Detailed simulation of morphodynamics: 1. Hydrodynamic model

M. Nabi,[1] H. J. de Vriend,[2,3] E. Mosselman,[2,3] C. J. Sloff,[2,3] and Y. Shimizu[1]

Received 26 January 2012; revised 31 October 2012; accepted 2 November 2012; published 20 December 2012.

[1] We present a three-dimensional high-resolution hydrodynamic model for unsteady incompressible flow over an evolving bed topography. This is achieved by using a multilevel Cartesian grid technique that allows the grid to be refined in high-gradient regions and in the vicinity of the river bed. The grid can be locally refined and adapted to the bed geometry, managing the Cartesian grid cells and faces using a hierarchical tree data approach. A ghost-cell immersed-boundary technique is applied to cells intersecting the bed topography. The governing equations have been discretized using a finite-volume method on a staggered grid, conserving second-order accuracy in time and space. The solution advances in time using the fractional step approach. Large-eddy simulation is used as turbulence closure. We validate the model against several experiments and other results from literature. Model results for Stokes flow around a cylinder in the vicinity of a moving wall agree well with Wannier's analytical solution. At higher Reynolds numbers, computed trailing bubble length, separation angle, and drag coefficient compare favorably with experimental and previous computational results. Results for the flow over two- and three-dimensional dunes agree well with published data, including a fair reproduction of recirculation zones, horse-shoe structures, and boiling effects. This shows that the model is suitable for being used as a hydrodynamic submodel in the high-resolution modeling of sediment transport and formation and evolution of subaqueous ripples and dunes.

Citation: Nabi, M., H. J. de Vriend, E. Mosselman, C. J. Sloff, and Y. Shimizu (2012), Detailed simulation of morphodynamics: 1. Hydrodynamic model, *Water Resour. Res.*, *48*, W12523, doi:doi:10.1029/2012WR011911.

1. Introduction

[2] The deformation of alluvial beds under the action of flowing water produces complex geometries that change in time. Solving such problems is one of the main challenges in computational morphodynamics. Methods based on boundary-fitted grids offer an attractive way to deal with complex geometries, because they greatly simplify the imposition of boundary conditions. However, the possibilities to fit the boundaries are limited for structured grids, whilst unstructured grids are computationally more expensive. Moreover, boundary-fitted grids must be regenerated after each deformation of the boundaries, which requires much computational effort. Generation of a grid suitable for complex geometries can also be troublesome and sometimes cannot be applied without multiblock techniques. These issues lead to complexity and possibly instability in the computational process. To avoid these problems, we use a fixed Cartesian grid with special treatment of the boundary zones.

[3] Methods based on Cartesian grids have attracted special attention, now that techniques to handle complex geometries have become available. The Cartesian grids are fixed and intersect the boundaries, which requires special treatment of these boundaries, such as cut-cell techniques, or ghost-cell immersed-boundary methods. In the former approach, the cells that intersect the boundaries are cut. This approach has been used for inviscid [*Bayyuk*, 1996; *Quirk*, 1994] and viscous flow computations, including fixed and moving boundaries [*Mittal et al.*, 2008; *Udaykumar et al.*, 2001; *Ye et al.*, 1999; *Kirkpatrick et al.*, 2003]. It represents the boundary location accurately, but a boundary can intersect a cell in many ways, which causes complexity in programming and reduces computational efficiency. Ghost-cell immersed-boundary techniques force the flow at the boundary using simple interpolation, which does not increase computation time significantly. This method has been applied to viscous flows with fixed [*Gilmanov et al.*, 2003; *Tseng and Ferziger*, 2003; *Balaras*, 2004] and moving boundaries [*Fadlun et al.*, 2000; *Yang and Balaras*, 2006]. The cases they describe show that the ghost-cell immersed-boundary method is suitable for problems with very complex geometries.

[4] Uniform Cartesian grids are still expensive for simulation of alluvial processes with large spatial and temporal scales, because the entire domain has to be covered by cells of the same size. The regions with highest gradients determine the cell size, whereas they usually cover only a small

[1]Laboratory of Hydraulic Research, Hokkaido University, Sapporo, Japan.
[2]Section of Hydraulic Engineering, Delft University of Technology, Delft, Netherlands.
[3]Deltares, Delft, Netherlands.

Corresponding author: M. Nabi, Laboratory of Hydraulic Research, Graduate School of Engineering, Hokkaido University, Sapporo, Japan. (m.nabi@eng.hokudai.ac.jp)

0043-1397/12/2012WR011911

fraction of the domain. A locally refined Cartesian grid is therefore preferable. This method has been applied successfully to inviscid flows [*Aftosmis et al.*, 1998; *Clarke et al.*, 1986; *Zeeuw and Powell*, 1993; *Waymel et al.*, 2006; *Ham et al.*, 2002] and viscous flows [*Martin et al.*, 2008; *Iaccarino and Verzicco*, 2003]. *Aftosmis et al.* [1998] consider the grid as fully unstructured and discretize the equations on a single grid, whereas *Ham et al.* [2002] treat the grid as multilevel and use a hierarchical tree data structure. In a multilevel grid, each grid covers a subdomain of a coarser grid. Some authors have suggested [*Coirier and Powell*, 1996] or demonstrated [*Berger et al.*, 2000; *Ham et al.*, 2002] that the hierarchical tree structure for the grid is amenable to multigrid methods, which are more efficient than the Krylov space methods for large matrices [*Saad*, 2003].

[5] To avoid complexity and the need of regridding after each time step, we solve the flow on a Cartesian grid that is locally refined in high-gradient regions such as the boundary layer near the river bed. A ghost-cell immersed-boundary method is applied, enforcing the no-slip condition on the bed surface via momentum forcing at the bed. The current method interpolates the flow for the ghost cells in a way similar to *Balaras* [2004]. The current paper focuses on describing the numerical methods for the hydrodynamic part of the solver, validating the accuracy of the applied approach and demonstrating the capabilities of the solver in complex dynamic geometries. The cases used for this hydrodynamic validation are Stokes flow around a cylinder in the vicinity of a moving wall, flow around a cylinder at higher Reynolds numbers, and flow over two-dimensional (2-D) and three-dimensional (3-D) dunes. Two companion papers (M. Nabi et al., Detailed simulation of morphodynamics: 2. Sediment pick-up, transport and deposition; 3. Ripples and dunes, submitted to *Water Resour. Res.*, 2013a, 2013b) show that the model performs well as a hydrodynamic submodel in the high-resolution modeling of sediment transport and formation and evolution of subaqueous ripples and dunes.

