INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com/journal/nmf). DOI: 10.1002/fld.3724

Moving grid method for numerical simulation of stratified flows

S. Koltakov*,† and O. B. Fringer

Environmental Fluid Mechanics Laboratory, Stanford University, Stanford, CA 94305-4020, USA

SUMMARY

We develop a second-order accurate Navier–Stokes solver based on r-adaptivity of the underlying numerical discretization. The motion of the mesh is based on the fluid velocity field; however, certain adjustments to the Lagrangian velocities are introduced to maintain quality of the mesh. The adjustments are based on the variational approach of energy minimization to redistribute grid points closer to the areas of rapid solution variation. To quantify the numerical diffusion inherent to each method, we monitor changes in the background potential energy, computation of which is based on the density field. We demonstrate on a standing interfacial gravity wave simulation how using our method of grid evolution decreases the rate of increase of the background potential energy compared with using the same advection scheme on the stationary grid. To further highlight the benefit of the proposed moving grid method, we apply it to the nonhydrostatic lock-exchange flow where the evolution of the interface is more complex than in the standing wave test case. Naive grid evolution based on the fluid velocities in the lock-exchange flow leads to grid tangling as Kelvin–Helmholtz billows develop at the interface. This is remedied by grid refinement using the variational approach. Copyright © 2012 John Wiley & Sons, Ltd.

Received 12 March 2012; Revised 19 July 2012; Accepted 23 July 2012

KEY WORDS: moving grids; numerical viscosity; background potential energy; moving mesh PDE; interfacial gravity wave; lock-exchange flow

1. INTRODUCTION

Moving grid systems become increasingly important in a variety of engineering applications [1]. These systems are characterized by internal boundaries or interfaces separating regions with different physico-chemical properties. Across the interfaces, material properties and flow features can vary rapidly. These interfaces can undergo complex deformations making interface tracking very challenging. Accurate solution of transport equations relies on well-resolved internal and external moving boundaries. Finally, flow field discontinuities across interfaces suffer from limitations of inadequate grid resolution. This makes Eulerian description of fluid motion inadequate, because of the changing location and shape of interfaces. Ideally, one would like to have grid points travel with the flow, in the Lagrangian fashion, to keep interfaces well-resolved. However, this leads to poor grid quality when grid points overrun one another in regions of rapid flow variability. Some middle ground is clearly desired.

There are many adaptive grid methods in the literature [2, 3]. Most of them can be grouped into two distinctive sets: h-adaptive and r-adaptive methods. The first class of methods is based on adaptive mesh refinement type of strategy that dynamically adds points during the simulation in areas where an increased resolution is necessary and coarsens the mesh in the areas that fade out of interest. This is a viable strategy, although with its own complications. One problem is to find

*Correspondence to: S. Koltakov, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305-4020, USA.
†E-mail: koltakov@stanford.edu

exactly which cells to modify, that is, tracking precise location of the interface and the other is on the implementation side of efficient maintenance of variable-size data structures [4].

Another way to approach the problem of adaptive resolution is to employ r-adaptivity, in which the number of computational nodes is unchanged, but the grid points are dynamically moved during the simulation based on some strategy that concentrates the nodes in areas where they are most needed. Because the computational grid size is kept fixed, this method is easier to implement than the refinement method. Also, because the nodes are normally relocated to align mesh elements with the solution, the moving mesh method improves the simulation accuracy [5]. This is precisely the approach we take in this paper.

There are two distinct classes of moving mesh methods. In the first class, mesh movement is coupled with the physical differential equations. The moving finite element method of Miller *et al.* [6] and the moving finite difference method of Dorfi *et al.* [7] belong to this category. In the second class of methods, the mesh equations are decoupled from the original differential equations, requiring some form of conservative interpolation scheme to propagate the solution from the old to the new mesh [8]. In this paper, we explore the first approach, partly because adding r-adaptivity to an existing curvilinear grid solver can be straightforward and this allows researchers with static codes to swiftly gain the advantages of r-adaptivity by leveraging their existing infrastructures.

Deciding on an optimal moving grid strategy is a challenging task due to its non-uniqueness. The simplest approach is to move the grid based on fluid velocities in the Lagrangian fashion. Although this method might neutralize numerical diffusion associated with the nonlinear advection term, resulting computational grids typically suffer large distortions and possible tangling. One widely used approach to deal with mesh tangling is the arbitrary Lagrangian–Eulerian (ALE) technique [3], which relies on continuous rezoning and remapping between Lagrangian and Eulerian grids. Unfortunately, this process requires interpolation of geometry and flow variables [2]. Alternatively, the variational approach of Tang *et al.* [9, 10] provides an intuitive framework of moving grid nodes based on the so-called *monitor function* that can be constructed to resolve a certain physical quantity (e.g., density). To obtain the node distribution at each time step, an elliptic PDE that concentrates computational nodes in the areas of interest needs to be solved. Even though solving a 3D elliptic PDE is expensive, we will describe strategies to speed up the solution process while maintaining the benefits of adaptive mesh refinement.

Moving the grid during a simulation is not without its difficulties. Quite often, if not performed properly, the algorithm may violate mass conservation, an essential requirement for the pressure solver convergence and ultimate stability of the simulation. This problem is addressed by Chou *et al.* [11], where they describe the condition of *consistency with continuity* (CWC) as it applies to moving generalized curvilinear coordinates. We adopt their approach to defining grid velocities and contravariant fluxes used in the discretization of scalar and *geometric conservation laws* (GCL). Via this GCL, the grid motion is incorporated into the original system of conservation laws.

In addition to increased accuracy, adaptive treatment of interfaces yields lower amounts of numerical diffusion that is an imperative if realistic resolution of interfacial gravity wave dynamics is desired [11]. We measure numerical artifacts with the help of the *background potential energy* [12], growth of which implies numerical diffusion and decay implies either anti-diffusion or the development of non-monotonicity.

The importance of grid adaptivity is highlighted in [13] on the example of classic laboratory-scale lock-exchange problem, which is one of test cases in this paper. The gravity current front speed is highly dependent on the amount of numerical diffusion added by the solution method, thus minimizing that it is of high importance in accurate simulations of complex physical processes. Results obtained with the moving grid method compare favorably with the published results in [14, 15]. The main contribution of this paper is the synthesis of existing grid adaptation methods for optimal simulation of stratified flows.

The next section describes the hydrodynamics of the problem we are solving and its numerical discretization. Section 3 is dedicated to the background potential energy and its connection to numerical properties of scalar advection schemes. In Section 4, we discuss the added complexity

DOI: 10.1002/fld

of preserving mass conservation in the moving grid case. Section 5 introduces two strategies of moving the computational grid as well as the hybrid approach that combines them; it also describes the variational approach to grid adaptivity. Finally, in Section 6, the moving grid method is applied to two test cases: the interfacial gravity wave and the non-hydrostatic lock exchange.

2. GOVERNING EQUATIONS AND NUMERICAL METHOD

2.1. Governing equations

We discretize the Navier–Stokes equations with the Boussinesq approximation along with the scalar transport equation for density ρ,

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) = \mathcal{S}_i, \tag{1}$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0, \tag{2}$$

subject to the continuity equation

$$\frac{\partial u_j}{\partial x_j} = 0, \tag{3}$$

where the Einstein summation convention is assumed, $j = 1, 2, 3$, and u_j is the jth-component of the Cartesian velocity. $\mathcal{S}_i$ represents effects of pressure, viscous stresses, and body forces, as in

$$\mathcal{S}_i = -\frac{1}{\rho_0}\frac{\partial p}{\partial x_i} + \nu\frac{\partial^2 u_i}{\partial x_j \partial x_i} - \frac{g}{\rho_0}(\rho - \rho_0)\delta_{i3}, \tag{4}$$

where ν is the constant kinematic viscosity, ρ_0 is the reference density, and g is gravitational acceleration. Equation (2) assumes no molecular diffusivity for density, which is a good approximation when the Schmidt number is large, such as that in a salt-stratified flow.

In the case of moving grid simulation, we have a transformation from the fixed Cartesian coordinate system (x_1, x_2, x_3) into the moving curvilinear coordinate system $(\xi_1(x_1, x_2, x_3, t), \xi_2(x_1, x_2, x_3, t), \xi_3(x_1, x_2, x_3, t))$. The transformed system of equations, corresponding to (1-3), in strong-conservation-law form is given by

$$\frac{\partial}{\partial t}(J^{-1}u_i) + \frac{\partial}{\partial \xi_m}(u_i U_m) - \frac{\partial}{\partial \xi_m}(u_i U_{g,m}) = S_i, \tag{5}$$

$$\frac{\partial}{\partial t}(J^{-1}\rho) + \frac{\partial}{\partial \xi_m}(\rho U_m) - \frac{\partial}{\partial \xi_m}(\rho U_{g,m}) = 0, \tag{6}$$

$$\frac{\partial U_m}{\partial \xi_m} = 0, \tag{7}$$

where J is the Jacobian of transformation, and U_m and $U_{g,m}$ are contravariant volume fluxes corresponding to fluid and grid node velocities, respectively, viz.

$$U_m = J^{-1}\frac{\partial \xi_m}{\partial x_j}u_j,\ U_{g,m} = J^{-1}\frac{\partial \xi_m}{\partial x_j}u_{g,j}, \tag{8}$$

DOI: 10.1002/fld

and where the Cartesian velocity is given by $u_{g,j} = dx_{g,j}/dt$. The right-hand side of (5) is given by

$$S_i = -\frac{1}{\rho_0}\frac{\partial}{\partial \xi_m}\left(J^{-1}\frac{\partial \xi_m}{\partial x_i}p\right) + \nu\frac{\partial}{\partial \xi_m}\left(G^{mn}\frac{\partial u_i}{\partial \xi_n}\right) + J^{-1}\frac{g}{\rho_0}(\rho-\rho_0)\frac{\partial \xi_m}{\partial x_i}\frac{\partial x_3}{\partial \xi_m}, \tag{9}$$

where

$$J^{-1} = det\left(\frac{\partial x_i}{\partial \xi_m}\right),\ G^{mn} = J^{-1}\frac{\partial \xi_m}{\partial x_i}\frac{\partial \xi_n}{\partial x_j}. \tag{10}$$

When there is no density variation, Equation (6) reduces to

$$\frac{\partial J^{-1}}{\partial t} + \frac{\partial U_m}{\partial \xi_m} - \frac{\partial U_{g,m}}{\partial \xi_m} = 0. \tag{11}$$

Substituting the continuity equation in generalized curvilinear coordinates (7) into (11), we obtain the GCL for updating the Jacobian of transformation on the basis of the grid velocity [11].

$$\frac{\partial}{\partial t}(J^{-1}) = \frac{\partial U_{g,m}}{\partial \xi_m}. \tag{12}$$

2.2. Numerical discretization

Following [16], we discretize our Equations (5–7) on a non-staggered grid, with Cartesian velocities and pressure defined at cell centers, whereas contravariant volume fluxes are defined at the faces. The momentum equations are integrated with second-order time accuracy, with the Adams–Bashforth method for the explicit terms and the Crank–Nicolson for the implicit terms. Convective and off-diagonal viscous terms are treated explicitly, whereas the diagonal viscous terms are treated implicitly to eliminate the viscous stability limit. The convective terms in (5) and (6) are discretized with the QUICK [17] and SHARP [18] schemes, respectively. The fractional step method is employed following [16], and the pressure Poisson equation is solved with the multigrid method. The simulation code was written in the C++ programming language and parallelized with the message passing interface similar to [19].

