Multiscale approach to fully resolved boundary layers using adaptive grids

Citation for published version (APA):
Panda, A., Peters, E. A. J. F., Baltussen, M. W., & Kuipers, J. A. M. (2017). Multiscale approach to fully resolved boundary layers using adaptive grids. In J. E. Olsen, & S. T. Johansen (Eds.), *Progress in Applied CFD – CFD2017: Proceedings of the 12th International Conference on Computational Fluid Dynamics, 30 May - 1 June 2017, Trondheim, Norway* (pp. 709-715). SINTEF Academic Press.

Document status and date:
Published: 31/05/2017

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

MULTISCALE APPROACH TO FULLY RESOLVED BOUNDARY LAYERS USING ADAPTIVE GRIDS

A. PANDA[1]*, **E.A.J.F. PETERS**[1], **M.W. BALTUSSEN**[1], **J.A.M. KUIPERS**[1]

[1]Multiphase Reactors Group, Department of Chemical Engineering and Chemistry,
Eindhoven University of Technology, 5600 MB Eindhoven, THE NETHERLANDS

* E-mail: a.panda@tue.nl

ABSTRACT

Bubbly flows are omnipresent in most industrial processes. Often the intended use of such processes is to facilitate efficient mass and heat transfer for reactive flows. Mass and heat transfer coupled with fluid flow in gas-liquid systems gives rise to multiscale transport phenomena. Because of large Schmidt and (possibly) Prandtl numbers in the liquid phase the concentration and temperature boundary layers are much thinner than the momentum boundary layers. When using fully resolved CFD modeling on an uniform grid, these small scales would demand an overall refinement which requires an immense computational effort. Here, however, a hybrid mesh approach is used which couples a fixed Cartesian grid for the hydrodynamics and a tree structure based mesh, which can be adaptively refined for heat and mass transfer. Tree based adaptive refinements commonly suffer from low order accurate numerical schemes. A higher order finite volume scheme on a parallel tree data structure for solving the convection-diffusion equation has been implemented using an implicit formulation. The resulting set of linear algebraic equation are then solved with AMG class of matrix solvers. This approach presents a solution to resolve the fine boundary layers of scalar transport for realistic range of Schmidt and Prandtl numbers. The present study will demonstrate the robustness of this framework to capture sharp boundary layers in fairly simple analytical flow fields. A detailed comparison is performed with overall refined simulations on multi core parallel architectures.

Keywords: adaptive grids, fully resolved simulation, boundary layer, multiscale transport, heat and mass transfer .

NOMENCLATURE

Greek Symbols

ρ	Mass density, $[kg/m^3]$
μ	Dynamic viscosity, $[kg/ms]$
ϕ	General scalar, $[-]$

Latin Symbols

Sc	Schmidt Number, $[-]$
Pr	Prandtl Number, $[-]$
h	Cell size, $[-]$
L	Cell level, $[-]$
L_i	Error norms, $[-]$
p	Pressure, $[Pa]$.
$\mathbf{u}$	Velocity, $[m/s]$.
V	Volume of cell, $[m^3]$.
a	Area of cell face, $[m^2]$.
c	Concentration, $[mol/L]$
T	Temperature, $[^\circ C, ^\circ K]$
C	Cell, $[-]$

Sub/superscripts

d	Direction.
f	Face centre.
n	Time step.
i	Index i.
j	Index j.
$1,2,\infty$	Type of error norm.

INTRODUCTION

Bubble columns are one of the widely used and very interesting processing units for contacting gas-liquid flows in the chemical industry. In these bubble columns the gas phase is dispersed in the continuous liquid phase. The rising bubbles increase the mixing of the liquid phase thus providing improved mass and heat transfer properties. In spite of the simple structure of bubble columns, the physical phenomena occurring at the gas liquid interface is still not completely understood. Therefore, the design of the columns is mainly based on experimental data and empirical or semi-empirical correlations. To improve the design of bubble columns, a deeper understanding of the physics is necessary.

Due to the increase of computational power, it is possible to simulate many flow situations using direct numerical simulation (DNS), which simulates the flow based solely on first principles. Although, the methods to determine the hydrodynamics of multiphase flows are well known, adding new physics i.e. mass transfer, heat effects or chemical reaction still are faced with a lot of challenges. The challenges occur due to the multitude of length scales and time scales these phenomena posses. From the dimensionless numbers which describe these phenomena, such as Schmidt (Sc) number in case of mass transfer and Prandtl (Pr) number in case of heat transfer, it can be determined that the boundary layers in many industrial cases can vary by several orders of magnitude. For example, in case of mass transfer from a bubble to liquid, the mass diffusivity is generally at least two to three orders of magnitude smaller than momentum diffusivity. This condition enforces a constraint on the mesh resolution and the time step required for sufficiently resolving mass transfer boundary layer at the bubble interface thus increasing the computational cost.

