMR} is the total elapsed CPU time for the fixed embedded mesh and AMR cases, respectively (using exactly the same number of compute nodes). Equation (6) provides the most straightforward approach to performance evaluation. Equation (7) computes the AMR efficiency as defined by Fuster et al. [5], since it accesses how much AMR speedup could be attained for a given reduction of the number of cell counts. For instance and speedup of 1.0 would correspond to a reduction in computation time in proportion to the reduction in total cell count. Equation (8) is aimed at providing the insight into whether AMR efficiency is substantially affected by the AMR implementation, so it would only consider the time cost of solving the governing equations of VoF, namely, the transports of liquid fraction and momentum, and the pressure-Poisson solution. Naturally, a speedup value above one computed by either Eqs. (6)-(8) represents favorable AMR performance.

3. Standard exercises

The following standard cases are often used when analyzing the performance of an interface capturing scheme. They are repeated here to establish the benchmark expectation for the performance of AMR. The cases correspond to 3D pure-advection [6][8], stationary wave dynamics [9], and the 3D Rayleigh-Plateau breakup of a cylindrical liquid element [21], and are illustrated in Fig. 3. The minimum mesh size of each exercise is Δx=1/320, $\lambda/\Delta x$=160, and Δx=0.5μm, respectively. Here λ is the wavelength of the initial perturbation imposed on the stationary wave.

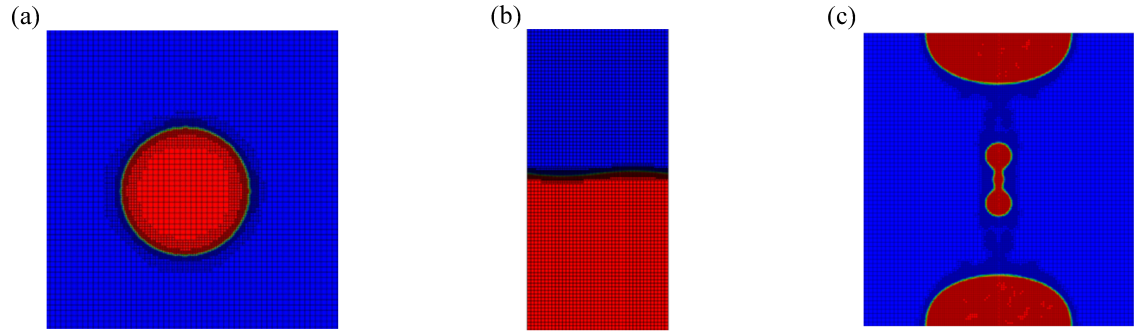

Figure 3 Partial results of the standard exercises showing the liquid fraction contours: (a) 3D pure advection showing the initial object shape [6][8] (b) stationary wave dynamics [9], and (c) Rayleigh-Plateau breakup of a cylindrical liquid element [21].

Figure 4 shows the $\text{speedup}_{(n)}$ performance of AMR based on Eq. (6) both on an instantaneous level (blue curve) and on a time-average level (red-curve). The average $\text{speedup}_{(n)}$ using AMR is 4 to 7, essentially confirming the AMR benefit reported in the literature. In all of these cases, grid refinement/coarsening operations were performed every 3, 5, and 5 time steps, respectively for pure advection, stationary wave dynamics, and Rayleigh-Plateau breakup. The minimum values in instantaneous $\text{speedup}_{(n)}$ correspond to time steps when AMR is active, and conversely the maximum values in $\text{speedup}_{(n)}$ coincide with AMR being dormant.