3. ADVECTION AND NUMERICAL DIFFUSION ON ADAPTIVE AND FIXED GRIDS

3.1. Advection schemes

Selecting the right advection scheme becomes critical in discretization of the transport Equation (2). The discrete version of (2) after the application of the finite-volume method is

$$\rho^{n+1}_{i,j,k} = \rho^{n}_{i,j,k} + \frac{\Delta t}{\Delta x}(u_w\rho_w - u_e\rho_e) + \frac{\Delta t}{\Delta y}(v_s\rho_s - v_n\rho_n) + \frac{\Delta t}{\Delta z}(w_f\rho_f - w_b\rho_b), \tag{13}$$

where $\rho_{i,j,k}$ is density averaged over cell (i, j, k) and the subscripts denote faces (East$(i + 1/2)$, West$(i - 1/2)$, North$(j + 1/2)$, South$(j - 1/2)$, Front$(k + 1/2)$, and Back$(k - 1/2)$), over which the fluxes are averaged in space and time [20]. By setting $\rho_{w(i)} = \rho_{e(i-1)}$, $\rho_{s(j)} = \rho_{n(j-1)}$, $\rho_{f(k)} = \rho_{b(k-1)}$ and computing just three flux-face values per cell, conservation of mass is automatically guaranteed.

The simplest approach to advection is the first-order upwinding, for which, assuming $u_e > 0$, $v_n > 0$, and $w_b > 0$, density values at faces are given by

$$\rho_e = \rho_n = \rho_b = \rho_{i,j,k}. \tag{14}$$

This method is highly diffusive [21] and is first-order accurate but guarantees monotonicity and is simple to implement. In Fringer *et al.* [22], 12 standard and two novel advection schemes were studied in terms of their diffusive or anti-diffusive effects. First-order upwind was found to be the most diffusive, and the SHARP scheme [18] was shown to produce the best results overall.

DOI: 10.1002/fld

Originally designed to guarantee monotonicity for steady flows, the SHARP scheme is also capable of handing highly unsteady flows when coupled with the second-order Adams–Bashforth scheme [16], causing slight overshoot behavior in the presence of sharp fronts.

Assuming $u_e > 0$, the density face value ρ_e using SHARP is given by

$$\rho_e = \frac{1}{2}(\rho_{i,j,k} + \rho_{i+1,j,k}) - CF\left(\tilde{\rho}_{i,j,k}\right)(\rho_{i-1,j,k} - 2\rho_{i,j,k} + \rho_{i+1,j,k}), \tag{15}$$

where $CF\left(\tilde{\rho}_{i,j,k}\right)$ takes on one of the seven algebraic forms depending on the value of $\tilde{\rho}_{i,j,k}$, which is the normalized adjacent upstream density value based on its two adjacent neighbors and the direction of upwinding.

These two advection schemes, being the extreme cases in terms of numerical diffusion, will be extensively studied in combination with the stationary and moving grid methods applied to the interfacial sloshing wave test case in this paper.

3.2. Background potential energy

The numerical diffusion or anti-diffusion of the density field resulting from the numerical solution method can be quantified with the evolution of the background potential energy. According to [12], the total potential energy can be split into its available and background components

$$E_p = E_a + E_b, \tag{16}$$

where E_a is the energy available to be converted into motion, and E_b is the potential energy in its background state. The latter is defined as a volume integral over the domain, viz.

$$E_b = g\int_V \rho z^*\left(\vec{x},t\right)dV, \tag{17}$$

where $z^*\left(\vec{x},t\right)$ is the height of the fluid parcel with density ρ in its background state. An increase in background potential energy implies numerical diffusion in the absence of molecular diffusion of density, whereas a decrease is an indication of either numerical anti-diffusion or the development of non-monotonicity.

A convenient way to demonstrate the relative increase in the background potential energy is to plot the relative change in the background potential energy

$$\Delta E_b^*(t) = \frac{E_b(t) - E_b(0)}{E_a(0)}, \tag{18}$$

where the superscript * signifies nondimensional quantity, $E_b(0)$ is the initial background potential energy, and $E_a(0)$ is the initial available potential energy, as defined in Equation (16).

3.3. Computation of background potential energy

In a two-dimensional domain (the analysis is easily extended to three dimensions) with a discrete density distribution given by $\rho_{i,k}$ and a cell volume distribution $\delta V_{i,k}$; the total potential energy of the domain E_p can be discretely evaluated as

$$E_p = g\sum_{i,k=1}^{N_i,N_k}\rho_{i,k}z_{i,k}\delta V_{i,k}, \tag{19}$$

where N_i and N_k are the total number of grid points used to discretize the domain in each of two dimensions. The height of the centroid of cell (i,k) is denoted by $z_{i,k}$.

A discrete equivalent of the background potential energy is given by

$$E_b = g\sum_{n=1}^{N_i\times N_k}\rho_n^* z_n^*\delta V_n^*, \tag{20}$$

DOI: 10.1002/fld

where ρ_n^* is the sorted equivalent of the two-dimensional density field $\rho_{i,k}$ that was arranged in a descending order. δV_n^* corresponds to the volume of the cell with density ρ_n^*. The height of the sorted density field is computed with

$$z_{n+1}^* = z_n^* + \frac{\delta V_{n+1}^*}{A\left(z_n^*\right)} \tag{21}$$

for $n = 1, \ldots, N_i \times N_k$. $A\left(z_n^*\right)$, the planform area of the cell, equals to δx_i in the two-dimensional case is described here. For $n = 1$, the height of the lowest cell is given by

$$z_1^* = z_0 + \frac{\delta V_1^*}{2A(z_0)} \tag{22}$$

which corresponds to the vertical center of the cell. z_0 is the vertical coordinate of the bottom of the domain.

Computation of the background potential energy requires implementation of a sorting algorithm to obtain the sorted density field ρ_n^*. In the serial version, we employ the standard Quicksort algorithm. However, on a parallel system, this standard sorting strategy is not applicable because of the distributed nature of the density array. In order to minimize communication and keep the local density arrays balanced, we employ parallel sorting by regular sampling (PSRS), developed by Li *et al.* [23]. Another advantage of PSRS over Hyperquicksort, a parallel analog of Quicksort, is that it does not require the number of processors to be a power of 2.

The basic idea behind the PSRS algorithm that sorts n elements on p processors consists of four phases. In the first phase, each process performs a sequential quick sort of its share of the elements. In the second phase, one process gathers data items from counterpart processes with local indices $0, n/p^2, 2n/p^2, \ldots, (p-1)(n/p^2)$ and sorts the combined list of samples. From the sorted list, it picks $p-1$ items with indices $p + \lfloor p/2 \rfloor - 1, 2p + \lfloor p/2 \rfloor - 1, \ldots, (p-1)p + \lfloor p/2 \rfloor$ to be used as pivots and broadcasts them to the counterpart processes. In the third phase, each process i uses the pivots to dissect its list into p partitions. Then it keeps the ith partition and sends the jth partition to process j, for all $j \neq i$. In the forth and final phase of PSRS, each process merges its p partitions into a single list. At this point, the elements are sorted and are distributed in increasing order among all processes in a disjoint manner.

Li *et al.* [23] proved that the largest number of elements any process might have to merge is less than twice its share of the elements, that is, $2n/p$. In practice though, the largest partition size normally deviates by only a few percent from the average partition size. The computational complexity of the PSRS algorithm is $O((n/p)(\log n + \log p))$. Because $n \gg p$, the communication time is dominated by sublist exchange among processes in phase 3.

4. CONSISTENCY WITH CONTINUITY

The concept of CWC has been discussed by various authors [20, 21, 24, 25] and is an essential requirement for mass and momentum conservation. A convenient definition of CWC is given in [25]: *a discretization of the advection equation is consistent with continuity if given a spatially uniform scalar field as an initial datum and a general flow field, the discretized scalar advection equation reduces to the discretized continuity equation.* For stationary grids, CWC implies that the same discrete divergence operator must be used for both the continuity and the scalar transport equations. Thus, it is straightforward to ensure CWC for stationary grids. However, in the case of moving grids, special care must be taken, namely, an additional conservation equation must be solved simultaneously with the mass, momentum, and energy conservation equations. This additional Equation (12), preventing the formation of artificial mass sources, is referred to as the GCL by [26] or *space conservation law* by [27].

4.1. Geometric conservation law

To guarantee geometric conservation in the discrete sense, Equation (12) must be appropriately discretized. A naive discretization will most likely lead to excessive mass creation induced by the

DOI: 10.1002/fld

non-conservative property of the moving grid, as discussed in [11]. Also, there is an additional complexity of moving the grid in multiple dimensions because of unaccounted parts of swept volume by the cell faces that arises when each dimension is treated independently. Alternatively, if coordinate directions are coupled by the operator splitting scheme, the CWC condition is not satisfied, as demonstrated by Leonard *et al.* [20]. Fortunately, in the case when grid node positions are known before and after the time step, it is possible to construct the finite-volume discretization of Equation (12) that ensures CWC, viz.

$$J^{-1}\Big|_{i,j,k}^{n+1} = J^{-1}\Big|_{i,j,k}^{n} + \Delta t \left(U_{g,i+\frac{1}{2},j,k}^{n+\frac{1}{2},n} - U_{g,i-\frac{1}{2},j,k}^{n+\frac{1}{2},n} + V_{g,i,j+\frac{1}{2},k}^{n+\frac{1}{2},n} - V_{g,i,j-\frac{1}{2},k}^{n+\frac{1}{2},n} + W_{g,i,j,k+\frac{1}{2}}^{n+\frac{1}{2},n} - W_{g,i,j,k-\frac{1}{2}}^{n+\frac{1}{2},n} \right), \tag{23}$$

where the superscript $n + 1/2$ on the contravariant volume fluxes refers to the intermediate values of the metric quantities that are used to evaluate them from grid velocities. For example, $U_{g,i\pm\frac{1}{2},j,k}^{n+\frac{1}{2},n}$ is given by

$$U_{g,i\pm\frac{1}{2},j,k}^{n+\frac{1}{2},n} = \left(\begin{vmatrix} \frac{\partial x_2}{\partial \xi_2} & \frac{\partial x_2}{\partial \xi_3} \\ \frac{\partial x_3}{\partial \xi_2} & \frac{\partial x_3}{\partial \xi_3} \end{vmatrix}^{n+\frac{1}{2}} u_g^n + \begin{vmatrix} \frac{\partial x_3}{\partial \xi_2} & \frac{\partial x_3}{\partial \xi_3} \\ \frac{\partial x_1}{\partial \xi_2} & \frac{\partial x_1}{\partial \xi_3} \end{vmatrix}^{n+\frac{1}{2}} v_g^n + \begin{vmatrix} \frac{\partial x_1}{\partial \xi_2} & \frac{\partial x_1}{\partial \xi_3} \\ \frac{\partial x_2}{\partial \xi_2} & \frac{\partial x_2}{\partial \xi_3} \end{vmatrix}^{n+\frac{1}{2}} w_g^n \right)_{i\pm\frac{1}{2},j,k} \tag{24}$$

For details on why Equation (23) ensures geometric conservation both locally and globally, see [11].