Several different methods can be used to enable the computation of all of these different length and time scales. First of all the overall mesh can be refined (Darmana *et al.*, 2006; Roghair *et al.*, 2016). However, the overall mesh refinement leads to billions of cells to solve boundary layer problems corresponding to realistic *Sc* ($Sc \sim 10^3$). This requires a lot of computational effort therefore, leading to limitation of simulating systems which have very modest *Sc* numbers ($1 \lesssim Sc \lesssim 10$). Secondly the boundary layer can be approximated by using subgrid scale models (Aboulhasanzadeh *et al.*, 2012; Gründing *et al.*, 2016), which are based on self similar solutions corresponding to the mass boundary layer. However, in industrial reactors, the flow patterns are complex due to frequent bubble-bubble interactions, which significantly affect the bubble hydrodynamics and thereby influencing the interfacial mass transfer. In such dense bubbly flow regimes validity of overly simplistic boundary layer assumptions is not well established. Finally, the mesh could be adapted according to the local gradients in scalar field (Deising *et al.*, 2016). Adaptive mesh refinement (AMR) has its own set of computational challenges specific to different methods of mesh refinement. It's one of the advanced research areas in CFD as many well formulated methods which work on simple Cartesian grids cannot be simply extended to adaptive meshes.

Among many different AMR strategies, notable are block structured AMR, overset or Chimera grids, tree based AMR etc. For non deformable interfaces, a Chimera or overset grid presents a plausible solution to resolve the boundary layer. For interfaces which are deformable such as those occurring in bubbly flows and for an optimal mesh count, a tree based grid adaption is well suited. Tree based AMR also suffers from some limitations such as efficient data structures, parallelization, load balancing using graph partitioning methods and efficient matrix solvers. It is important to note the third limitation pertaining to matrix solvers is more stringent, as the linear system of equations on such adaptive grids leads to unsymmetric matrices which are computationally expensive to solve. Algebraic multigrid class of matrix solvers are suited for such systems which, if combined with parallelization can give a performance improvement. In any CFD calculations, most of the time is spent on solving the Poisson problem arising from the incompressibility constraint on the momentum equation. Solving it becomes even more difficult for multiphase flows with high density ratios such as air-water system. AMR grids based on a dynamic data structure such as quadtree or octree are considerably slower for the same grid count because of issues related to dynamic memory layout and reduced vectorization in comparison to array based data structures which are used for Cartesian grids.

To circumvent this problem, a novel hybrid grid based method is proposed for fully resolved multiscale transport on adaptive grids. The essence of the idea is to only solve the scalar transport equations on the AMR grid while solving the hydrodynamics on the Cartesian grid. The methodology and implementation of a general scalar convection diffusion equation on parallel adaptive mesh refined grids is described in later sections. The resulting convection diffusion solver is verified and validated with model test cases. Also, a methodology is presented to encode a staggered velocity field onto an AMR grid, which can then be used to fetch velocity values from a staggered Cartesian hydrodynamics grid. A divergence free interpolation is also presented, for interpolating velocity values during mesh refinement on to the finer grid.

MODEL DESCRIPTION

Spatial Discretization

Generally, a flow domain can be represented in the form of control volumes which are square in 2D (or a cube in 3D). These square (or cubic) meshes are adapted using a quadtree (or octtree) data structure which also finds its use in image processing & computer graphics (Popinet, 2003). Figure 1 depicts a representation of such discretization in case of a 2D tree. Each control volume is called as a *cell* and can have 4 descendants (8 in case of 3D) called *children*, in which case, the cell is called as *parent* of these children. A cell which has no parent cell is called as the *root* cell while cells which do not have any children are called as *leaf* cells. The length of any cell edge is denoted by *h*. The *level* of a cell is defined with the root cell as reference 0 and children being one level higher than the parents. The spatial discretization in the form of quadtree needs to satisfy 2:1 balance constraint, which states that adjacent cells' levels must not differ by more than one i.e. a maximum one hanging node is possible at any cell face. This constraint restricts the number of possible cases which in a way helps in devising efficient numerical schemes on such spatial discretization.

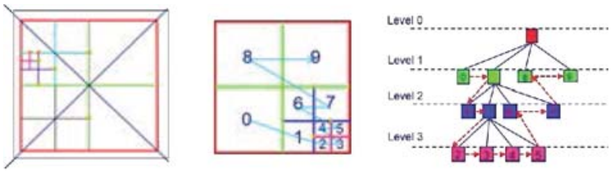


Figure 1: Schematic diagram of quadtree spacial discretization.