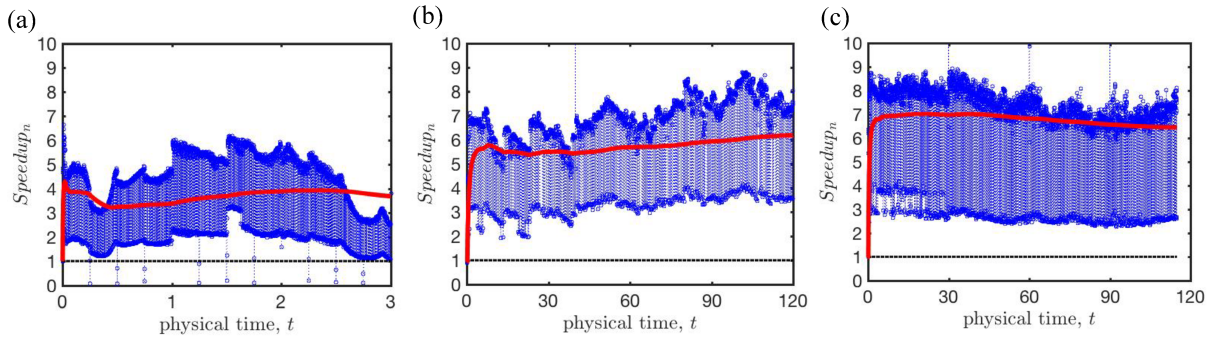

Figure 4 Speedup of AMR for the standard exercises: (a) 3D pure advection (b) stationary wave dynamics, and (c) Rayleigh-Plateau breakup of a cylindrical liquid element. The instantaneous and time-averaged values are presented in blue and red curves, respectively.

4. Spray Configuration

Now we would turn the focus to the more realistic spray atomization problems using the injection velocity (U_{inj}), and hence the Weber number (We), as a means of controlling atomization severity. The Weber number is given by,

$$We = \rho_L U_{inj}^2 D/\sigma, \tag{9}$$

where ρ_L is the density of the liquid and σ is the surface tension coefficient. The test cases for spray configuration are summarized in Table 1.

Table 1 The description of the test cases for spray configuration using a water-ethyl acetate system. The properties of water are: density (ρ_L) is 997 kg/m^3, kinematic viscosity is 8.927×10^{-7}m^2/. The properties of ethyl acetate are: density is 902 kg/m^3, and kinematic viscosity is 4.723×10^{-7}m^2/s. The surface tension coefficient (σ) is 0.0068kg/s^2.

	Case #1	Case #2	Case #3
U_{ini}	25 m/s	50 m/s	100 m/s
We	8,247	32,989	131,960

To avoid running into issues related to significant changes in the atomization behavior as the mesh resolution is enhanced, an adequate grid size, Δx, was a priori identified. This adequate mesh refinement is defined as the largest grid refinement for which the intact liquid length (L_{liq}) reaches reasonable numerical convergence. The intact liquid length is defined as

$$\frac{< L_{liq} >}{D} = \frac{1}{t_f - 100\mu s} \int_{100\mu s}^{t_f} L_{core}(t')dt', \tag{10}$$

where 100 μs is an elapsed time used to ensure the initial transient has passed, and the final value of t_f is adequately long to ensure a statistically stationary result. The orifice diameter, D, is equal to 90μm. Results corresponding to $D/\Delta x$ =18, 24, 28, are shown in Fig. 5(a) and Fig. 5(b) for injection velocities of We=32,989 (U_{inj}=50 m/s) and We=131,960 (U_{inj}= 100 m/s), respectively. Good convergence is attained at $D/\Delta x$ =28, where the difference between the finer grids is 5.6% for We=32,989, and 2% for We=131,960. Consequently, $D/\Delta x$ =28 would be regarded as the benchmark mesh size in the following analysis.