4.2. Consistent discretization of scalar transport and grid node velocities

Now that we have the appropriate spatial discretization of the GCL, it is important to apply the same temporal discretization technique to both the geometric (12) and scalar mass (6) conservation laws to satisfy the CWC condition. Failure to do so will induce significant errors, because of the excessive mass creation, that grow proportionally with the time step size [28].

Another caveat is the definition of the grid node velocities that are based on the known positions of grid nodes at each time step. Generally speaking, grid node velocities, constrained only by the node locations and the CWC condition, are not unique. For example, Demirdžić *et al.* [28] provides an alternative definition of grid node velocities and contravariant volume fluxes that ensures geometric conservation.

The second subtlety with grid velocities is their influence on the overall temporal time accuracy. Assume that the grid velocities are discretized with the first-order differences, on the basis of just two sets of node locations, but the scalar transport and GCL equations are discretized to a higher order in time. The overall time accuracy of the scheme will then be limited by the first-order discretization of the grid velocities [11]. Therefore, to maintain the second-order temporal accuracy of our moving grid method, we use a second-order accurate approximation for the grid velocity

$$\mathbf{u_g}|_{i,j,k}^{n} = \frac{2}{3}\frac{\mathbf{x}_{i,j,k}^{n+1} - \mathbf{x}_{i,j,k}^{n}}{\Delta t} + \frac{1}{3}\mathbf{u_g}|_{i,j,k}^{n-1} + O(\Delta t^2) \tag{25}$$

on the basis of the AB2 method.

5. MOVING GRIDS

In order to reduce numerical diffusion generated by the advection scheme, we propose moving the computational mesh dynamically in a way that will minimize the significance of the advection term. From Equations (5) and (6), one can see that when $U_m = U_{g,m}$, the equations are effectively written in a coordinate system that eliminates advection. One option is to move the mesh with the fluid flow in a Lagrangian way. Although a viable option, this unfortunately leads to grid tangling as described in Section 5.1. To deal with those issues, we propose adjusting the computational grid

DOI: 10.1002/fld

with the variational moving mesh approach during the course of the simulation, as will be described Section 5.2. Another extreme case is to move the grid based on the variational approach alone. This way, the interface is profusely enveloped by the grid points, and the errors associated with the advection term should also decrease. Unfortunately, this particular moving mesh strategy also has its flaws as described in the following.

5.1. Arbitrary Lagrangian–Eulerian computational mesh based on fluid flow

Moving the computational mesh with the fluid flow alone, what is generally referred to as the Lagrangian moving mesh, is an attractive strategy for several reasons. For once, it completely eliminates the advection term in the momentum and scalar equations, if $\mathbf{U}_{\mathbf{grid}} = \mathbf{U}_{\mathbf{fluid}}$. Discretization of this term is well-known for its contribution to the numerical diffusion of the solution method. Thus, eliminating it should address the problem. On the negative side lies the effect of wave overturning in the short run that can deteriorate or sometimes completely destroy the quality of the underlying computational grid. Figure 1(a) depicts a computational grid moving with the fluid flow in the Lagrangian fashion. The grid nodes of cells located at the interface are pushed in different directions by the fluid flow promptly destroying the mesh. Another disadvantage of the Lagrangian moving mesh approach is the negative effect of the time-averaged mean flow that corrupts the computational grid in the long run, as demonstrated in Figure 2(b). Initially, the fluid interface was uniformly resolved by the mesh at $t = T/4$, as depicted in Figure 2(a). However, as the simulation progresses, after two periods, the middle part of the interface loses its original resolution, whereas the left and right interfacial boundaries attract more nodes as a result of the time-averaged flow (the Stokes' drift).

One way to drive the mesh with the fluid flow without destroying mesh quality is to constrain grid motion to only the vertical component. This way, each vertical column will stay unchanged throughout the simulation; however, the cells belonging to it will adjust vertically, resolving the interface that happens to lie across the column. In some cases, this strategy may be what is needed to resolve the interface and minimize the numerical diffusion associated with the advection term. One example where pure vertical mesh motion could be sufficient are ocean flows, where the relative displacement of the interface is negligible compared with the lateral dimensions of the problem. On the contrary, resolving the breaking wave on the ocean shoal will be problematic with this method due to the overturning nature of the flow. Constraining the mesh motion to the vertical will not break the grid nor will it accurately resolve the interface that will be stacking on itself. In that case, an alternative grid moving strategy is in order.

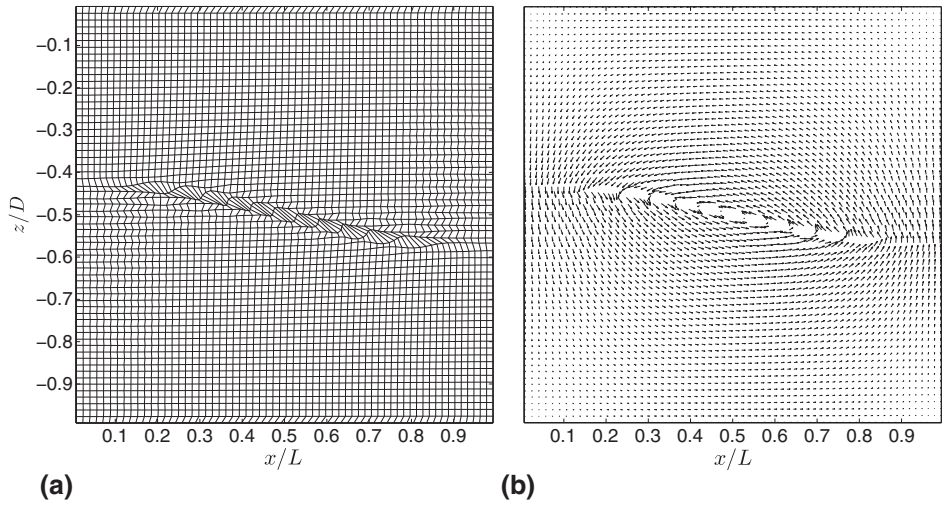

Figure 1. Instantaneous grid deterioration as a result of the Lagrangian moving mesh strategy for the sloshing gravity wave: (a) computational grid; (b) velocity vectors.

DOI: 10.1002/fld

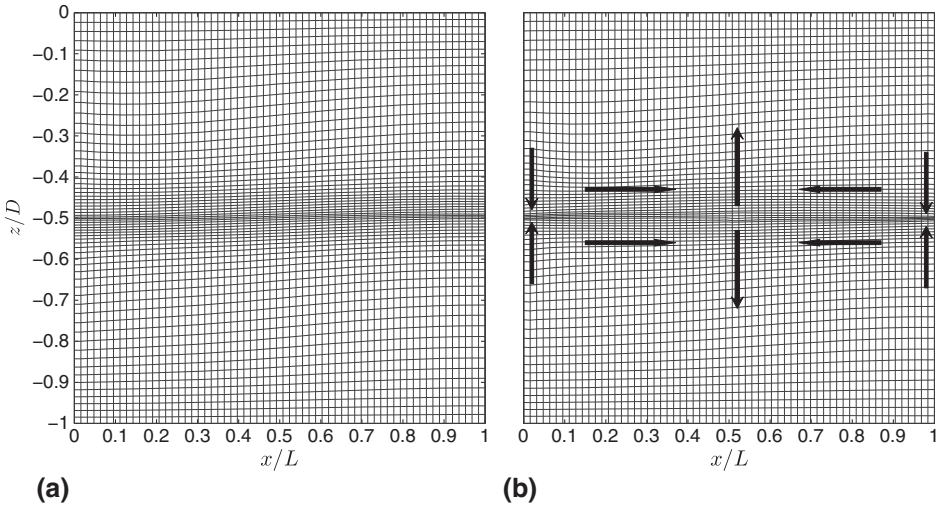

Figure 2. Long-term grid deterioration with the time-averaged mean fluid flow for the sloshing gravity wave. Computational grid at: (a) $t = T/4$; (b) $t = 9T/4$. Arrows indicate the direction of the time-averaged mean flow.

5.2. *Variational moving mesh approach*

Using the variational approach to mesh generation, in the spirit of Tang *et al.* [9] for 1D/2D and [10] for 3D, we can redistribute grid points from any initial configuration to concentrate them in regions of large solution variation. Once the new grid node positions are known and Jacobians of transformation are recalculated with (12), we proceed to solving our system of hydrodynamic PDEs (5–7) on the updated grid. The following outlines the procedure of grid redistribution based on the latest available density profile.