2. Governing Equations

[6] The governing equations for the fluid are the full 3-D, unsteady, incompressible Navier-Stokes equations written in terms of primitive variables. These equations are given below in terms of volume-filtered variables.

$$\frac{\partial \overline{u}_j}{\partial x_j} = 0, \tag{1}$$

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial \overline{u}_i \overline{u}_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{P}}{\partial x_i} + \frac{\partial}{\partial x_j}[2(\nu + \nu_t)\overline{S}_{ij}], \tag{2}$$

where x_i's are the coordinates, t is the time, $\overline{P}$ is the modified pressure, ρ_0 is the mass density, $\overline{u}_i$ is the filtered velocity component in x_i direction, ν and ν_t are the molecular and turbulent viscosities, respectively, and S_{ij} is the resolved strain rate tensor:

$$\overline{S}_{ij} = \frac{1}{2}\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right). \tag{3}$$

In large eddy simulation (LES), the large eddies are solved directly, whereas the smaller ones are captured by a turbulence closure. LES uses volume filtering, eliminating eddies smaller than the grid cell volume. The effect of the small scales on the resolved part of turbulence appears in the subgrid-scale (SGS) stress term

$$\tau_{ij} = \overline{u_i u_j} - \overline{u}_i \overline{u}_j \tag{4}$$

which must be modeled. The SGS effect is modeled using the Smagorinsky model [*Smagorinsky*, 1963], in which the turbulence viscosity is defined as

$$\nu_t = (C_S \overline{\Delta})^2 |\overline{S}|, \tag{5}$$

where $C_S = 0.16$ is Smagorinsky constant, and

$$|\overline{S}| = \sqrt{2\overline{S}_{ij}\overline{S}_{ij}}. \tag{6}$$

$$\overline{\Delta} = (\Delta x.\Delta y.\Delta z)^{1/3}. \tag{7}$$

Fixed boundaries are treated with a wall-function technique rather than a dynamic model, as there is no viscous sublayer on rough boundaries such as a river bed. Dynamic SGS model is more accurate if the bed is smooth. As the bed is covered by sediment, the bed cannot be treated same as a smooth bed. Viscous sublayer can be destroyed on rough bed (rough because of sediment not because of ripples). The necessary condition for the dynamic model is imposing at least one cell in the viscous sublayer, because the dynamic model interpolates the boundary condition linearly [*Fadlun et al.*, 2000; *Balaras*, 2004; *Tseng and Ferziger*, 2003; *Yang and Balaras*, 2006]. The following expression for the near-bed region is adopted:

$$\frac{u_p}{u_*} = \frac{1}{\kappa}\ln\frac{y_p}{y_0}, \tag{8}$$

where y_p is the distance of a near-bed grid point to the bed, u_p is flow velocity at the near-bed grid point, u_* is local bed shear velocity, κ is the von Kármán constant, $y_0 = k_s/30$, and $k_s = 2.5d_{50}$, in which k_s is roughness height and d_{50} is bed median sediment diameter. As we deal with uniform sediment, d_{50} can be replaced by the sediment diameter d.

[7] The wall-function techniques yield satisfactory results over flat beds, but in regions with mildly complex geometries, and hence in the flow separation zones, the results may deviate from the accurate solution. The aim of the current model is to be applied for simulation of sediment transport and morphodynamical changes. As the flow in the separation zones is weak, it does not pick up much sediment, so that wall-function techniques are a suitable choice.

3. Numerical Methods

3.1. Discretization

[8] Equations (1) and (2) are solved with a second-order accurate, two-step fractional step method. The diffusion and advection terms in the momentum equations are discretized by a second-order finite-volume method. Pressure and

velocities are discretized on a staggered grid, in which the pressure is located in the center and the velocities are located on the faces of the cells. The solution advances in time using an Adams-Bashforth-Crank-Nicolson method, which conserves second-order accuracy in time. The diffusion terms are discretized using an implicit Crank-Nicolson scheme, the advection terms with a second-order Adams-Bashforth scheme. This eliminates the viscous stability constraint, which can be quite stringent in case of viscous flows. The two-step fractional step method estimates the velocity field by the predictor

$$\frac{1}{\Delta t}(\mathbf{u}^* - \overline{\mathbf{u}}^{N-1}) + \frac{3}{2}C(\overline{\mathbf{u}}^{N-1}) - \frac{1}{2}C(\overline{\mathbf{u}}^{N-2}) + \frac{1}{2}D(\mathbf{u}^* + \overline{\mathbf{u}}^{N-1}) + G\overline{P}^{N-1/2} = \mathbf{f}(t^{N-1/2}), \tag{9}$$

where N indicates the time step. C, D, and G are spatial operators for advection, diffusion, and pressure, respectively. $\mathbf{f}$ is a forcing term that must be determined such that $\overline{\mathbf{u}}^{N+1}$ satisfies the boundary conditions immersed in the computational domain. The resulting velocity field $\mathbf{u}^*$ is not divergence free, and the velocity and pressure have to be corrected by a pressure-correction term

$$\overline{\mathbf{u}}^{N+1} = \mathbf{u}^* - (\Delta t/\rho)\nabla\delta p, \tag{10}$$

$$\overline{P}^{N+1} = \overline{P}^N + \delta p, \tag{11}$$

where δp is the pressure correction, and it is found by solving the Poisson equation

$$\nabla^2\delta p = \frac{\rho}{\Delta t}\nabla\mathbf{u}^*. \tag{12}$$

3.2. Boundaries

[9] The ghost-cell method used here is based on the application of direct forcing at the immersed boundaries, an approach similar to that used by *Mohd-Yusof* [1997] and *Fadlun et al.* [2000]. The force is a function of time and space, and is defined such that the desired boundary conditions are satisfied in every time step. The location of this force does not necessarily coincide with a grid point, so interpolation or extrapolation to these points is needed. The magnitude of the forcing can be derived from the Navier-Stokes equations:

$$\frac{u_G^{N+1} - u_i^N}{\Delta t} = \mathrm{RHS}_i + f_i, \tag{13}$$

where RHS is the right-hand side of the Navier-Stokes equations and includes the advection, diffusion, and pressure terms. u_G^{N+1} is the velocity on the immersed boundaries, and f_i is the forcing function. The force f_i in equation (13) can be determined exactly if the unknowns are defined exactly at the boundaries. This is generally not the case in complex geometries. Interpolation and extrapolation techniques to determine the forcing term on the ghost cells can be classified into two categories: (1) schemes that allocate the ghost cells inside the fluid in the vicinity of the boundary [*Balaras*, 2004] and (2) techniques that consider the first cell in the "solid," i.e., outside the fluid domain, as the ghost cell [*Tseng and Ferziger*, 2003].

[10] In the present study, the former category is used, i.e., the ghost cells are located in the fluid in the vicinity of the boundary. They are defined as cells that have at least one neighbor cell located in the solid. To classify the nodes, an integer flag is used: nodes with flag –1 are located in the solid, nodes with flag 1 in the fluid. Ghost nodes have flag zero. There are several ways to interpolate the velocities for the ghost cells to satisfy the boundary conditions. In the present work, the ghost cell is located on the bed normal vector passing through it, and it forms the imaginary point I as shown in Figure 1. The imaginary point is extended into the fluid to find the required nodes (in fluid) for a trilinear (bilinear in 2-D) interpolation. In the case, some of required nodes still fall in the solid, the extension can be continued. Later on, the logarithmic interpolation of equation (8) between point I and the bed is applied to interpolate the values for the ghost points.

[11] In the above algorithm, boundary conditions for the pressure near the interface are not imposed explicitly, but they are essentially implicit in the source term of the Poisson equation [*Balaras*, 2004].

3.3. Mesh Refinement

[12] Another problem with a Cartesian grid, on top of the complications arising from the immersed boundaries, is the problem of resolution. A simple Cartesian grid treats all portions of flow equally; all cells in the grid are squares or cubes of the same size. A simple-structured Cartesian grid requires a large number of cells to capture the small eddies in a turbulent flow, whence a fully structured Cartesian grid can be computationally demanding. To resolve real-life geometries and flows, some form of grid adaptation may therefore be necessary.