Governing Equations

The generalised scalar convection-diffusion equation along with the Navier Strokes equation can be written as :

Scalar Convection Diffusion Equation

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = D \nabla^2 \phi + F_s \tag{1}$$

Navier-Stokes Equation

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \nabla \cdot (\mathbf{uu}) = -\nabla p + \rho \mathbf{g} + \nabla \cdot \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \mathbf{F}_\sigma \tag{2}$$

Continuity Equation

$$\nabla \cdot \mathbf{u} = 0 \tag{3}$$

Although, the Navier-Stokes equations are solved on a regular Cartesian grid, the scalar transport equations are solved on a grid that can be adaptively refined as the solution proceeds in time. Here ϕ can be any generalized scalar such as the concentration or the temperature. For the accurate representation of the velocity field on this grid, a Cartesian grid which is sufficiently fine to resolve the hydrodynamics of multiphase flows is used. Generally the hydrodynamic boundary layer, can be resolved accurately with a coarser grid in comparison to mass transfer boundary layer. Also, solving a Poisson problem to meet the incompressibility criterion especially in case of multiphase flows with high density ratios is the most time consuming task amongst the whole solution time. Hence, solving only the scalar transport equations on an AMR grid leads to a decrease in computational effort compared to a situation where the hydrodynamics and scalar transport are both solved on the same grid. Finally, a Cartesian grid results in symmetric matrices forming the system of linear algebraic equations which can then be solved with robust matrix solvers with efficient vectorization and parallelization.

Convection

The unsteady convection part of the problem can be written as follows:

$$\frac{\partial\phi}{\partial t} + \nabla\cdot(\mathbf{u}\phi) = 0 \tag{4}$$

The above equation can be discretized using finite volume technique, resulting into

$$\phi^{n+1} = \phi^n + \frac{\Delta t}{\Delta V}\Big(\sum_{faces} {u_f}^{n+1}\phi_f^n h^2\Big) \tag{5}$$

The convection of a scalar can be numerically resolved using upwind class of methods which have good transport properties. For an adaptive grid based on a 2:1 balanced tree data structure, in combination with a first order method, two cases on a cell face can occur either both of the sides are at the same level or else there is a hanging node at the face resulting from level difference on either side (see figure 2). For a conventional Cartesian grid the convective flux at face center can be obtained for an upwind scheme, by the upwinded value of scalar at the face given as $\phi_f = \phi$ whereas, in case of face with a hanging node, the upwinded scalar in each subface, ϕ_{f1} and ϕ_{f2} is calculated first and is then averaged to obtain the upwinded scalar viz. $\phi_f = (\phi_{f1}+\phi_{f2})/2$. The flux can then be estimated assuming that the value of velocity at the face is known apriori.

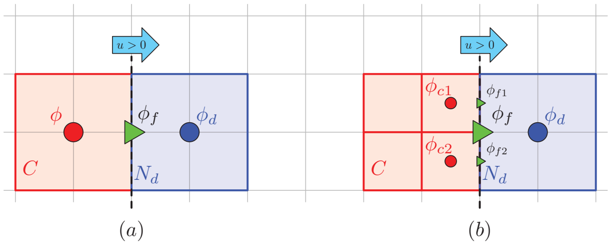


Figure 2: Upwind method on quadtree (a) both cells are at same level (b) neighbor is not a leaf cell.

Diffusion

The unsteady state diffusion part of the problem can be written as:

$$\frac{\partial\phi}{\partial t} = \nabla\cdot(D\nabla\phi) + S_\phi \tag{6}$$

When equation 6 is discretized using a semi-implicit scheme, equation 7 will be obtained.

$$\frac{\phi^{n+1}-\phi^n}{\Delta t}\Delta V = (1-\beta)\nabla\cdot(D^n\nabla\phi^n) + \beta\nabla\cdot(D^{n+1}\nabla\phi^{n+1}) + S_\phi^n \tag{7}$$

To calculate the diffusive flux at each face, a numerical estimation of the gradient of the scalar has to be determined at the cell face. For conventional Cartesian grid, this is normally done by central differencing, which is second order accurate for cells of same size. However, when this scheme is applied for cell face with a hanging node , the scheme is first order accurate instead of second order accurate. This type of differencing would result in schemes with inconsistent order of convergence, across the whole domain. To circumvent this problem, one needs to specifically look for schemes which are overall second order accurate. In this paper, a scheme is implemented which is overall second order accurate for the Laplacian operator in space. To verify the implementation of the Laplacian operator, the order of convergence is studied and compared with Gerris flow solver (Popinet, 2003), which uses the same discretization scheme for the pressure Poisson equation.

To enable the calculation of the transport via diffusion, the gradient at the face centers should be discretized. While this is trivial on a regular Cartesian grid, this is more difficult within the adaptive framework. In practice, only three cases are possible for the construction of face centered gradients for a cell C with a neighbor N_d in the direction d (see figure 3).