In the conventional variational approach, the desired grid node distribution is achieved by minimizing the functional of the following form:

$$E\left[\vec{\xi}\right] = \frac{1}{2}\int_{\Omega}\left(\frac{\partial x_i}{\partial \xi_n}^{T} M_{nm}^{-1}\frac{\partial x_i}{\partial \xi_m}\right) d\Omega, \tag{26}$$

where M is referred to as the *monitor function* and is a 3×3 positive definite matrix. The Euler–Lagrange equation corresponding to functional (26) determines the new distribution of the mesh

$$\frac{\partial}{\partial \xi_m}\left(M_{mn}^{-1}\frac{\partial \vec{x}}{\partial \xi_n}\right) = 0. \tag{27}$$

Following [9] to circumvent computational difficulties involved in interchanging the dependent and independent variables, we replace Equation (27) by

$$\frac{\partial}{\partial x_j}\left(M_{ji}^{-1}\frac{\partial \vec{\xi}}{\partial x_i}\right) = 0. \tag{28}$$

Equation (28) is much simpler to solve in order to obtain the desirable grid node distribution. To better resolve areas of high density variation, we base the monitor function M on the density gradient, viz.

$$M = \sqrt{1 + \alpha\left(\frac{\partial \rho}{\partial x_i}\right)^2}\, I, \tag{29}$$

where α is a tunable parameter, ρ is the density field, and I is the identity matrix. This monitor function allows clustering of grid points close to the fluid interface, where density experiences a

DOI: 10.1002/fld

large gradient. Because ρ is defined on the computational grid, its derivatives after application of the chain rule are given by

$$\frac{\partial \rho}{\partial x_i} = \frac{\partial \rho}{\partial \xi_m} \frac{\partial \xi_m}{\partial x_i}. \tag{30}$$

In order to ensure convergence of (28) with respect to mesh refinement, we discretize the gradients in the computational domain $\left(\frac{\partial \rho}{\partial \xi_m}\right)$ with fourth-order accuracy using the standard five-point stencil finite difference approximation of the first derivative. Because we use a scalar-type monitor function (29) for simplicity, Equation (28) becomes

$$\partial_{x_i}\left(\omega \partial_{x_i} \vec{\xi}\right) = 0, \tag{31}$$

where $\omega = \sqrt{1 + \alpha\left(\frac{\partial \rho}{\partial x_i}\right)^2}$. Equation (31) is a nonlinear elliptic PDE, which is solved with a Gauss–Seidel iteration procedure to obtain solution at iteration $\nu + 1$ from iteration ν,

$$\begin{aligned} 0 = {} & a_{i+\frac{1}{2},j,k}\left(\vec{\xi}^{[\nu]}_{i+1,j,k} - \vec{\xi}^{[\nu+1]}_{i,j,k}\right) - a_{i-\frac{1}{2},j,k}\nabla_i \vec{\xi}^{[\nu+1]}_{i,j,k} \\ & + b_{i,j+\frac{1}{2},k}\left(\vec{\xi}^{[\nu]}_{i,j+1,k} - \vec{\xi}^{[\nu+1]}_{i,j,k}\right) - b_{i,j-\frac{1}{2},k}\nabla_j \vec{\xi}^{[\nu+1]}_{i,j,k} \\ & + c_{i,j,k+\frac{1}{2}}\left(\vec{\xi}^{[\nu]}_{i,j,k+1} - \vec{\xi}^{[\nu+1]}_{i,j,k}\right) - c_{i,j,k-\frac{1}{2}}\nabla_k \vec{\xi}^{[\nu+1]}_{i,j,k}, \end{aligned} \tag{32}$$

where $\nabla_{i/j/k}$ denotes the backward difference operator in the subscript direction, and

$$\begin{aligned} & a_{i\pm\frac{1}{2},j,k} = \frac{1}{4}\sum_{p,q=\pm\frac{1}{2}} \omega_{i\pm\frac{1}{2},j+p,k+q}, \quad b_{i,j\pm\frac{1}{2},k} = \frac{1}{4}\sum_{p,q=\pm\frac{1}{2}} \omega_{i+p,j\pm\frac{1}{2},k+q}, \\ & c_{i,j,k\pm\frac{1}{2}} = \frac{1}{4}\sum_{p,q=\pm\frac{1}{2}} \omega_{i+p,j+q,k\pm\frac{1}{2}}. \end{aligned} \tag{33}$$

Fortunately, the physical PDE solutions do not change drastically at consecutive time steps and thus (32) to converges in a few iterations if the initial guess is the grid at the old time step. After the iteration scheme (32) converges, the monitor function is recalculated from the new density field $\rho^{[\nu+1]}_{i,j,k}$, which is interpolated from the old to the new grid after each iteration. The iteration scheme (32) is then repeated with the updated monitor function several times, whereas the incremental motion of computational nodes is exceeding a pre-specified threshold (i.e., $||\vec{\xi}^{[\nu+1]} - \vec{\xi}^{[\nu]}||_2 > \epsilon$). In our experience, only a few (i.e., 3–5) outer iterations were sufficient to stabilize computational nodes to within 5% length of the corresponding grid edge.

In a parallel implementation, in which the number of nodes per processor remains constant, it is necessary to interpolate the density field at grid nodes near inter-processor boundaries. This requires large enough halo regions so that interpolating density at the new grid node locations does not require inter-processor communication. The width of this halo region depends on the distance traveled by grid nodes in one time step. From our experience, keeping the halo region covering the adjacent processors was sufficient as in each time step the interface was staying safely within domain bounds of adjacent processors. Also, it is advisable to use some temporal or spatial smoothing on the monitor function to obtain smoother meshes. One of the reasons for using smoothing is to avoid very singular meshes and/or large approximation error around the stiff solution areas.

DOI: 10.1002/fld

We apply the standard low pass filter, sometimes multiple times as will be indicated in Section 6, after we update the monitor function with the new density field at each outer iteration, viz.

$$\begin{aligned}\tilde{\omega}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = {} & \frac{1}{8}\omega_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \\ & + \frac{1}{16}\Big(\omega_{i+\frac{3}{2},j+\frac{1}{2},k+\frac{1}{2}} + \omega_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} + \omega_{i+\frac{1}{2},j+\frac{3}{2},k+\frac{1}{2}} \\ & \qquad + \omega_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} + \omega_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{3}{2}} + \omega_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\Big) \\ & + \frac{1}{32}\Big(\omega_{i+\frac{3}{2},j+\frac{3}{2},k+\frac{1}{2}} + \omega_{i+\frac{3}{2},j+\frac{1}{2},k+\frac{3}{2}} + \omega_{i+\frac{1}{2},j+\frac{3}{2},k+\frac{3}{2}} + \omega_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} \\ & \qquad + \omega_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + \omega_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + \omega_{i+\frac{3}{2},j-\frac{1}{2},k+\frac{1}{2}} + \omega_{i+\frac{3}{2},j+\frac{1}{2},k-\frac{1}{2}} \\ & \qquad + \omega_{i-\frac{1}{2},j+\frac{3}{2},k+\frac{1}{2}} + \omega_{i+\frac{1}{2},j+\frac{3}{2},k-\frac{1}{2}} + \omega_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{3}{2}} + \omega_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{3}{2}}\Big) \\ & + \frac{1}{64}\Big(\omega_{i+\frac{3}{2},j+\frac{3}{2},k+\frac{3}{2}} + \omega_{i+\frac{3}{2},j+\frac{3}{2},k-\frac{1}{2}} + \omega_{i+\frac{3}{2},j-\frac{1}{2},k+\frac{3}{2}} + \omega_{i+\frac{3}{2},j-\frac{1}{2},k-\frac{1}{2}} \\ & \qquad + \omega_{i-\frac{1}{2},j+\frac{3}{2},k+\frac{3}{2}} + \omega_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{3}{2}} + \omega_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}\Big)\,. \end{aligned} \tag{34}$$

Boundary nodes have to be distributed separately with a similar, although lower dimensional, elliptic PDE. In many flow situations, discontinuities may start at the boundaries (e.g, vertical boundaries in the sloshing wave example or horizontal in the lock exchange) or move towards them at a later time (e.g., vertical boundaries in the lock-exchange example), thus, it is important to have a boundary distribution strategy in place to improve solution quality in the boundary regions. A reasonable strategy is to move the boundary nodes with the same speed as the tangential component of the speed for the internal points adjacent to those boundary nodes. In our case, because the boundaries are planar, we update nodes of the western boundary with

$$(\xi_1,\xi_2,\xi_3)^{[\nu+1]}_{0,j,k} = (\xi_1,\xi_2,\xi_3)^{[\nu]}_{0,j,k} + (0,\xi_2,\xi_3)^{[\nu+1]}_{1,j,k} - (0,\xi_2,\xi_3)^{[\nu]}_{1,j,k}\,, \tag{35}$$

where the superscript $[\nu+1]$ on the right-hand side refers to the new position of internal nodes updated at each iteration of the discrete mesh Equation (32). The redistribution of nodes at the other five boundaries can be carried out in a similar way.

5.3. Hybrid moving mesh method

Given deficiencies with the pure Lagrangian and variational approaches when moving the computational mesh, we combine both methods with the following procedure. We use the first method of evolving the computational mesh in a Lagrangian way as the default evolution approach. At regular intervals (either fixed or based on some other metric), we apply the moving mesh PDE method to 'regularize' the grid. This hybrid approach yields near-Lagrangian evolution of the grid in terms of minimizing the advection terms. It also circumvents the main disadvantage of a purely Lagrangian method which leads to eventual deterioration of the computational mesh.

5.4. Performance considerations

As will be demonstrated in Section 6, the moving grid method exhibits superior properties in terms of numerical diffusion minimization as well as error reduction. However, as expected, these advantages are achieved at a higher computational cost. Fortunately, there are parameters that can greatly accelerate the method. For once, it is the convergence of the Gauss–Seidel iteration loop for Equation (28) that can be controlled by the maximum iteration count or error tolerances. Both are adjustable parameters due to the nature of diminishing returns at higher iteration counts. Another parameter is α in the monitor function (29), which translates into the intensity of 'pull' towards the areas of interest. Making this parameter too high may over concentrate the nodes at the interface that tends to deteriorate grid quality. Another way of tuning the moving grid method is by spatially

DOI: 10.1002/fld

smoothing the monitor function. This raises the lower bound on interface curvature frequencies resolvable by the computational grid. This sometimes is critical as sharp interface curvature attracts more computational nodes than may be desired (see Figure 14). Finally, one does not have to adjust the grid at every time step. Because of spatial smoothing, introduced by Equation (34), there is a 'halo' region of fine grid resolution around the interface. Therefore, when the interface moves only a few grid cells per time step (reasonable requirement for numerical stability) and the low pass filter (34) is repeatedly applied to the monitor function, running the grid adjustment only every few time steps produces similar results. We dynamically decide on the necessity of grid adjustment by running one iteration of the Gauss–Seidel solver of discrete mesh Equation (32) at every time step and assessing how much the nodes move in that iteration. If the motion is minimal (i.e., $||\vec{\xi}^{[\nu+1]} - \vec{\xi}^{[\nu]}||_2 \leq \epsilon$), we proceed with the stationary part of the code, saving us the cost of solving the elliptic PDE (28) and the discrete GCL (23) to update Jacobians for the stationary curvilinear solver. Optimal choices of these parameters, to minimize invocation of the grid evolution component of the algorithm and yet stay within the desired limits of the numerical diffusion growth, can reduce the overhead for some simulations to a negligible fraction of the total simulation time, with the pressure solver constituting the performance bottleneck. On average, the moving grid simulations we ran were roughly 20% slower as compared with the their fixed-grid counterparts with the same number of computational cells.