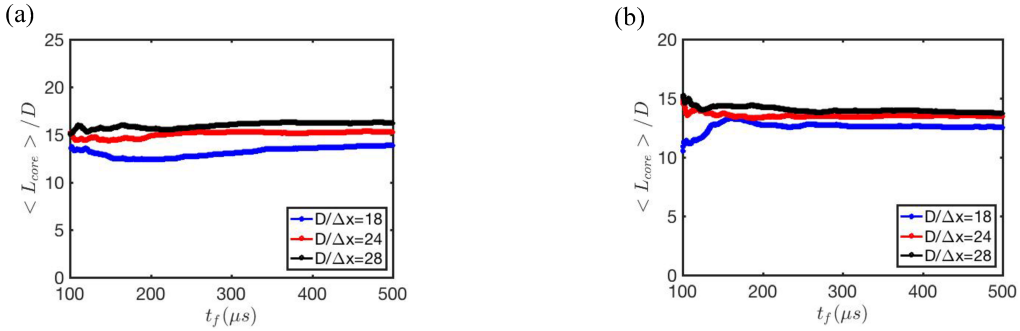

Figure 5 Grid sensitivity test: (a) *We*=32,989, (b) *We*=131,960.

The injection velocity $\boldsymbol{U}$ is described in the cylindrical coordinate ($\boldsymbol{e_r}$, $\boldsymbol{e_\theta}$, $\boldsymbol{e_Z}$) by,

$$\boldsymbol{U} = 0.01\left[U_{inj} - U_{inj}(1 - r/R)^{1/7}\right]\boldsymbol{e_r} + 0\boldsymbol{e_\theta} + U_{inj}(1 - r/R)^{1/7}\boldsymbol{e_z}, \tag{11}$$

where r is the radial distance, and R is the radius of the orifice radius. The one-seventh power law stems from the common findings of the mean injection velocity profile in the channel flows. The radial velocity formulation is motivated by velocity profile computations using the ECN Spray A configuration.

For the AMR mesh shown in Fig. 2(b), a uniform coarse grid is used as the initial condition. As spray is injected from the orifice, the cluster of grids of $O(10)$ cells across the interface with the interface being the midplane would be adaptively refined. For the fixed embedded mesh, on the other hand, the static telescope-shaped refined grid with the extent of 600μm×600μm×5400μm respectively in *x*-, *y*-, *z*-directions is imposed, expecting for containing all possible atomization regions (see Fig. 2(a)). Consequently, the number of grids in AMR should always be smaller than that of fixed embedded mesh, and the difference is expected to shrink as time evolves. This is confirmed in Fig. 6, where the three cases shown have a respective value for *We* = 8,247 (U_{inj}=25m/s), *We* = 32,989 (U_{inj}=50m/s), and *We* = 131,960 (U_{inj}=100 m/s).

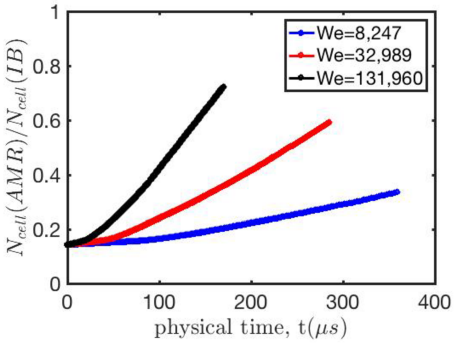

Figure 6 Ratio of AMR and fixed grid computational cells.

The speedup results are displayed in Fig.7. In Fig. 7(a), speedup$_{(Cell)}$, based on Eq. (7) is shown as a function of time corresponding to the different atomization cases. In Fig. 7(b), the speedup$_{(\alpha,U,P)}$, is displayed, but now excluding the costs associated with refinement and coarsening of the mesh. To enforce a consistent comparison, the time step size is the same for both AMR and the fixed embedded mesh. The findings show two trends. The first trend is that initially, when the interfacial area has not grown much, AMR outperforms the fixed embedded mesh significantly. This is in agreement with the conventional two-phase flow cases that demonstrate that for mild changes in the interfacial area the efficiency introduced by AMR is well worth its usage. However, as the calculation proceeds and atomization leads to the vast growth of interfacial area at various locations, the performance of AMR decays to such a degree that it becomes less efficient than the fixed mesh. Hence, the purpose of AMR is no longer met in the main part of the atomization process.