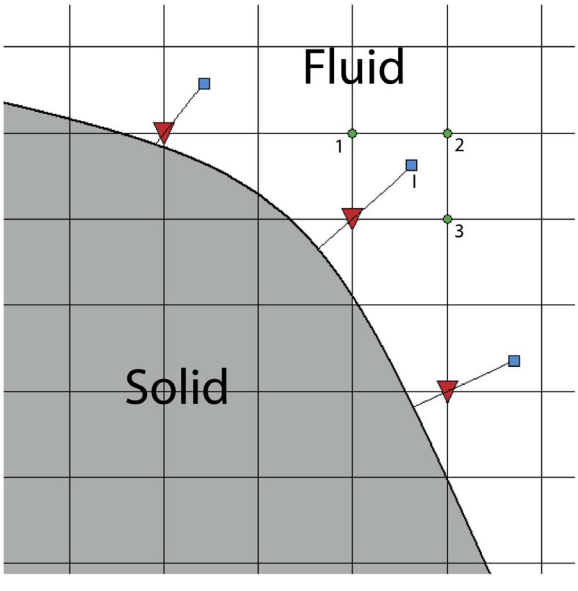

Figure 1. Interpolation stencil in 2-D. The ghost node "I" is interpolated bilinearly using the nearest points in the fluid.

[13] To optimize the demand for computational resources, we use an adaptive multilevel Cartesian mesh with local refinement, such that more grids cells can be placed in high-gradient regions such as boundary layers. Figure 2 shows such kind of grid over a dune. The grid generation process starts from a relatively coarse Cartesian grid (a base grid) encompassing the overall domain. The grid generation then proceeds by successive refinements. Each coarse cell is divided into eight equal children cells (four children in 2-D) to form the next level of refinement (see Figure 2). The corresponding data structure is discussed in the next section.

3.4. Data Structure

[14] For a typical structured-grid code, the data structure can be contained entirely in an (i,j,k) index. Unstructured grid codes usually have a significantly more complex data structure. As the connectivity of such a grid is of prime importance, information must be stored on which cells are neighbors of a given cell.

[15] There are two common types of data structure for this type of refined Cartesian grids: the hierarchical tree structure and the fully unstructured approach. In view of drawbacks associated with fully unstructured grids, the hierarchical tree data structure is chosen here. The finer grids are considered as subdomains of the coarser grids, which are treated separately as a kind of nested grids. Figure 3 shows a fully unstructured grid and a multilevel grid.

[16] *Coirier and Powell* [1996] used the hierarchical tree structure in their Euler and Navier-Stokes solver, demonstrating that the tree structure provides a logical means of finding cell-to-cell connectivity and allows straightforward isotropic refinement and coarsening through tree growth and pruning.

[17] The solution of viscous incompressible flows requires that careful attention be paid to the velocity-pressure coupling to prevent nonphysical oscillations in the solution. This is normally accomplished by using a staggered-grid method. This requires calculation and storage of face-based data. The storage and management of faces between cell neighbors requires the definition of new tree structures for faces. The tree structures for faces are managed in a comparable way as those for the cells. Three kinds of faces (perpendicular to *x*-, *y*-, and *z*-coordinates) require three extra trees. In a 3-D mesh, the cells form an octree, and the faces form three quad-tree data structures.

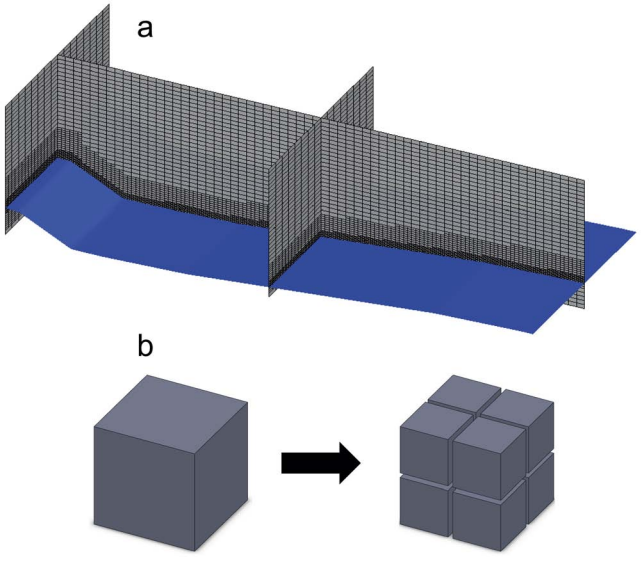

Figure 2. (a) Multilevel Cartesian grid over a dune. (b) The cells of the coarse grid are divided into eight children to form the finer grid.

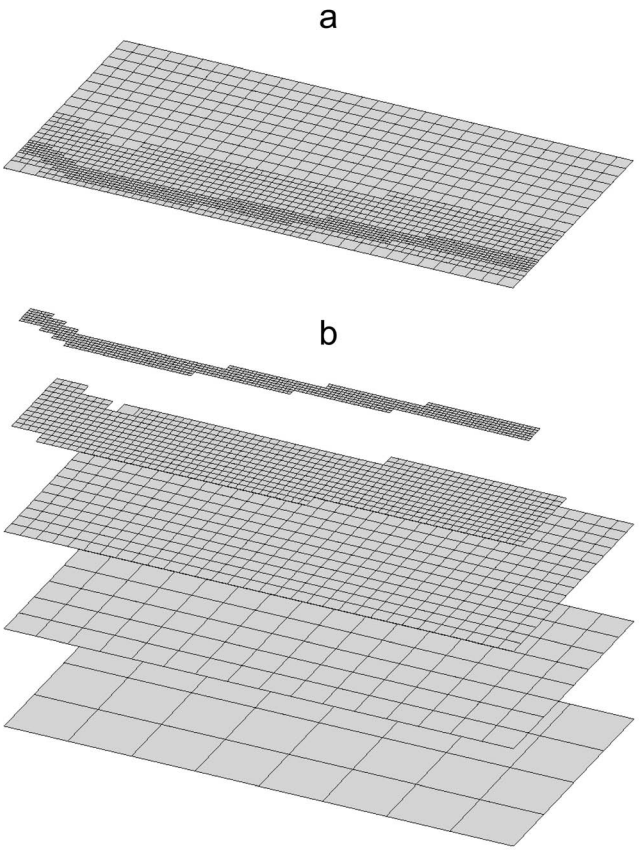

Figure 3. A fully (a) unstructured grid and (b) a multilevel grid.

[18] Figure 4 shows the structure of cell-cell connections in 2-D. Furthermore, the boundary conditions for any grid level are defined by ghost cells and ghost faces. The ghost cells and ghost faces are managed in such a way that they surround all subdomains. The values on the ghost cells and ghost faces can be interpolated from the values on the coarser grid. A high-order interpolation is necessary to conserve the order of accuracy. We implemented a third-order interpolation to conserve second-order accuracy.