6. SIMULATION RESULTS

6.1. Example1: interfacial sloshing wave

We demonstrate the benefits of the ALE adaptive mesh strategy with the standing interfacial gravity wave. We run this test case first with the static grid to acquire a baseline and then run it with a mesh that moves with the vertical fluid velocity (i.e., $u_{g,3} = u_3$). All simulations were run for two wave periods to provide enough time history for comparison and yet prevent the computational grid in the moving mesh case to deteriorate significantly. In this example, we do not use the PDE (28) because we want to eliminate the need to adjust α and other parameters associated with the variational approach to mesh adaptation.

The domain is initialized with a finite-amplitude deep-water standing wave in an inviscid fluid with the initial shape of the interface ζ approximated to second-order in steepness ka as (see [29])

$$k\zeta(x) = ka\left[\left(1 - \frac{(ka)^2}{64}\right)\cos kx - \frac{(ka)^2}{8}\cos 3kx\right], \tag{36}$$

where $k = \frac{\pi}{L}$ is the wavenumber, L is the domain length, and a is the amplitude. The initial density distribution is given by

$$\rho(x,z) = -\frac{\Delta\rho}{2}\tanh\left[\frac{2\tanh^{-1}\alpha}{k\delta}(kz - k\zeta + kd/2)\right], \tag{37}$$

where the density difference between two layers is $\Delta\rho/\rho_0 = 0.03$, the interface thickness is $k\delta = 0.05\pi$, the initial interface steepness is $ka = 0.1$, $\alpha = 0.99$, and d is the depth.

The evolution of the sloshing wave is computed in a $1.0 \times 1.0\ m$ tank on a 64×64 grid with the initial grid node distribution based on the variational approach (described in Section 5.2) applied to the analytical representation of the initial density distribution (37), as shown in Figure 3(b). Boundary conditions are no-slip at the lower boundary and free-slip at all other boundaries. The total simulation time is $2T$ and the time step is $\Delta t = 0.003T$, yielding a maximum CFL number of roughly 0.1. The governing equations for momentum, scalar transport, and the GCL are advanced in time with the second-order Adams–Bashforth method. The normalized residual of the multigrid pressure Poisson equation solver is 10^{-8}. As shown in Figure 4, the computational grid adapts to provide increased resolution at the fluid interface during the simulation. To demonstrate the superiority of the moving grid method, we compare the background potential energy growth

DOI: 10.1002/fld

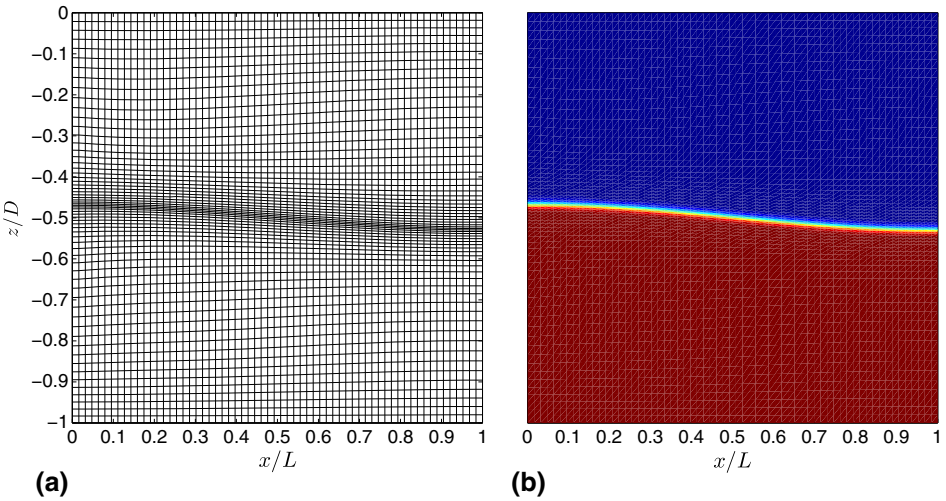

Figure 3. Interfacial sloshing wave test case: (a) typical grid produced by the variational approach ($\alpha = .001$, three outer iterations of Equation (32) were used) based on the density field from (b). (b) Initial density field.

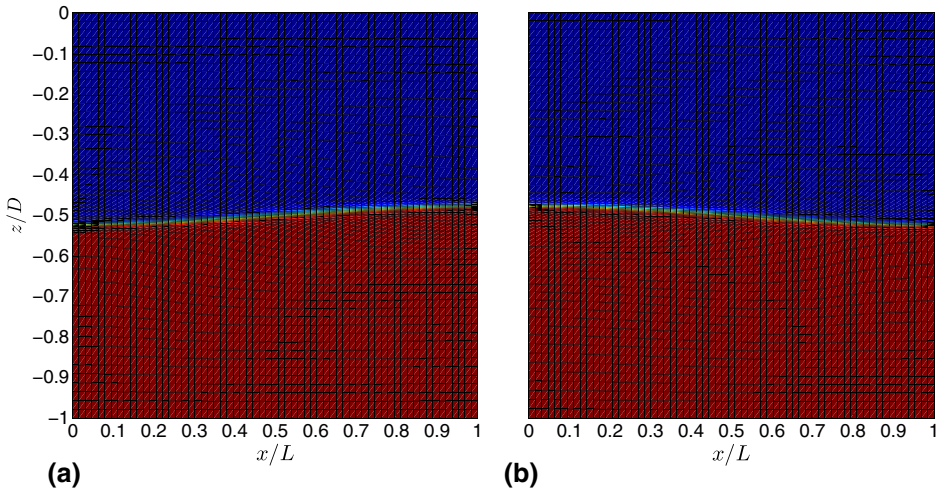

Figure 4. Interfacial sloshing wave test case at: (a) $t = T/2$; (b) $t = T$.

rates with and without the moving grid. We then compare error magnitudes and rates of spatial convergence for the two methods.

According to [22], the best overall performance (out of standard advection schemes) in terms of minimizing numerical diffusion on a fixed grid was demonstrated by the SHARP scheme [18]. The most diffusive was first-order upwinding. We run four grid scenarios for both upwind and SHARP advection schemes and use the same number of grid cells for the moving and static grid cases. The two moving grid scenarios are the variational approach and grid evolution based on the vertical fluid velocity. The static grid scenarios include a uniform Cartesian grid and a grid that was adapted to resolve the initial location of the fluid interface, as depicted in Figure 3(a). As Figures 5 and 6 demonstrate, adapting the grid allows to decrease the background potential energy growth rate by several orders of magnitude. The second static grid scenario was included to provide an equivalent starting point for the moving and static grid methods. In that scenario, once the fluid starts moving, this grid remains stationary. However, because of the nature of the example after an oscillation period, the fluid interface returns to its original position, at which point in time numerical diffusion is smaller. Using the initial node distribution shown in Figure 3(a) reduces the rate of growth of background potential energy at the beginning of simulation. However, once the density interface

DOI: 10.1002/fld

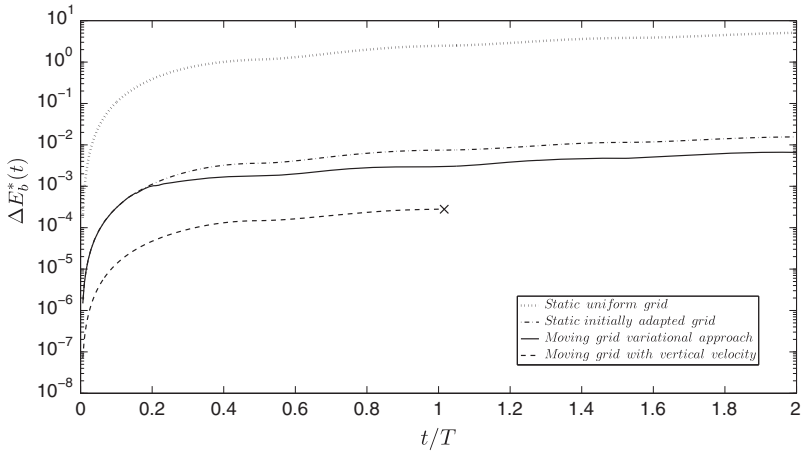

Figure 5. Evolution of the background potential energy for the interfacial sloshing wave test case on a static and moving grid using first-order upwind scheme for scalar advection. The '×' for the bottom graph represents early termination of simulation due to the grid deterioration as described in Section 5.1. The log scale for the ordinate was used due to several orders of magnitude difference between results for uniform and adaptive grids.

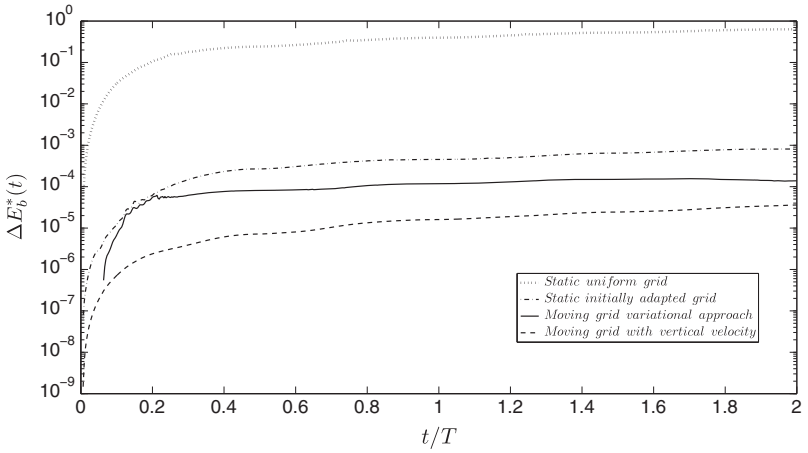

Figure 6. Evolution of the background potential energy for the interfacial sloshing wave test case on a static and moving grid using the SHARP scheme for scalar advection. The log scale for the ordinate was used due to several orders of magnitude difference between results for uniform and adaptive grids.

leaves the initially refined area of the mesh, the rate of growth increases, as demonstrated in Figures 5 and 6, where the curves for the initially adapted static and moving grid cases diverge roughly after $t = 0.2T$. It is possible to refine the larger area of the mesh that would fully cover the oscillating interface at all times and obtain similar results for both the moving and fixed-grid approaches. Seemingly, an attractive alternative to moving the grid, this modified fixed-grid strategy is not applicable in every situation as will be demonstrated by the test case in Section 6.2, where no initial grid node distribution can benefit the lock-exchange evolution at every time step of simulation and the only feasible moving grid strategy is based on the variational approach. As demonstrated in Figures 5 and 6, moving the grid based on the vertical velocity preserves the background potential energy at a higher rate as compared with the variational grid approach. This happens due to the targeted annihilation of advection terms in equations (5) and (6) with the ALE grid and only approximate reduction of numerical diffusion when grid nodes are drawn closer to the interface in the variational framework. However, moving the grid based on the fluid velocity is problematic in the long run due to grid tangling as a result of the mean flow net effect, as shown in Figure 2. This inadvertent grid deterioration may lead to a premature termination of a simulation that otherwise

DOI: 10.1002/fld

accurately preserves the background potential energy. Our moving grid simulation based on the vertical velocity terminated shortly after $t = T$ using the first-order upwind advection scheme, as demonstrated in Figure 5 and around $t = 2.25T$ using the SHARP advection scheme (beyond the axes limits in Figure 5). To avoid this problem, we introduce periodic grid regularization based on the variational moving mesh approach as discussed in Section 5.3. The regularization procedure can be initiated at regular intervals (roughly every $T/2$ in our example with $\alpha = .001$ and 10 smoothing iterations with Equation (34)) or can be based on some grid quality measure (e.g, ratio of smallest and largest cell volumes in the grid). Grid regularization leads to a small discontinuity in the background potential energy as shown in Figure 7. The solid line in the figure refers to the hybrid moving grid approach with the variational regularization procedure applied twice and using first-order upwind.