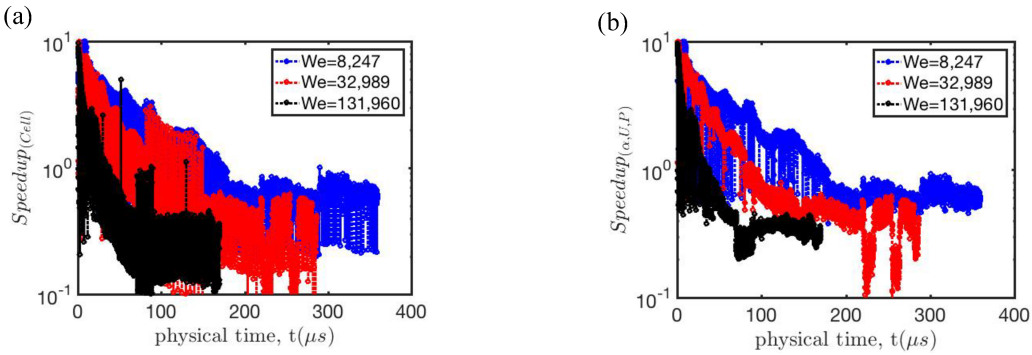

Figure 7 Speedup analysis for spray configurations: (a) total CPU time, (b) the CPU time excluding the costs of mesh refinement and coarsening.

The second trend indicates that the time and the severity of the AMR performance decay occurs sooner and more drastically as the global Weber number increases. Again, this is linked to an increased severity of the atomization process, and is discussed in the followed section.

5. Discussion

As the Weber number increases, the speedup offered by AMR suffers drastically to the point where it eventually becomes slower than the fixed embedded mesh. This is related to the large quantity of refinements taking place in an increasing number of places as the atomization process becomes more severe with increasing Weber number and time. Motivated by this finding, we define the following interfacial area density,

$$\rho_\Gamma = A_\Gamma/\Omega, \tag{12}$$

where A_Γ is the interfacial area, and Ω is a relevant macroscopic domain. The refined domain within AMR for fully covering the interfacial regions, Ω_R, is defined as,

$$\Omega_R = A_\Gamma K_1 \Delta x_\Gamma, \tag{13}$$

where K_1 is $O(10)$ and is more than large enough to contain the interfacial region within the VoF representation along with a band of 5 or more neighboring cells. The interfacial cell size, Δx_Γ, is the size within this interfacial region, such that $\Delta x_\Gamma = \Delta x_{min}$. Combining Eqs. (12) and (13) leads to the normalized interfacial area density,

$$\Omega_R/\Omega = \rho_\Gamma K_1 \Delta x_\Gamma. \tag{14}$$

From the results shown below in Fig. 8, it is evident that the high efficiency of AMR coincides with a value for Ω_Γ/Ω that is significantly below one. These conditions pertain to an interfacial region that is well below the size of the macroscopic domain. Of course, as atomization develops and the interfacial area expands throughout Ω, the ratio, Ω_R/Ω, grows considerably leading to the difficulties with AMR performance as indicated in Fig. 8. The efficiency of AMR could be improved by systematically reducing, Δx_Γ, since this would directly cause the interfacial region ratio, Ω_R/Ω, to be reduced well below one. In fact, this coincides with the conclusions of Laurmaa et al. [7] and Zuzio and Estivalezes [9].

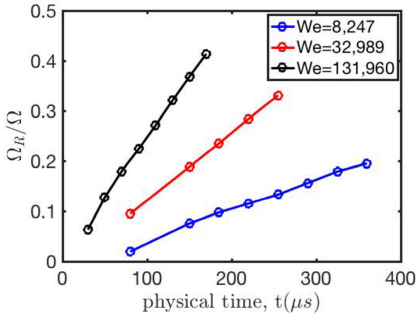

Figure 8 Evolution of the ratio of interfacial and macroscopic regions.