(I) N_d is at the same level and is a leaf cell

(II) N_d is at a lower level than the cell C, or double the size of cell C

(III) N_d is at a higher level than the cell C, or half the size of cell C

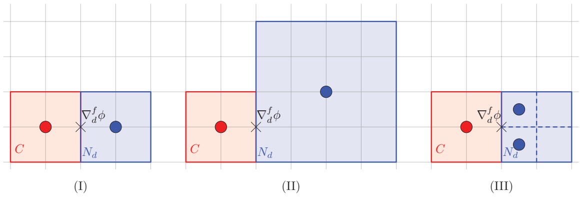


Figure 3: Different cases while calculating gradient at faces.

In case of (I) the stencil reduces down to a conventional Cartesian grid and thus one can write the simple centered difference discretization. In case of (II) a second order polynomial is fitted passing through the cell, C, the neighbor N_d and the opposite cell $\hat{\phi}_d$. Using the slope of the polynomial at the examined interface, the gradient at the face center can be calculated. Based on the configuration of the opposite cell, two subcases can arise due to 2:1 balanced tree: an undivided leaf cell at the same level of cell C or a parent cell with 4 leaf cells (8 in 3D). With the definition of ϕ_6 and ϕ_7 from figure 4, the expression for, respectively, an undivided and divided left neighbor are:

$$h\nabla_d^f\phi = -\frac{\phi}{3} - \frac{\hat{\phi}_d}{5} + \frac{8}{15}\phi_6 \tag{8}$$

$$h\nabla_d^f\phi = -\frac{2}{9}\phi - \frac{8}{27}\phi_7 + \frac{14}{27}\phi_6 \tag{9}$$

However as shown in figure 4, ϕ_6 should also be interpolated from the values of scalar in the cells neighboring the neighbor N_d, in the perpendicular direction to d ($\hat{N}_{\perp d}$ and $N_{\perp d}$) leading to more cases which needs to be included in an efficient fashion (Popinet, 2003). A sample case is shown in figure 4. Additional issues arise when extending to 3D because the two perpendicular that needs to be traversed to obtain the value of ϕ_6 are non-intersecting unlike the case in 2D. The details of this have been left out of the scope of this paper.

In case (III), N_d is at the same level but not a leaf cell. In this case the gradient at the face is constructed as the average of

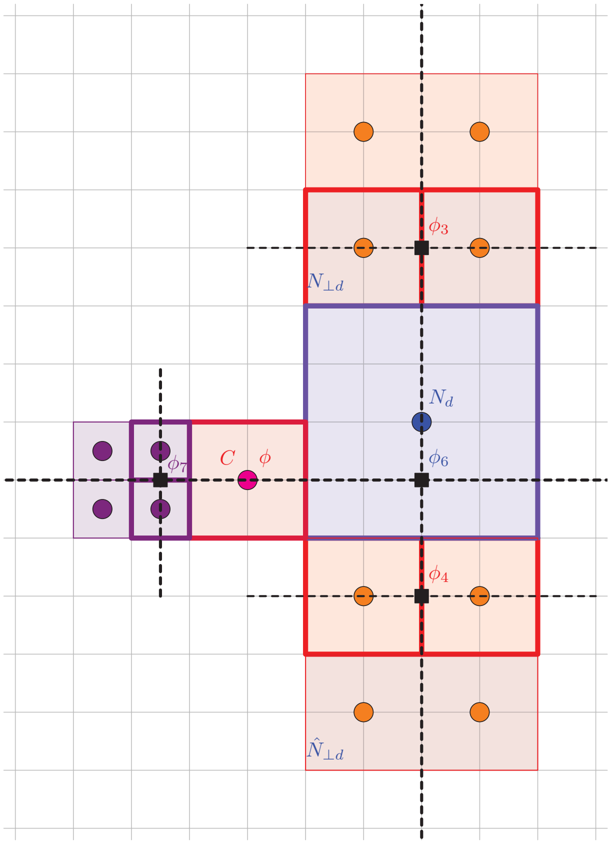


Figure 4: Sample stencil for face centered gradient calculation in case where $\hat{\phi}_d$, $\hat{N}_{\perp d}$ and $N_{\perp d}$ is not a leaf cell.

the gradients constructed from the children cells of N_d which share the face with the cell C. These gradients can then be computed by following the same scheme as case (II).
The resulting set of simultaneous linear equations resulting from this discretization of the Laplace operator is solved using robust matrix solver based on algebraic multigrid (AMG) method. Specifically, an open source library called HYPRE (Falgout and Yang, 2002) is used to solve the resultant equations using the boomerAMG module. The solver scales to 10^3 number of processors based on the MPI model and uses parallel graph partitioning methods for load balancing. The base framework used for implementation of the code is an open source library which provides efficient parallel tree data structure, called p4est (Burstedde *et al.*, 2011).