[19] There is no explicit treatment necessary for interfaces in the current model. For fully unstructured grid, the interface between two grids has to be treated explicitly, in such a way that the mass stay conserved. Therefore, special treatment for the flux is necessary. As we apply a multilevel grid, the grid is decomposed into several locally independent grids as shown in Figure 3. Each grid is solved independently with the boundary conditions in its interface which are interpolated from the coarser grid. Although we

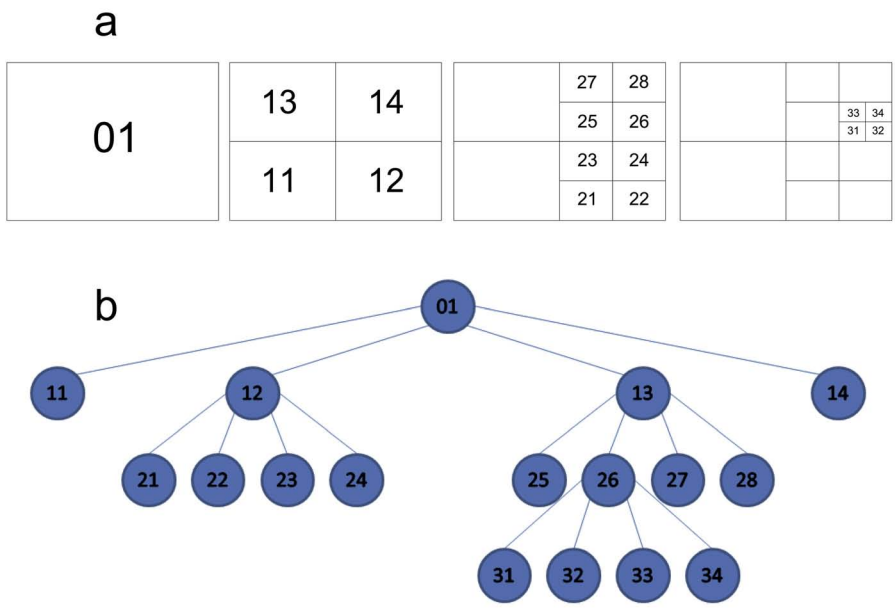

Figure 4. (a) Cell refinement and (b) data structure for multi-level Cartesian grid. A tree data structure is defined to identify the intergrid connections.

say independently, but in fact they are not independent, and they are linked by restriction and prolongation operators in the multigriding process.

[20] However, some papers claim that the order of accuracy decreases at the interfaces between the grids, because the gradients of variables are not continuous there, although the variables stay continuous. They solve this deficiency by a third-order interpolation of the fluxes between the fine and course grids to smooth the gradient. In our model, we solved this deficiency by increasing the order of interpolation for restriction operator (changed from second to fourth order). This method implicitly corrects the gradient of the variable on the grids interfaces. Although the number of operations will increase for a higher-order interpolation, but we still believe that the model becomes more efficient as the searching process for the suitable cells for interpolation (and decide which call can be used for a better performance) in former approach can be a more time-consuming process.

[21] By this kind of data structure, the parent-children connectivity can be directly employed to find the related cells and faces on the finer and coarser grids, which makes the implementation of a multigrid method straightforward and does not have the problem of hanging nodes breaching the symmetry of coefficient matrices, as may be the case with fully unstructured grids.

3.5. Moving Boundaries

[22] The general algorithm outlined in section 3.1 is directly applicable to moving boundary problems, such as alluvial processes. For moving boundary problems, the forcing approach and the related interpolations must be re-evaluated every time the location of the interface is updated. Moreover, complications may arise that are usually related to the time advancement scheme. This is due to the fact that the role of the grid points near the interface changes from time step to time step, as the solid body moves through the fixed grid. Thus, a forcing point may change to a fluid or a solid point, or conversely.

[23] The solution of the momentum equation at time step n requires physical values of the velocity vector and pressure, as well as their derivatives from time step $n-1$ at all fluid points. If the interface changes location, the required values from time step $n-1$ may be nonphysical. Due to Courant-Friedrichs-Lewy (CFL) restriction of the present scheme the boundaries cannot move by more than one grid cell in each time step, which results in three possible changes in the flags of the points near the interface,

[24] 1. The interface does not pass any grid point. In this case, the grid points do not change role and no stability problems arise.

[25] 2. The interface moves toward the fluid and passes a forcing point. The forcing point becomes solid and the nearest fluid point becomes a forcing point. Because the new solid point has a history from the previous time step, no stability problems arise.

[26] 3. The interface moves toward the solid and passes a grid point. The forcing point now becomes fluid and the next solid point becomes a forcing point. Because the new forcing point has no history from the previous time step, stability problems may arise.

[27] Situation 3 is shown in Figure 5 for a 2-D case. In the left part of Figure 5 the derivative $\partial uv/\partial y$ makes use of velocities v_1 and v_2. The values of v_1 and v_2 are both physical because they are located outside the solid. In a later stage, the boundary moves toward the solid (see the right Figure 5) and the role of u_2 changes (from forcing to fluid point). The same derivative now makes use of v_3 and v_4, while v_3 is located in the solid and has no physical value. To avoid such instabilities, *Yang and Balaras* [2006] applied a field-extension procedure in which the velocity and pressure fields are extrapolated to the first point in the solid phase (pseudofluid points) at the end of each

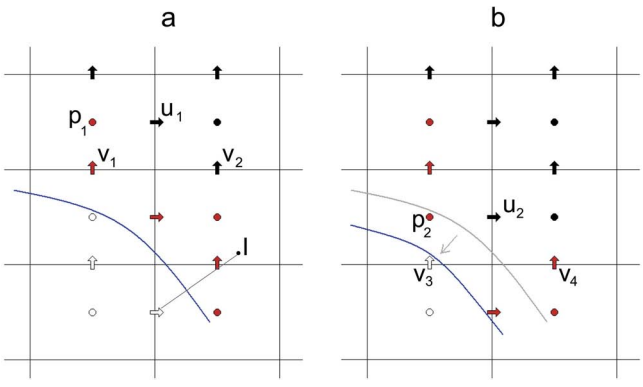

Figure 5. (a) Moving the boundary can bring nonphysical values for the derivatives in momentum equations. (b) The role of u changes from ghost cell to fluid cell, and it gives nonphysical values of $\partial vu/\partial y$. The white, black, and red colors clarify the solid points, the fluid points, and the forcing points, respectively.

deformation. They constructed the values of the velocity at the pseudofluid points using a procedure that is similar to the procedure applied by *Tseng and Ferziger* [2003]. However, a test of the present model with and without field extension showed no significant difference in the results. Therefore, we stick to the forcing procedure described in section 3.1, also after deformation of the model domain.

[28] In ghost-cell immersed-boundary methods, the pressure has to be interpolated around the velocity nodes. The interpolated pressure is to be used in momentum equations and as mentioned earlier, the solution of pressure Poisson equation does not need a forcing approach. Unlike the velocities, the pressure at the immersed boundaries cannot be determined explicitly. The boundary condition for pressure is of the Neumann type and can be expressed in normal derivative form as

$$\frac{\partial p}{\partial n} = 0, \tag{14}$$

with the normal derivative to be found from:

$$\frac{\partial p}{\partial n} = \frac{\partial p}{\partial x_i} n_i, \tag{15}$$

where n_i's are the components of the normal unit vector, $\vec{n}$.

[29] Applying an interpolation procedure similar to that for the velocities to an image point of pressure located in the fluid, the pressure value in the ghost cell can be determined as

$$p_p = p_I, \tag{16}$$

where p_p is the ghost-cell pressure and p_I is the pressure at the image point.

[30] We should also note that, due to the staggering of the mesh, the selection of points on the pressure grid is not only based on their relative location with respect to the interface but also on their association to velocity values that are non-physical. As a result, the pressure is also extended to nodes on the pressure grid that would have been classified as boundary nodes on the velocity grid.