6. Conclusion

In this study, an evaluation of the benefits of AMR for simulations concerning liquid jet atomization is provided. Beginning with canonical or standard two-phase problems, our findings agree with the literature, i.e. AMR offers attractive speedups. However, in the case of atomization, the results show that the AMR performance becomes significantly poor, even worse than a well-designed fixed mesh, once atomization has developed and as the Weber number increases. A key metric used to determine when we expect to achieve good results is provided, namely $\Omega_R/\Omega = \rho_\Gamma K_1 \Delta x_\Gamma$. Good performance is expected for values of this metric that are well below one, and conversely as this ratio grows, AMR performance suffers. This metric indicates that systematic reductions in Δx_Γ, lead to the continued strong performance by AMR. Unfortunately, since Δx_Γ, is inherently linked to the time step size through the CFL restriction, this continued reduction in interfacial grid spacing will at some point stop being practical in realistic spray configurations.

7. Acknowledgments

The support from the Caterpillar Corporation and the National Science Foundation (#1703825) is gratefully acknowledged. We are also thankful for the Advanced Computing Initiative at UW-Madison for providing computational resources.

References

[1] Berger, M. J., Oliger, J., *Journal of Computational Physics* 53: 484-512 (1984)
[2] Berger, M. J., Colella, P., *Journal of Computational Physics* 82: 64-84 (1989)
[3] Theodorakakos, A., Bergeles, G., *International Journal for Numerical Methods in Fluids* 45: 421-439 (2004)
[4] Malik, M., Fan, E., and Bussmann, M., *International Journal for Numerical Methods in Fluids* 55: 693-712 (2007)
[5] Fuster, D., Bague, A., Boeck, T., and et al., *International Journal of Multiphase Flow* 35: 550-565 (2009)
[6] Anjos, G.R., Borhani, N., Mangiavacchi, N., Thome., J.R., *Journal of Computational Physics* 270: 366-377 (2014)
[7] Laurmaa, V., Picasso, M., and Steiner, G., *Computers and Fluids* 131: 190-204 (2016)
[8] Chen, X., Yang, V., *Journal of Computational Physics* 269: 22-39 (2014)
[9] Zuzio, D., Estivalezes, J. L., *Computer & Fluids* 44: 339-357 (2011)
[10] Xie, Z., Pavlidis, D., Percival, J. R., and et al., *International Journal of Multiphase Flow* 67: 104-110 (2014)
[11] Strotos, G., Malgarinos, I., Nikolopoulos, N., and Gavaises, M., *International Journal of Multiphase Flow* 85: 96-109 (2016)
[12] Yang, W., Jia, M., Sun, K., and Wang, T., *Fuel* 174:25-35 (2016)
[13] Tavangar, S., Hashemabadi, S. H., and Saberimoghadam, A., *Fuel Processing Technology* 132: 153-163 (2015)
[14] Jain, M., Prakash, R. S., Tomar, G., and Ravikrishna, R. V., *Proceedings of Royal Society A* 471:20140930 (2015)
[15] Tonini, S., Gavaises, M., and Theodorakakos, A., *International Journal of Heat and Fluid Flow* 29: 427-448 (2008).
[16] Xue, Q., Kong, S. C., *Computers and Fluids* 38: 939-949 (2009)
[17] Li, X., Soteriou, M. C., *50th AIAA Aerospace Sciences Meeting*, Nashville, TN, January 9–12, AIAA Paper No. 2012-0175 (2012).
[18] OpenFOAM-2.1.1 2012 *OpenFOAM: The Open Source CFD Toolbox* http://www.openfoam.com/ (2012)
[19] Deshpande, S. S., Anumolu, L., and Trujillo, M. F., *Computational Science & Discovery* 5 014016 (2012)
[20] Baniabedalruhman, A., *Dissertation, Michigan Technological University* (2015)
[21] Lafrance, P., *Physics of Fluids* 18: 428-432 (1974)