Boundary Conditions

At the domain boundary, both Dirichlet and Neumann boundary conditions can be applied with second order accuracy. For case (I) a ghost cell based approach is used to enforce Dirichlet boundary condition for the cell face at the boundary. For cases (II) and (III) where the value is interpolated in case N_d is a boundary cell, a ghost cell across the boundary face is created. This cell is of the same size of the neighbor cell N_d leaving only two possibilities in each case and thus resulting 4 additional stencils in 2D.

Staggered Velocity Field

The velocity needed to calculate the convective fluxes of the scalar is only known on the hydrodynamics mesh. The velocities are, however, needed on the smaller adaptive grid for the scalar transport. Hence, to obtain the values of velocity on the adaptive grid, the velocity should be transferred to the adaptive grid. Generally in an unstructured grid data structures, it is a common practice to encode all the possible state variables at a collocated grid, but the in house hydrodynamics code for gas-liquid interfaces has been developed on a staggered grid framework. The used staggered arrangement is beneficial for the upwind advection scheme used for the convection. To ensure a divergence free field a piecewise linear interpolation method is used to interpolate the velocity field (Roghair *et al.*, 2016). The interpolation can be visualized in figure 5. For example, the velocity field in the x-direction is calculated using the following set of equations:

$$u_{x_{i,j-\frac{1}{4}}} = u_{x_{i,j+\frac{1}{4}}} = \frac{u_{x_{i-\frac{1}{2},j}} + u_{x_{i+\frac{1}{2},j}}}{2} \tag{10}$$

$$u_{x_{i-\frac{1}{2},j-\frac{1}{4}}} = u_{x_{i-\frac{1}{2},j+\frac{1}{4}}} = u_{x_{i-\frac{1}{2},j}} \tag{11}$$

$$u_{x_{i+\frac{1}{2},j-\frac{1}{4}}} = u_{x_{i+\frac{1}{2},j+\frac{1}{4}}} = u_{x_{i+\frac{1}{2},j}} \tag{12}$$

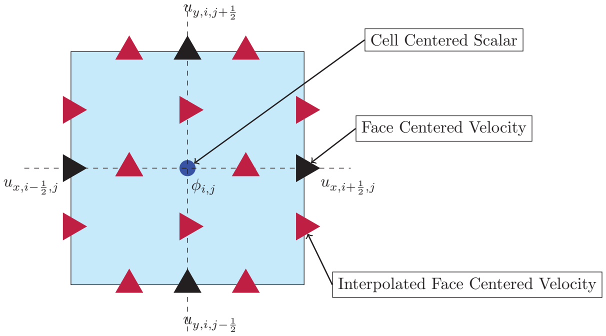


Figure 5: Staggered velocity interpolation.

Results

Convection

To test the implementation of the convection of scalar, a blob of a material is advected through a steady shear-free velocity field. To ensure the problem is a pure convection problem, diffusivity is set to zero. Typically the blob is given a conical distribution. The 2D transport equation for a scalar ϕ is solved over a unit square centered at the origin, domain boundaries being $x = \pm 0.5$ and $y = \pm 0.5$. The initial scalar distribution is prescribed to be zero except within a region of conical blob located with center at $(0,0)$ and radius $r = 0.1$, where the distribution of ϕ is given as $sin^2((\pi/2)(1-\frac{r}{0.1})$. The test was carried out for two values of velocities viz. $u = 1, v = 0$ and $u = 1, v = 1$. Figure 6 shows two snapshots for an inclined flow profile (a), and a parallel flow field (b). As with any other first order upwind method, this method suffers in principle from artificial diffusion, however here, the numerical diffusion is very small because of adaptive grids. In the test case with inclined flow field, a 94% reduction in grid requirements in adaptive framework (limited to maximum refinement of 10) was observed in comparison to a Cartesian grid at 10 levels of refinement($2^{2\times10} = 1048576$ grid cells) for the same numerical diffusion. The time step for advection is chosen in such a way that the CFL Number ($u_{max}\Delta t/\Delta x_{min}$) is always less than 0.5, where Δx_{min} is the cell size of the smallest quadrant in the domain and u_{max} is the maximum velocity in the domain.

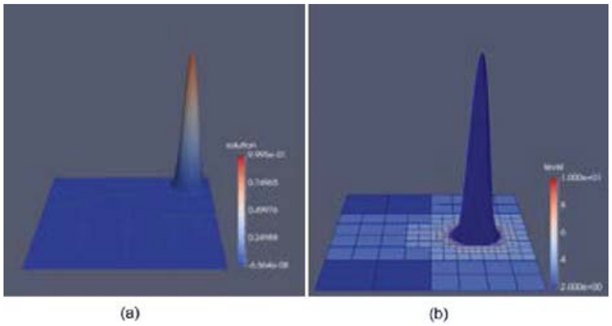


Figure 6: Convection of Scalar in (a) inclined flow field (solution), (b) parallel flow field (grid)