3.6. Solution Techniques

[31] Full approximation storage (FAS) multigrid cycles have been implemented to solve the momentum and pressure correction equations, with the following time-stepping algorithm.

[32] 1. Solve the momentum equations
(a) Apply one FAS cycle for momentum in x direction.
(b) Apply one FAS cycle for momentum in y direction.
(c) Apply one FAS cycle for momentum in z direction.
(d) Calculate the scalar fields, like turbulence closure.
(e) Check the residual. If it is not converged, go to (a).

[33] 2. Apply FAS cycles for the pressure correction until convergence.

[34] 3. Correct the velocity field and pressure.

[35] 4. Reforce the flow beside the moving solid boundaries (or apply field extension).

[36] 5. Go to step 1.

[37] FAS multigrid for the pressure correction is found to be more stable than the linear multigrid. It is probably because of the presence of hanging nodes that bring complexity as well as nonlinearity to the existing interpolations. The local iteration process in step 1 is very fast if an implicit-explicit method (i.e., the Crank-Nicolson-Adams-Bashforth method) is used. Fully explicit methods are restricted to small CFL numbers, and they can be very expensive. Fully implicit methods, on the other hand, are unconditionally stable, but they require more iterations to achieve convergence. It is possible to apply large time steps in fully implicit methods, but turbulence involves multiple time scales and the required time step is not allowed to exceed its smallest time scale. Hence, fully implicit techniques with small time steps (in the order of the Kolmogorov time scale) can be more expensive than implicit-explicit methods. The code is fully parallelized by OpenMP on shared memory computers, and an efficiency of 75% is achieved.

4. Numerical Experiments

[38] To validate the present model, we simulated different cases that have been extensively documented in literature. We started from determining the order of accuracy of the model by solving Wannier flow. Subsequently, uniform flow around a cylinder and flow over fixed 2-D and 3-D dunes were simulated and compared with published numerical and experimental results.

4.1. Wannier Flow

[39] To verify the order of accuracy of the model including ghost cells at immersed boundaries, we simulated Stokes flow around a cylinder in the vicinity of a moving wall. This problem has been solved analytically by *Wannier* [1950]. Wannier obtained the exact solution for Stokes flow around a spinning cylinder located inside a larger rotating cylinder. This problem can be simplified by an infinite increase in the radius of the outer cylinder. The outer cylinder thus changes its role to an infinite horizontal wall moving in its own plane. Therefore, the flow is bounded by the wall on one side and extends to infinity in other directions.

Here we consider the case where the inner cylinder is not spinning.

[40] The flow is calculated for three different grid resolutions, namely 16×16, 32×32, and 64×64 as base grids. All grids are locally refined by two more levels around the cylinder and the moving wall (3-levels grids). To minimize the effect of slip errors, which are more significant at low Reynolds numbers, the boundary is reforced after the correction in the fractional step method. The velocity field and the grid structure are shown in Figure 6. The results of the current simulation have been compared with Wannier's relations, and the $\mathcal{L}_2$ norm errors in the velocities have been plotted in Figure 7, which shows a convergence rate of slightly more than 2, i.e., second-order accuracy of the velocities.

4.2. Flow Around a Circular Cylinder

[41] The simulation of flow around a cylinder is an interesting test case, because its behavior can change dramatically as a function of Reynolds number. At high Reynolds numbers, the flow separates and forms a von Kármán vortex street. This flow has been studied extensively, numerically, as well as experimentally.

[42] We performed simulations at three different Reynolds numbers, $Re = 20$, 40, and 100, based on the free-stream velocity and the diameter of the cylinder. This flow case has been studied previously by *Tritton* [1959], *Dennis and Chang* [1970], *Coutanceau and Bouard* [1977], *Fornberg* [1980], *Braza et al.* [1986], *Liu et al.* [1998], *Ye et al.* [1999], *Kim et al.* [2001], *Calhoun* [2002], *Tseng and Ferziger* [2003], *Russell and Wang* [2003], *Pan* [2006], and *Xu and Wang* [2006].

[43] The nondimensional far-field stream velocity was held constant at $U_\infty = 1$ m s^{-1}. The base calculations were done on a 1024×512 grid. Figures 8 and 9 show the streamlines and the vorticity contours, respectively, for the $Re = 40$ case. Figure 10 shows the expected trailing von Kármán vortex street at $Re = 100$. This particular simulation was run to a nondimensional time of 150 to ensure full development of the quasi steady state condition.

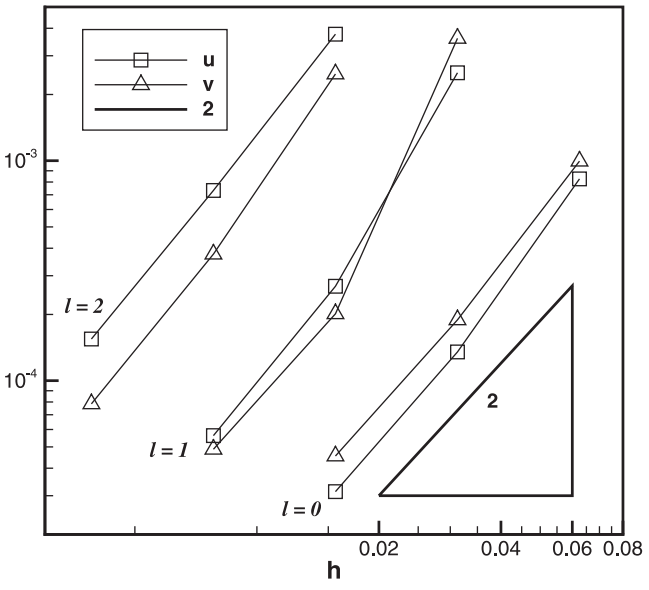

Figure 7. Error in the $\mathcal{L}_2$ -norm for the velocities in Wannier flow with source. $h = \Delta x = \Delta y$, and l indicates the level number.

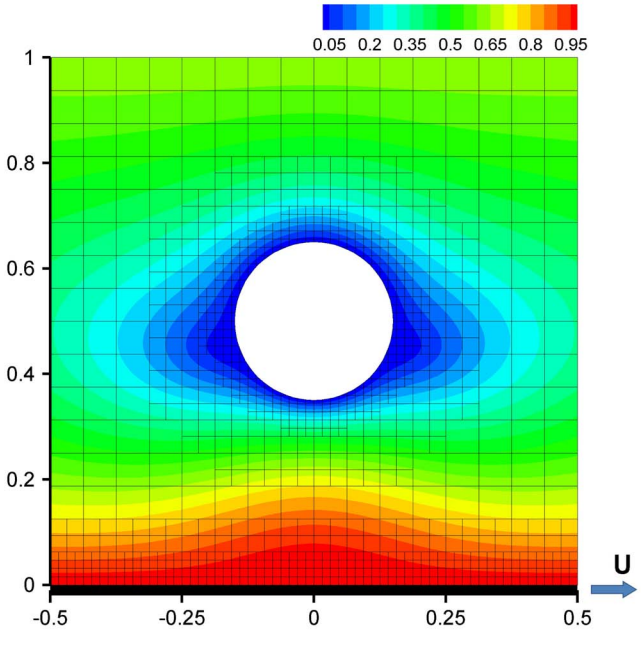

Figure 6. Computational domain and velocity field for Wannier flow. The grid is presented in a coarser form for visualization purposes.