By examining Figure 7, we can see that the hybrid moving grid method with first-order upwind advection preserves the background potential energy more efficiently than the SHARP advection scheme with both the initially adjusted and the uniform static grid approaches. As expected, the static grid strategy based on the adjusted initial profile preserves the background potential energy better than the uniform initial grid approach. Finally, when the SHARP advection scheme is combined with the variational grid adaptation, it preserves the background potential energy more efficiently than the first-order upwind hybrid moving mesh approach. The hybrid approach is not used for the SHARP scheme because the grid deterioration is slower for it as compared with the first-order upwind case, and for comparison purposes, all schemes were simulated for two wave periods only. Another reason for not applying the variational regularization procedure to the SHARP scheme is due to its minimal effect on the rate of change of the background potential energy. Thus, the real reason to regularize the mesh is to prolong the simulation by balancing the effect of the time-averaged mean flow. As in the upwind case, moving the grid based on the vertical velocity for the SHARP advection scheme conserves the background potential energy more efficiently compared with the variational approach. The periodic variation, especially noticeable in the uniform grid cases, occurs due to lower numerical diffusion, when the interface comes to rest at every half cycle.

To study the spatial convergence, a series of simulations was performed (i.e., $x - y$ grid dimensions: 32×32, 48×48, 64×64, 80×80) both for moving with vertical fluid velocity and static grids with the same time step of $\Delta t = 0.003T$ for a total time of $T/4$. The error was calculated with

$$e(\Delta x) = \frac{\sqrt{\sum_{i,j,k}\left(\rho_{\Delta x}(i,j,k) - \tilde{\rho}_{\frac{\Delta x}{2}}(i,j,k)\right)^2 J(i,j,k)}}{\sum_{i,j,k} J_{i,j,k}}, \tag{38}$$

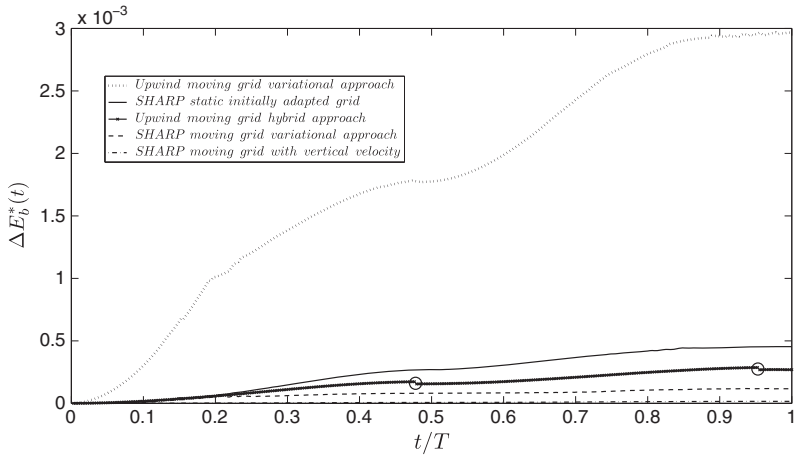

Figure 7. Effect of the moving grid scheme on evolution of the background potential energy for different advection and grid adaptation schemes. Circles on the bottom graph indicate application of the regularization procedure as described in Section 5.3.

DOI: 10.1002/fld

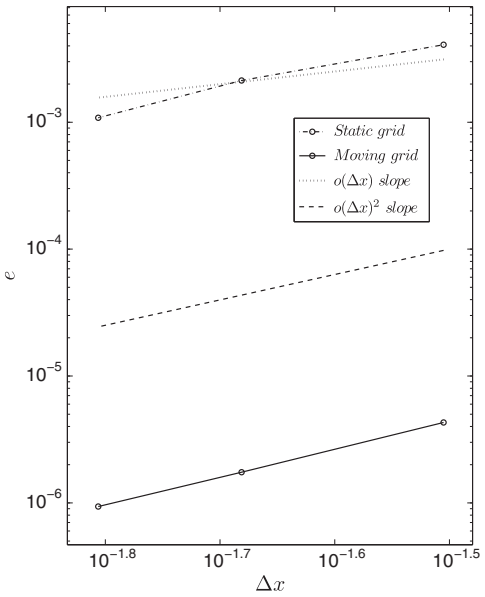

Figure 8. Spatial convergence of the moving and static grid approaches for the interfacial sloshing wave test case at $t = T/4$.

where the summation is across all cells in the Δx-sized grid with Jacobians, $J(i, j, k)$, representing their corresponding volumes. Here, $\tilde{\rho}_{\frac{\Delta x}{2}}$ is the density interpolated onto the coarse grid solution. The convergence results in Figure 8 demonstrate that both methods converge to second-order in grid size with the error being smaller for the moving grid method. It is worth noting that spatial convergence slightly deteriorates with time, thus the errors were calculated relatively close to the start of simulation, at $t = T/4$.

6.2. *Example 2: nonhydrostatic lock exchange*

Because of simplicity of interface motion, simulations of the internal gravity wave can be performed with the Lagrangian approach to mesh evolution. Though, undoubtedly, regular applications of the variational approach to mesh adaptation allowed to prevent mesh deterioration with the mean flow in the long run. In order to fully appreciate the benefits of the hybrid approach, we need to apply it to a fluid interface that moves more irregularly. The nonhydrostatic lock-exchange test case, following [15], presents an opportunity to test our moving mesh methodology on a more complex problem.

The lock-exchange flow is generated in a plane channel filled with two fluids of different density, initially separated by a vertical gate, as depicted in Figure 9(a). The density difference between separated fluids is $\Delta\rho/\rho_0 = 0.001$. When the gate is withdrawn, a mutual intrusion flow drives two fronts in opposite directions, generating Kelvin–Helmholtz billows at the fluid interface. The simulation corresponding to Figure 9 is performed in a $0.8 \times 0.1 \times 0.1$ m domain on a $128 \times 32 \times 8$ grid with a no-slip condition at the lower boundary and free-slip at all other boundaries. This configuration makes possible the investigation of fluid behavior with a no-slip condition at the bottom and a free-slip condition at the top in one simulation. The molecular viscosity is set to $\nu = 10^{-6}$ m^2 s^{-1}, and there is no physical scalar diffusivity. For this simulation, the total time is $15T$, capturing both gravity currents reaching their corresponding walls, and the time step is $\Delta t = 0.01T$, yielding a maximum CFL number of roughly 0.25. Here, $T = \sqrt{D/2g'}$, where $g' = g\Delta\rho/\rho_0 = 0.01$ m s^{-2} is the reduced gravity. Starting with a uniform grid at $t = 0$, the moving grid adaptation was applied at each iteration with $\alpha = 2.5 \times 10^{-6}$, $\epsilon = 3 \times 10^{-4}$ tolerance for the Gauss–Seidel solver, and $n_{smooth} = 4$ smoothing iterations for the monitor function. Qualitative differences between the static and moving grid methods are well-demonstrated by comparing Figures 10 and 11. In the moving grid case, with the same number of computational nodes, the interface is sharper and the

DOI: 10.1002/fld

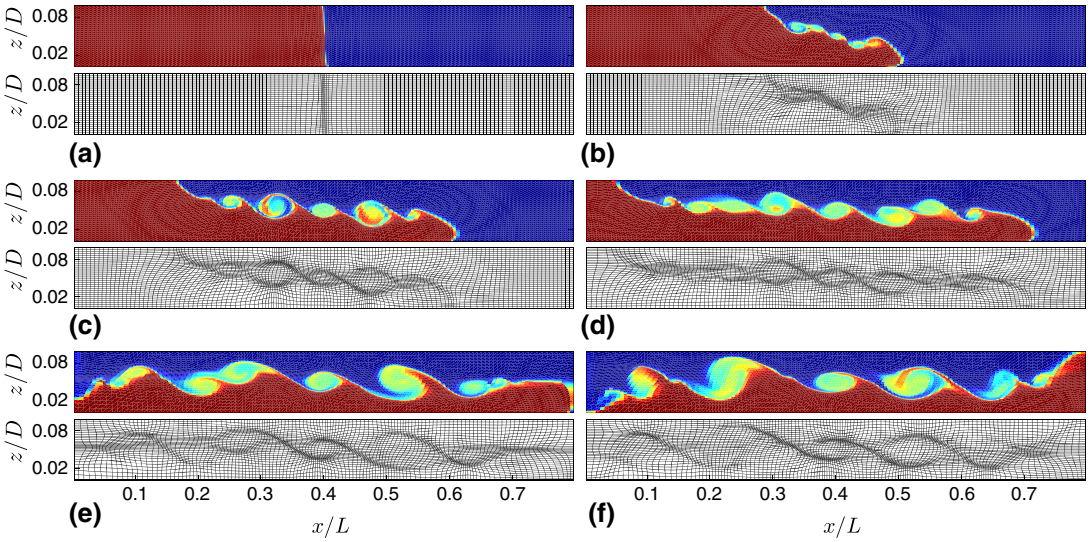

Figure 9. Lock-exchange test case and the moving grid at $t/T = 0.1$(a), 4(b), 7.5(c), 10(d), 12.5(e), and 15(f). This vertical slice is taken at $z/W = 0.5$.