Diffusion

To test the implementation of the transport of scalar diffusion, the heat conduction equation on a rectangular slab of length, L and width, W is simulated. The slab is exposed to Dirichlet boundary conditions on all boundaries. In this test case, the top surface is maintained at 100 °C while the rest of the boundaries are maintained at 0 °C. The analytical solution of this steady state problem in terms of non dimensionalised temperature, θ is given by Equation 13.

$$\theta(x,y) = \frac{2}{\pi}\sum_{n=1}^{\infty}\frac{(-1)^{n+1}+1}{n}\sin\frac{n\pi x}{L}\,\frac{\sinh(n\pi y/L)}{\sinh(n\pi W/L)} \qquad (13)$$

Figure 7 shows the contour lines of the non-dimensionalised temperature and a comparison with this analytical solution. The L_1, L_2 and L_∞ error norms for the test problem were obtained to be 1.60e-05, 2.60e-4 and 0.022 respectively.

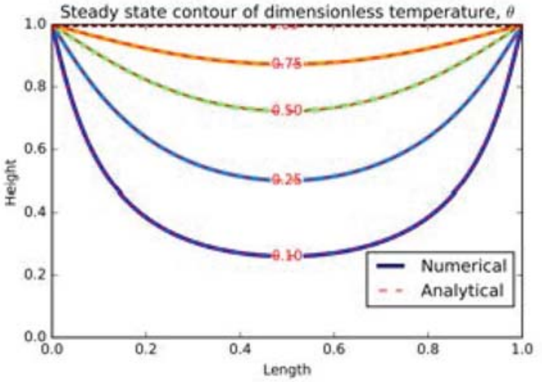


Figure 7: Heat Conduction in a Slab : Contour lines of dimensionless temperature in adaptive grid with analytical solution.

To ascertain the second order convergence of the Laplace operator, a convergence test is performed on Cartesian and adaptive grids. In this test a pressure Poisson equation is solved on a unit square domain centered around origin. The divergence of the intermediate velocity field is given by equation 14. When $k = l = 3$, the analytical solution is given by the pressure field of equation 15 using Neumann boundary conditions on all sides. κ in equation 15, is an arbitrary constant which in this case is the average value of the computed pressure over the entire domain.

$$\nabla\cdot U^{**}(x,y) = -\pi^2(k^2+l^2)\sin(\pi kx)\sin(\pi ly) \qquad (14)$$

$$\phi(x,y) = \sin(\pi kx)\sin(\pi ly)+\kappa \qquad (15)$$

The initial guess for the pressure field is a constant field. The problem is solved for two different domains: a domain at the L^{th} level of refinement (figure 8(a)) and the same domain with a circular patch of radius, $R = 0.25$ which is at $(L+2)^{th}$ level of refinement (figure 8(b)). To estimate the order of convergence, the problem is solved for different levels of refinement, L ranging from 3 to 10. For each simulation case, the volume (area in 2D) weighted norm of the error, $||ae||$ is shown in figure 9. The figure shows that the order of convergence is indeed two.

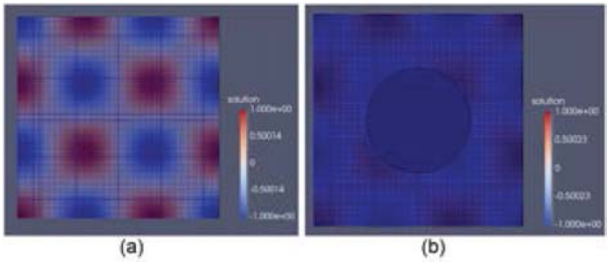
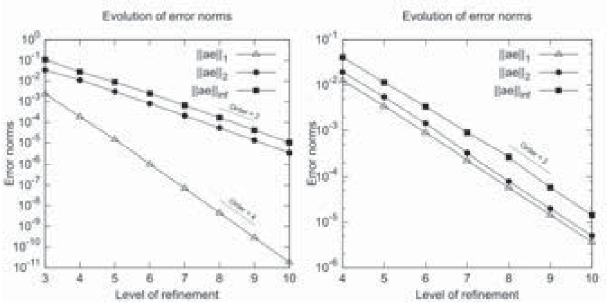


Figure 8: Domain for test case of Poisson equation (a) Cartesian grid (L = 7) (b) Grid used for evaluation of coarse/fine gradient operator (L = 7).