[44] Table 1 shows a summary of results for the cases with steady-state solution for two different Reynolds numbers, $Re = 20$ and 40, in terms of trailing bubble length, angle of separation, and drag coefficient. Table 2 shows the results for $Re = 100$ in terms of drag and lift coefficients. These geometric results compare favorably with both experimental and previous computational results.

4.3. Flow Over a Flat Bed

[45] To investigate the efficiency of the current model and compare with the standard structured grid models, a 3-D flow on a flat bed is simulated. Two cases, namely with local refining and without local refining, are considered. The size of the cells in the case without local refining was equal to the size of the finest cells in the case with local refining. The local refining was made by a base grid of $32 \times 32 \times 16$ and constructing three extra finer levels close to the bed. Table 3 shows the necessary time for simulating

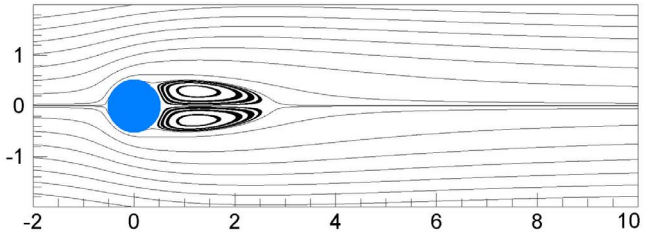

Figure 8. Streamlines for $Re = 40$ cylinder.

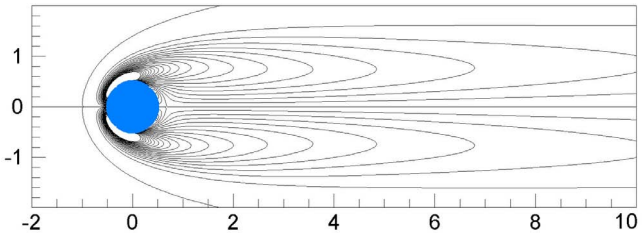

Figure 9. Vorticity for $Re = 40$ cylinder. Contour values are $-4 : 0.2 : 4$.

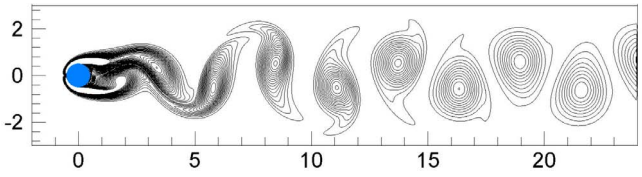

Figure 10. Vortex street behind cylinder at $Re = 100$. Contour values are $-2.5 : 0.1 : 2.5$.

Table 1. Summary of Results for $Re = 20$ and $Re = 40$[a]

	$Re = 20$			$Re = 40$		
	LZ	$L\theta$	C_D	L	θ	C_D
Ye et al. [1999]	0.9		2.0	2.3		1.5
Tseng and Ferziger [2003]				2.21		1.53
Pan [2006]	0.85		2.01	2.13		1.5
Xu and Wang [2006]	0.92	44.2	2.23	2.21	53.5	1.66
Tritton [1959]			2.22			1.48
Coutanceau and Bouard [1977]	0.73	42.3		1.89	52.8	
Fornberg [1980]	0.91		2.00	2.24		1.50
Dennis and Chang [1970]	0.94	43.7	2.05	2.35	53.8	1.52
Calhoun [2002]	0.91	45.5	2.19	2.18	54.2	1.62
Russell and Wang [2003]	0.94	43.3	2.13	2.29	53.1	1.60
Present	0.93	44.2	2.21	2.22	53.5	1.58

[a]L is the trailing bubble length, θ is the angle of separation, and C_D is the drag coefficient.

Table 2. Summary of Results for $Re = 100$[a]

	C_D	C_L	St
Braza et al. [1986]	1.36 ± 0.015	±0.250	
Liu et al. [1998]	1.35 ± 0.012	±0.339	0.164
Calhoun [2002]	1.33 ± 0.014	±0.298	0.175
Russell and Wang [2003]	1.38 ± 0.007	±0.300	0.169
Present	1.33 ± 0.013	±0.315	0.170

[a]C_D, C_L, and St are the drag coefficient, lift coefficient, and Strouhal number, respectively.

Table 3. Comparison of the Time Necessary Simulating the First 20 Time Steps

	Number of Cells	Time (s)
With refining	8,388,608	1940.7
Without refining	507,392	501

the first 20 time steps for both cases. The case without refining needed almost 16 times more cells to achieve the same accuracy of the case with local refining. Although local refining decreases the number of cells dramatically, it can be seen that the time of the simulation is decreased by a factor of 4, because the interpolations for the boundaries between the grid interfaces are time consuming.

[46] The efficiency of the current model is strongly dependent on the grid topology and the number of interface cells, which has to be interpolated from their coarser levels. The current example is a simple case to show an approximate efficiency of the model. If the area of the interfaces between the find and coarse grids increase, more cells are necessary to be interpolated, and the efficiency probably slightly decreases.

4.4. Flow Over Fixed 2-D Dune

[47] To validate the model for irregular bedforms, we simulated flow over 2-D dunes and, on this basis, we studied the flow over the dunes. *McLean et al.* [1994] conducted a series of experiments for the flow over 2-D dunes. The experimental flume was 22 m long, 0.9 m wide, and 0.9 m deep. Twenty fixed dunes of 0.8 m length and 0.04 m height were placed in the flume. The stoss side of the dunes was a half-cosine wave running from its lowest point at the trough to its highest point at the crest (Figure 11). The angle of the lee slope was 30°. Three flow conditions were investigated, with flow depths varying from 0.158 to 0.546 m. Table 4 shows the experimental conditions.

[48] We simulated runs 2 and 3 with the same flow conditions and bed geometry as in the experiments. The boundaries in streamwise and transverse directions were taken periodic. A pressure gradient was applied in streamwise direction to mimic the slope force. The value of the pressure gradient was calculated in such a way that the desired bulk velocity was achieved. For runs 2 and 3, the simulations are done on base grids of $128 \times 64 \times 32$ and $128 \times 64 \times 64$ in streamwise, spanwise, and normal direction, respectively, with one extra level of refining beside the bed. Figure 12

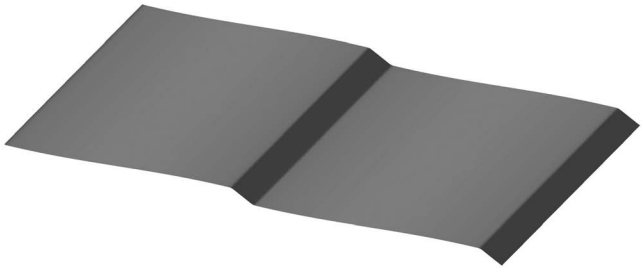

Figure 11. Schematic representation of the 2-D fixed, artificial dunes. Flow over the dunes runs from left to right.