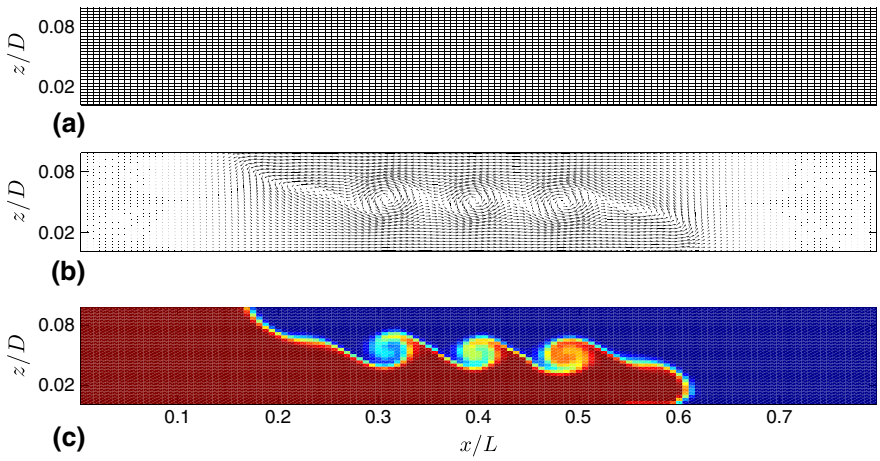

Figure 10. Computational grid, velocity vectors, and density field of the lock-exchange test case on a fixed grid at $t = 7T$. Vertical slice is taken at $z/W = 0.5$.

Kelvin–Helmholtz billows are larger and more resolved. Better interface resolution with the moving grid leads to the formation of five billows in Figure 11 as opposed to three in Figure 10. For a more quantitative comparison, we compare the methods based on three quantities: background potential energy evolution, spatial convergence, and propagation speed of gravity currents.

As in the internal wave test case, we calculate evolution of the background potential energy with Equation (18). Scalar advection is computed with the SHARP scheme for the static and moving grid cases. As shown in Figure 12, the moving grid preserves the background potential energy roughly by a factor of 2. Once the billows at the lock-exchange interface start forming, the fixed-grid method starts producing increasingly more numerical diffusion than the moving grid approach as demonstrated by the diverging background potential energy curves for both methods after $t = 5T$ in Figure 12.

Unlike in the standing wave problem in which the reduction in the growth of background potential energy was roughly one order of magnitude, the improvement in the lock-exchange case is much less significant. This difference can be explained by the fact that the movement of the interface in the sloshing wave example was essentially one dimensional. Yet, in the lock-exchange case,

DOI: 10.1002/fld

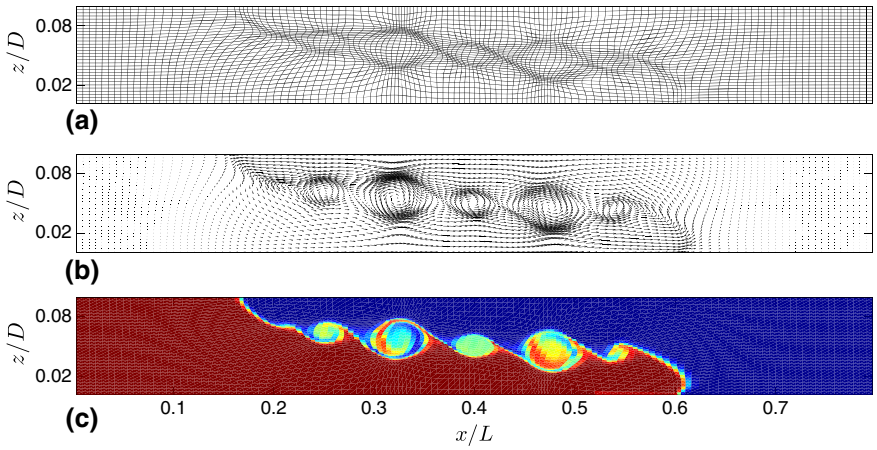

Figure 11. Computational grid, velocity vectors, and density field of the lock-exchange test case with the moving grid at $t = 7T$. Vertical slice is taken at $z/W = 0.5$.

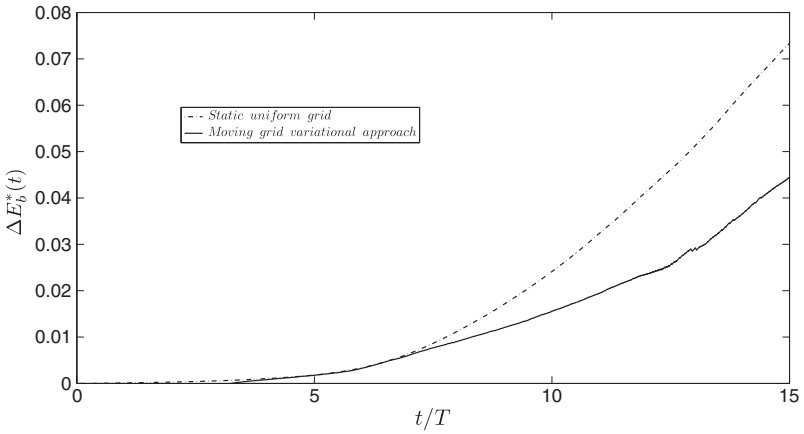

Figure 12. Evolution of the background potential energy for the lock-exchange test case on a static and moving grid using SHARP advection discretization scheme.

the interface evolution was much more complex and three dimensional. A close examination of Figure 12 reveals that the background potential energy curves for the uniform and moving grid methods diverge at the time when K–H billows form. Although the variational approach helps refine these instability regions, the grid does not move in a Lagrangian way that would most closely conserve the background potential energy. In the lock-exchange case, Lagrangian grid evolution would lead to tangling. However, in the sloshing way case, because of the regular behavior of the fluid interface, the variational mesh evolution closely mimics the Lagrangian moving mesh in terms of background potential energy conservation. There is also an issue of over resolving specific areas of interest at the interface that leads to poor grid conditioning. Thus, in the lock-exchange case, we must restrain the variational method from concentrating grid nodes in the areas of rapid density change. This was not as important a consideration in the sloshing wave case due to the smooth shape of the interface that was free of K–H billows. There is a balance to be achieved between infusing the interface with grid points as profusely as possible and maintaining convergence properties of the multigrid pressure solver affected within tolerable limits. Perhaps, in combination with more effective grid evolution strategies, the issue of slowing multigrid convergence could be addressed by enhancing the existing multigrid method (i.e., 'semicoarsening' strategies with plane relaxation or multiple semicoarse grid correction schemes [30]).

DOI: 10.1002/fld

To study the spatial convergence, a series of simulations was performed with 64×16, 128×32, 256×64, and 512×128 grid points both for static and moving grids with the same time step of $\Delta t = 0.01T$ for a total time of $2T$. The variational moving grid method was applied to the lowest resolution case with $\alpha = 5 \times 10^{-6}$, $\epsilon = 6 \times 10^{-4}$, and $n_{smooth} = 2$, but then α and ϵ were halved and n_{smooth} doubled with each consecutive grid refinement. The spatial convergence for this example was evaluated with Equation (38). Figure 13 demonstrates the second-order spatial convergence for both methods, with the moving grid approach being more accurate, roughly by a factor of 2. Table I compares Froude numbers calculated from our simulations with the direct numerical simulation (DNS) results in [15]. The Froude number is the ratio of the speed of the gravity current to the buoyancy velocity $u_b = \sqrt{g'D/2}$, and the Grashof number for these simulations is given by $Gr = (u_b D/2\nu)^2 = 1.25 \times 10^6$. In our lock-exchange simulations, the top front propagates to the left and is faster because of the free-slip boundary condition at the top of the domain, and the bottom front is slower because of the no-slip boundary condition at the bottom of the domain. The front speeds were computed by first finding the minimum and maximum locations of the density discontinuity in each simulation frame and then approximating the front velocities with the first-order time difference scheme. At the beginning of simulation, both fronts start moving

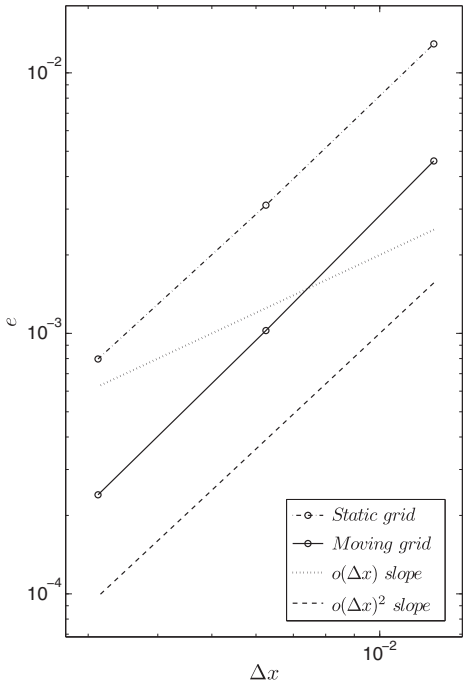

Figure 13. Spatial convergence of the moving and static grid approaches for the lock-exchange test case at $t = 2T$.

Table I. Comparison of the Froude number computations for the lock-exchange test case at different grid resolutions for the static and moving grid methods with those of Hartel *et al.* [15]. The no-slip boundary condition row represents the speed of the rightward-propagating front at the bottom, and the free-slip boundary condition row represents the speed of the leftward-propagating front at the top. For each type of boundary condition, the discrepancy with the result of Hartel [15] is also presented.

	128×32		256×64		512×128		
Boundary condition	static	moving	static	moving	static	moving	Hartel [15]
No-slip	0.5332	0.5499	0.5547	0.5642	0.5650	0.5700	0.5740
Hartel [15] - No-slip	0.0408	0.0241	0.0193	0.0098	0.0088	0.0040	0
Free-slip	0.6495	0.6547	0.6571	0.6583	0.6598	0.6602	0.6750
Hartel [15] - Free-slip	0.0255	0.0203	0.0179	0.0167	0.0152	0.0148	0

DOI: 10.1002/fld

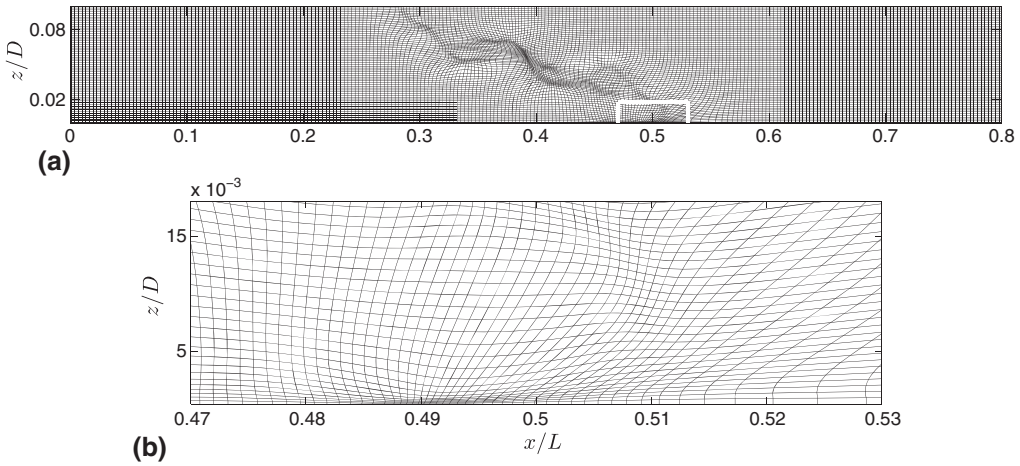

Figure 14. Over-concentration of computational nodes under the right-propagating gravity head of a lock-exchange simulation at $t = 4T$ due to the formation of the statically unstable head. Vertical 512×128 grid point slice is taken at $z/H = 0.5$. Figure (b) is a zoomed view of the white rectangular box in Figure (a) that displays only odd grid lines for clarity.

in opposite directions and after some time reach a steady velocity, the value of which we present in Table I. As the grid resolution is refined, both static and moving grid methods converge to the values obtained in [15] with the static grid results slightly lagging behind. In light of numerical diffusion, which slows down an interface, this result is not surprising and seems to correspond well with Figure 12 that shows numerical diffusion evolution for both methods. Ultimately, this means that the moving grid method does a better job of approximating the correct front speed, both for the no-slip and free-slip boundary condition cases, whereas the static grid method underpredicts both speeds for simulations with an equivalent number of grid points.