(a) **(b)**

Figure 9: Order of convergence test showing error norms for (a) Cartesian grid (b) Adaptive grid

Staggered Velocity Interpolation

For testing of staggered grid velocity interpolation, the grid was initiated with a divergence free analytical flow field at each face center. The velocity profile considered is given by Equation 16.

$$\boldsymbol{u}(x,y,z) = (0, -2sin^2(\pi y)sin(\pi z)cos(\pi z), \\ 2sin^2(\pi z)sin(\pi y)cos(\pi y)) \qquad (16)$$

The simulation is started at different level of refinements ($3^{rd} - 11^{th}$ level). Figure 10 shows the error with respect to the analytical solution after the interpolation of the velocity field. Because the initial level of refinement influences the error in the interpolated solution, the figure shows the error with respect to the level at which the original solution was created. Figure 11 shows a velocity vector plot for the refined grid at level 10 obtained from the initiated grid at level 8. The divergence has also been quantified in each of these refinement operations and was indeed found to be accurate to machine precision, i.e. 10^{-15} and 10^{-16}.

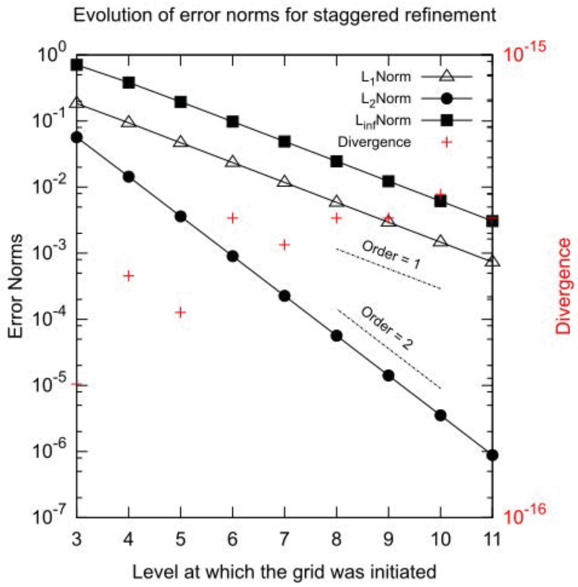


Figure 10: Error norms and divergence for staggered grid interpolation corresponding to different levels of refinement.

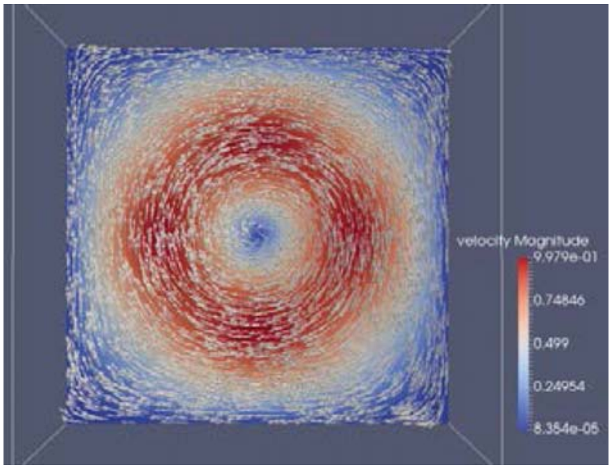


Figure 11: Velocity vector plot at 10^{th} level of refinement obtained from staggered grid interpolation of analytical flow field at 8^{th} level of refinement.

Concentration boundary layers

This test considers a stationary spherical bubble with radius R in a quiescent, zero gravity pool of liquid. The concentration within the sphere is initialized with a dimensionless concentration of 1. As, mass diffuses through the interface, a concentration boundary layer profile will evolve over time. The domain is refined at the 7^{th} level, while a zone near the bubble interface is refined upto the 12^{th} level (see figure 12). The bubble is simulated to be an infinite source by fixing the concentration inside the bubble to be 1. The concentration profiles are shown in figure 13.

For a diffusion equation written in spherical coordinates for a stationary bubble, an analytical solution can be obtained by imposing the boundary conditions: $c = c_0$ for $r = R$, $c = 0$ for $r = \infty$ and initial condition $c = 0$ for $r > R$. Because the simulation domain is finite, Neumann conditions are applied at the domain boundaries with a zero gradient. A diffusion coefficient $D = 10^{-6}$ m^2/s. was used. Figure 14 shows the concentration profile around a 4 mm sphere along the x-axis from the interface of the bubble and is compared with the exact solution. The interface is resolved with 8^{th} level of refinement with total cell count during 500^{th} time step to be 440784 which is 2.62 % that of a respective Cartesian mesh (16777216 cells).