Table 4. Experimental Conditions for 2-D Dunes from *McLean et al.* [1994][a]

Run	h (m)	U (m s^{-1})	Re	Fr	τ_T (Pa)	u_*^T (m s^{-1})	C_f	S ($\times 10^4$)
1	0.210	0.482	1.01×10^5	0.336	1.49	0.0386	0.0064	8.3
2	0.158	0.377	0.60×10^5	0.303	1.20	0.0346	0.0084	9.4
3	0.546	0.284	1.55×10^5	0.123	0.45	0.0212	0.0056	0.81

[a] h is the average water depth, U is the averaged velocity, Re is the Reynolds number, Fr is the Froude number, τ_T is the bed shear stress, u_*^T is the shear velocity, C_f is the total drag coefficient, and S is the channel slope.

shows the computed instantaneous streamwise velocity on this dune for runs 2 and 3, and Figure 13 the spatially and temporally averaged streamwise velocity. It can be seen that the circulation zone is small in run 2, whereas a relatively large recirculation zone is formed behind the lee side of the dunes in run 3. Figure 14 shows the instantaneous vorticity for these two runs. The vorticity behind the dune crest in the bottom figure seems to exhibit a boiling effect. The boiling is generated at the stoss side of the dune and separates from the crest into the flow. In a later stage, the boiling reaches the free surface and produces a small deformation in the free surface [*Best*, 2005]. Figure 15 shows the turbulence kinetic energy (TKE). The maximum TKE for run 2 is located close to the dunes crest, whereas the maximum in run 3 is found around the recirculation zone. The region of high TKE is larger in run 3, but the maximum value is much higher in run 2. Apparently, the large recirculation zone in run 3 is more effective in transferring turbulence to the region around it.

[49] To visualize the coherent structures of the flow, we followed the approach developed by *Hunt et al.* [1988], based on the second invariant of ∇u:

$$Q = -\frac{1}{2}\frac{\partial u_i}{\partial x_j}\frac{\partial u_j}{\partial x_i} = -\frac{1}{2}(|S|^2 - |\Omega|^2), \tag{17}$$

where S and Ω are the symmetric and antisymmetric components of ∇u. Figure 16 shows the coherent structure above the dune for runs 2 and 3 by plotting the isosurface $Q = 8\,\mathrm{s}^{-2}$. Clearly, the turbulence structures in run 2 are

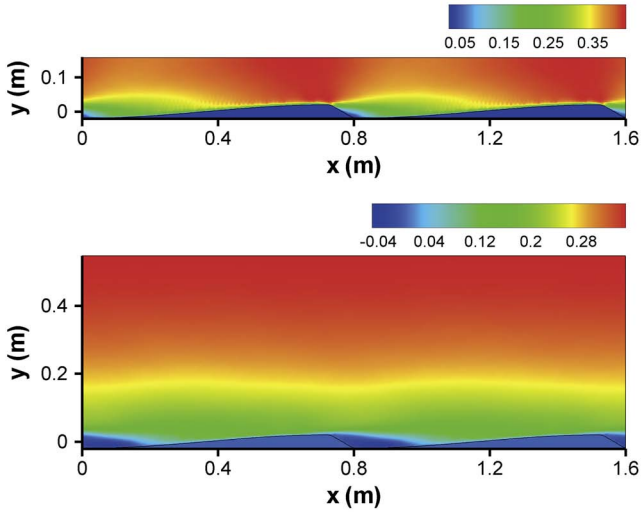

Figure 13. Spatially and temporally averaged streamwise velocity (m s^{-1}) for runs (top) 2 and (bottom) 3. Run 3 shows a larger recirculation than run 2.

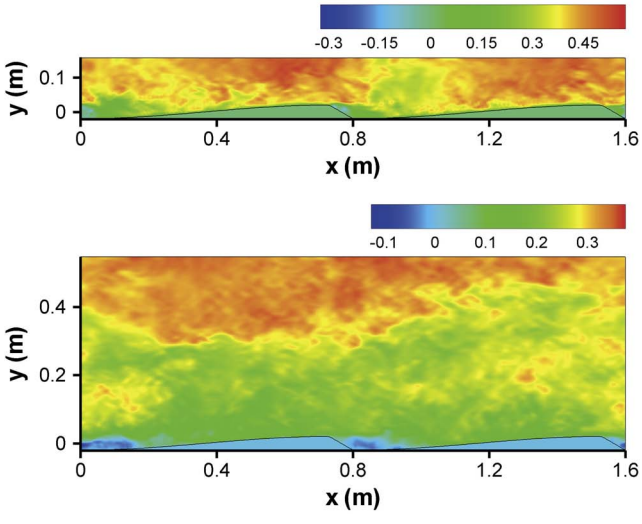

Figure 12. Instantaneous streamwise velocity (m s^{-1}) for runs (top) 2 and (bottom) 3.

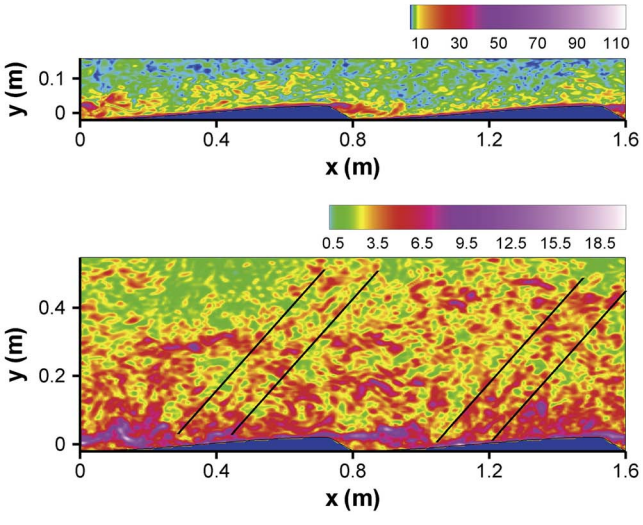

Figure 14. Instantaneous vorticity for runs (top) 2 and (bottom) 3. The boiling phenomena can be observed in run 3.

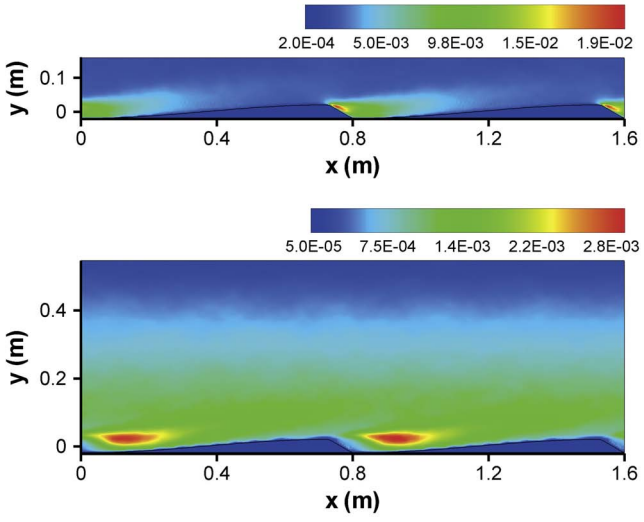

Figure 15. Spatially and temporally averaged TKE ($m^2\ s^{-2}$) for runs (top) 2 and (bottom) 3. TKE has a larger value in run 2, but the region of high TKE is larger in run 3.

smaller than those in run 3. In run 3, the recirculation zone generates large eddies that break down to smaller eddies on the stoss side of the next dune. Given the small recirculation zone in run 2, no large eddies can be generated by the recirculation zone. To show this phenomenon more clearly, the coherent structure is plotted in Figure 17 for a larger value of Q ($Q = 20\ s^{-2}$), thus filtering out the smaller structures. Figure 17 shows that in run 3 large structures are formed behind the lee side of dunes, whereas in run 2 the