It is noteworthy that the effect of the moving grid method is more pronounced for the no-slip front than it is for the free-slip front. This discrepancy is due to shape dissimilarities of the fronts. In the no-slip case, the front is convexing forward trapping some lighter fluid underneath, as shown in Figure 14, thus attracting more grid points to resolve this complexity and consequently reducing numerical diffusion at a higher rate. This discrepancy in refinement between two fronts could be better resolved by adapting the grid based on vorticity instead of density in Equation (29) or alternatively using anisotropic monitor function M, that is,

$$M = \begin{pmatrix} \sqrt{1+\alpha\left(\frac{\partial\rho}{\partial x}\right)^2} & 0 & 0 \\ 0 & \sqrt{1+\alpha\left(\frac{\partial\rho}{\partial y}\right)^2} & 0 \\ 0 & 0 & \sqrt{1+\alpha\left(\frac{\partial\rho}{\partial z}\right)^2} \end{pmatrix}. \tag{39}$$

This is the subject of the ongoing work and beyond the scope of this manuscript.

7. CONCLUSIONS

In this paper, we showed the advantages of r-adaptivity for minimizing numerical diffusion of advection schemes used for discretization of Navier–Stokes equations with the Boussinesq approximation. Using the same number of grid points with the hybrid moving approach, it was possible to achieve lower numerical diffusion with the first-order upwind advection discretization scheme than with the more accurate but expensive SHARP scheme on a static grid. In the interfacial gravity wave example, moving the mesh with the vertical component of fluid velocity yields a reduction by several orders of magnitude in the numerical diffusion compared with the static grid simulation on

DOI: 10.1002/fld

a uniform grid. For an objective comparison, an additional static grid scenario of initially adapting the grid with the variational approach and then keeping it unchanged during the simulation was also considered. Initial grid adaptation leads to a significant reduction of the background potential energy due to the oscillating nature of the interface in the sloshing wave test case. Moving the grid for the initially adapted interfacial sloshing wave example caused additional reduction of the numerical diffusion. The strategy of moving the grid based on the vertical fluid velocity yields lower growth rate of the background potential energy compared with the variational grid adaptation approach. However, in terms of robustness, the variational approach prevails at the cost of slightly higher rate of background potential energy growth. This suggested an alternative hybrid grid moving strategy of using the variational approach as a regular grid regularization scheme that was alleviating the negative long-term effect of the mean flow for the grid adaptation strategy based on the vertical fluid velocity. In the lock-exchange test case, the only feasible approach to grid motion due to its complexity is based on the variational approach.

Although qualitative advantages of the moving grid method over its static counterpart can be assessed from results of simulations with the equivalent number of cells, the static and moving grid methods are quantitatively compared, for both test cases in the paper, based on the background potential energy evolution and spatial convergence. The benefit of reduced numerical diffusion as a consequence of moving the grid in the lock-exchange test case results in more accurate capture of the gravity currents with both no-slip and free-slip boundary conditions. This is the third metric of comparison used in the lock-exchange test case. On the basis of this metric, the moving grid front speeds more closely match those of [15]. The computational overhead of the moving grid method is adjustable based on the desired quality of interface tracking and in our experience does not exceed 20% of the overall computation time.

Although this paper demonstrates the advantages of using the moving grid approach for simulations involving stratified flows, it is important to note that deciding on a grid node moving strategy can be quite challenging. Simple evolution of the grid based on the density field gradient is effective as a lower-order approximation of the fluid interface. The higher-order effects are more illusive and require a special treatment to move grid nodes more precisely to follow complex interfaces. This limitation of the simple interface tracking strategy is manifested by the less significant reduction of the background potential energy for the lock-exchange case compared with the sloshing wave example. Therefore, in the future, we plan to experiment with different monitor functions that control mesh evolution in the variational framework. One such approach is keeping track of the fluid interface with the level set function [31] and subsequently using it to construct a more accurate monitor function (29). This could be attractive in the presence of complex interfaces that level set methods resolve very well. There, of course, is the added burden of properly solving the level set PDE and more importantly, evaluating the benefits of more accurate interface representation relative to added computational cost. These are among topics of future research in dynamic meshes.

ACKNOWLEDGEMENTS

This research was supported by the Publishing Arts Research Council under grant no.98-1846389. We gratefully acknowledge the support of the ONR Physical Oceanography Program under Grant N00014-10-1-0521 (scientific officers Dr. C. Linwood Vincent, Dr. Terri Paluszkiewicz, and Dr. Scott Harper). Simulations were carried out on the Harold machine at the US Army Research Lab DoD Supercomputing Resource Center. We would also like to thank Mr. Yi-Ju Chou for many helpful discussions around implications of CWC with moving grids and LES solvers.

REFERENCES

1. Shyy W, Udaykumar HS, Rao MM, Smith RW. *Computational Fluid Dynamics with Moving Boundaries*. Taylor & Francis: London, 1996.
2. Jin C, Xu K. A unified moving grid gas-kinetic method in Eulerian space for viscous flow comnputation. *Journal of Computational Physics* 2007; **222**:155–175.
3. Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *Journal of Computational Physics* 1997; **135**:203–216.

DOI: 10.1002/fld

4. Piggott M, Pain C, Gorman G, Power P, Goddard A. h, r, and hr adaptivity with applications in numerical ocean modeling. *Ocean Modelling* 2005; **10**:95–113.
5. Huang WZ, Ren Y, Russell RD. Moving mesh methods based on moving mesh partial differential equations. *Journal of Computational Physics* 1994; **113**:279–290.
6. Miller K, Miller RN. Moving finite element. *SIAM Journal of Numerical Analysis* 1981; **18**:1019–1032.
7. Dorfi EA, Drury LO. Simple adaptive grids for 1-D initial value problems. *Journal of Computational Physics* 1987; **69**:175–195.
8. Tang T. Moving mesh methods for computational fluid dynamics. *Contemporary Mathematics* 2005; **383**:141–174.
9. Tang H, Tang T. Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws. *SIAM Journal of Numerical Analysis* 2003; **41**(2):487–515.
10. Tang H, Tang T. Multi-dimensional moving mesh method for shock computations. *Contemporary Mathematics* 2003; **330**:169–183.
11. Chou YJ, Fringer OB. Consistent discretization for simulations of flows with moving generalized curvilinear coordinates. *International Journal for Numerical Methods in Fluids* 2010; **62**:802–826.
12. Winters KB, Lombard PN, Riley JJ, D'Asaro EA. Available potential energy and mixing in density-stratified fluids. *Journal of Fluid Mechanics* 1995; **289**:115–128.
13. Hiester HR, Piggott MD, Allison PA. The impact of mesh adaptivity on the gravity current front speed in a two-dimensional lock-exchange. *Ocean Modelling* 2011; **38**:1–21.
14. Fringer OB, Gerritsen M, Street RL. An unstructured grid, finite-volume, nonhydrostatic, parallel coastal ocean simulator. *Ocean Modelling* 2006; **14**(3-4):139–278.
15. Hartel C, Meiburg E, Necker F. Analysis and direct numerical simulation of the flow at a gravity-current head. Part 1. Flow topology and front speed for slip and no-slip boundaries. *Journal of Fluid Mechanics* 2000; **418**:189–212.
16. Zang Y, Street RL, Koseff JR. A non-staggered grid, fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear coordinates. *Journal of Computational Physics* 1994; **114**:18–33.
17. Leonard BP. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer Methods in Applied Mechanics* 1979; **19**:59–98.
18. Leonard BP. Simple high-accuracy resolution program for convective modelling of discontinuities. *International Journal for Numerical Methods in Fluids* 1988; **8**:1291–1318.
19. Cui AQ, Street RL. Large-eddy simulation of coastal upwelling flow. *Environmental Fluid Mechanics* 2004; **4**:197–223.
20. Leonard BP, Lock AP, MacVean MK. Conservative explicit unrestricted-time-step multidimensional constancy-preserving advection schemes. *Monthly Weather Review* 1996; **124**:2588–2606.
21. Gross ES, Koseff JR, Monismith SG. Evaluation of advective schemes for estuarine salinity simulations. *ASCE Journal of Hydraulic Engineering* 1999; **45**:15–21.
22. Fringer OB, Armfield SW, Street RL. Reducing numerical diffusion in interfacial gravity wave simulations. *International Journal for Numerical Methods in Fluids* 2005; **49**(3):301–329.
23. Li X, Lu P, Schaeffer J, Shillington J, Wong PS, Shi H. On the versatility of parallel sorting by regular sampling. *Parallel Computing* 1993; **19**:1079–1193.
24. LeVeque R. High resolution conservative algorithms for advection in incompressible flow. *SIAM Journal of Numerical Analysis* 1996; **33**:627–665.
25. Lin S, Rood RB. Multidimensional flux-form semi-Lagrangian transport schemes. *Monthly Weather Review* 1996; **124**:2588–2606.
26. Thomas PD, Lombard CK. Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal* 1979; **17**(10):1030–1037.
27. Trulio JG, Trigger KR. Numerical solution of the one-dimensional hydrodynamics equations in an arbitrary time-dependent coordinate system. *University of California Lawrence Radiation Laboratory Report UCLR-6522*, 1961.
28. Demirdžić I, Peric M. Space conservation law in finite volume calculation of fluid flow. *International Journal for Numerical Methods in Fluids* 1988; **8**:1037–1050.
29. Thorpe SA. On standing internal gravity waves of finite amplitude. *Journal of Fluid Mechanics* 1968; **32**:489–528.
30. Naik NH, Rosendal JV. The improved robustness of multigrid elliptic solvers based on multiple semicoarsened grids. *SIAM Journal of Numerical Analysis* 1993; **30**:215–229.
31. Osher S, Sethian JA. Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics* 1988; **79**:12–49.

DOI: 10.1002/fld