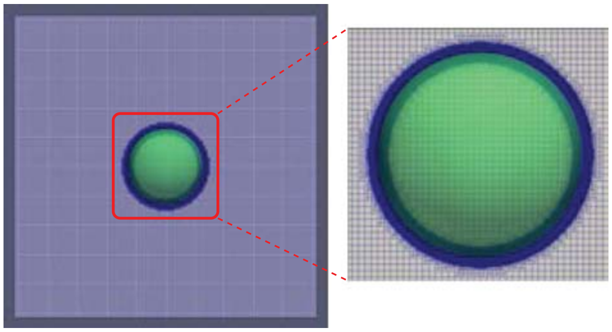


Figure 12: Initial grid for the species diffusion test.

$$\frac{c(r)}{H \cdot c_0} = \frac{R}{r}\left(1 - erf\left(\frac{r-R}{\sqrt{4Dt}}\right)\right) \tag{17}$$

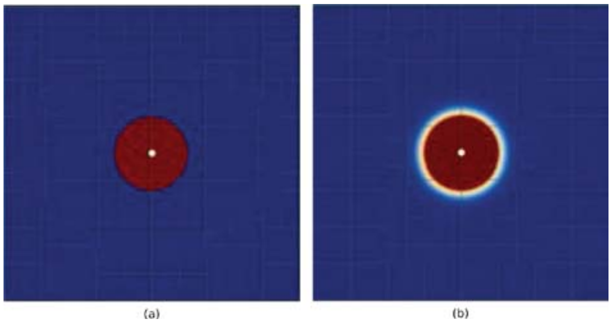


Figure 13: (a) Initial concentration profile inside stationary sphere. (b) Concentration profile around the sphere after 500 time steps.

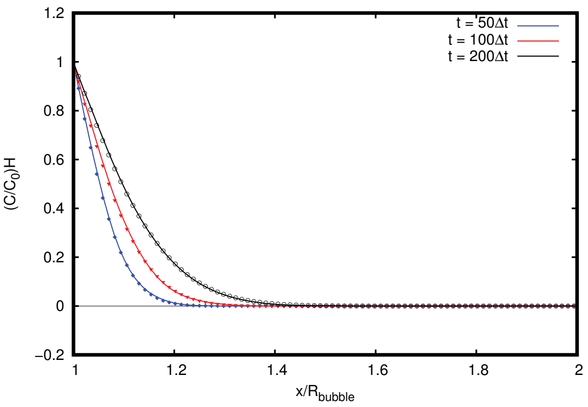


Figure 14: Dimensionless concentration profile in the vicinity of a stationary sphere, along the x-axis (lines) compared to the exact solution (markers).

CONCLUSION

In this paper, a methodology is proposed to solve heat and mass transfer equations on a different adaptive grid than the fixed Cartesian hydrodynamics grid. The details of the velocity access will be addressed in future work. The current work presents a robust way to set up a higher order accurate convection diffusion solver. The combined framework is tested for accuracy and order of convergence and furthermore, the method is found to be suitable for resolving boundary layers in bubbly flows. A methodology is also presented to encode staggered velocities on adaptive grids and to handle the grid adaptation without violating the divergence free criteria. In future research this methodology will be applied for the 3D simulation of bubbly flows with coupled heat and mass transfer.

ACKNOWLEDGMENT

This work is part of the Industrial Partnership Programme i36 Dense Bubbly Flows that is carried out under an agreement between Akzo Nobel Chemicals International B.V., DSM Innovation Center B.V., Sabic Global Technologies B.V., Shell Global Solutions B.V., Tata Steel Nederland Technology B.V. and the Netherlands Organisation for Scientific Research (NWO).

REFERENCES

ABOULHASANZADEH, B., THOMAS, S., TAEIBI-RAHNI, M. and TRYGGVASON, G. (2012). "Multi-scale computations of mass transfer from buoyant bubbles". *Chemical Engineering Science*, **75**, 456 – 467.

BURSTEDDE, C., WILCOX, L.C. and GHATTAS, O. (2011). "p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees". *SIAM Journal on Scientific Computing*, **33(3)**, 1103–1133.

DARMANA, D., DEEN, N.G. and KUIPERS, J.A.M. (2006). "Detailed 3d modeling of mass transfer processes in two-phase flows with dynamic interfaces". *Chemical Engineering & Technology*, **29(9)**, 1027–1033.

DEISING, D., MARSCHALL, H. and BOTHE, D. (2016). "A unified single-field model framework for volume-of-fluid simulations of interfacial species transfer applied to bubbly flows". *Chemical Engineering Science*, **139**, 173 – 195.

FALGOUT, R.D. and YANG, U.M. (2002). *hypre: A Library of High Performance Preconditioners*, 632–641. Springer Berlin Heidelberg, Berlin, Heidelberg.

GRÜNDING, D., FLECKENSTEIN, S. and BOTHE, D. (2016). "A subgrid-scale model for reactive concentration boundary layers for 3d mass transfer simulations with deformable fluid interfaces". *International Journal of Heat and Mass Transfer*, **101**, 476 – 487.

POPINET, S. (2003). "Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries". *Journal of Computational Physics*, **190(2)**, 572 – 600.

ROGHAIR, I., VAN SINT ANNALAND, M. and KUIPERS, J.A.M. (2016). "An improved front-tracking technique for the simulation of mass transfer in dense bubbly flows". *Chemical Engineering Science*, **152**, 351 – 369.