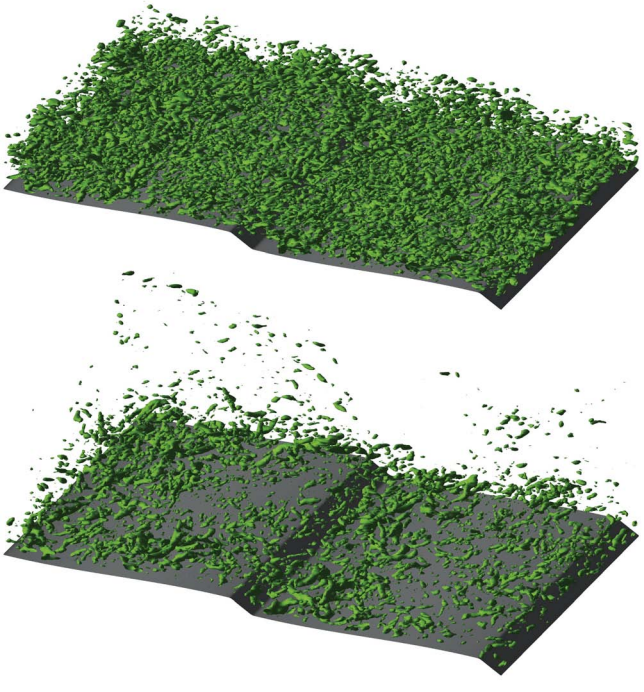

Figure 16. Coherent structures of turbulence over the dunes for runs 2 and 3 for $Q = 8\ s^{-1}$.

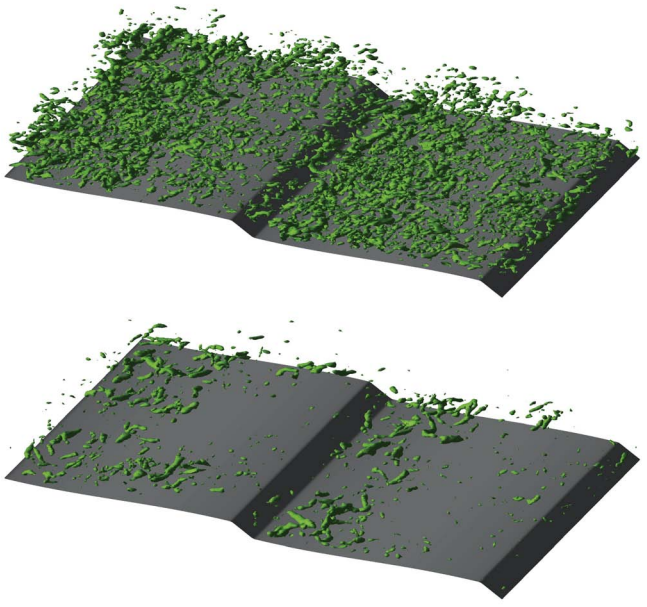

Figure 17. Coherent structures of turbulence over the dunes for runs 2 and 3 for $Q = 20\ s^{-1}$.

smaller structures still dominate. Further downstream, these highly 3-D structures interact with the wall and generate disturbances with the same spanwise wavelength. These disturbances evolve into "horseshoe" vortical structures. Figure 18 shows such kind of structures behind the lee side of a dune and on the stoss side of the next dune. Both legs of the horseshoe structure are associated with high levels of positive and negative streamwise vorticity (right and left leg, respectively, facing in the flow direction). As the two legs meet along the tip, regions of high spanwise vorticity are formed. As a result, strong vertical fluid motions are

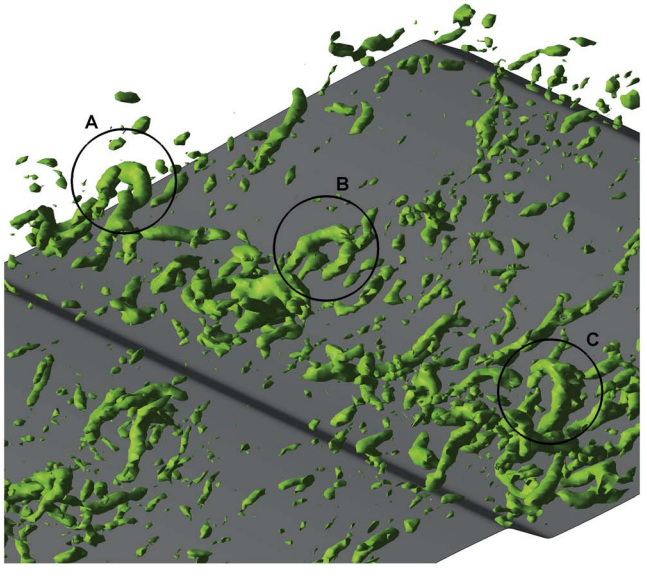

Figure 18. Horseshoe structures on the stoss side of dune in run 3 for $Q = 12\ s^{-1}$.

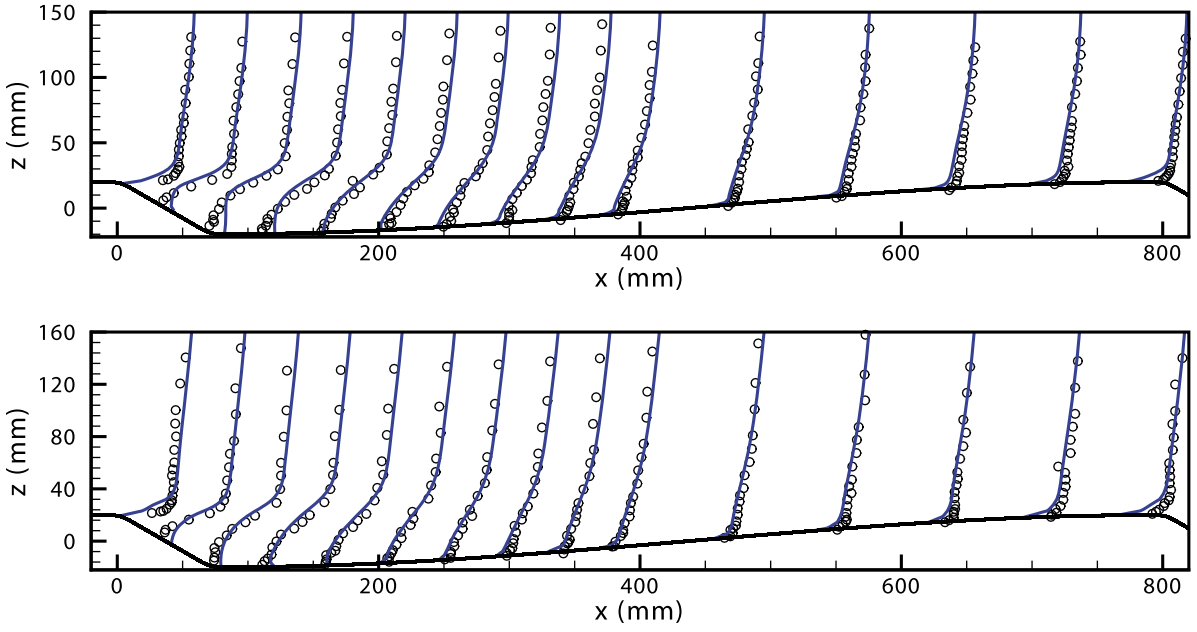

Figure 19. Comparison of nondimensional time-averaged streamwise velocity profiles over 2-D dune for (top) case 2 and (bottom) case 3. The circles are the measurements, and the solid lines are the simulated profiles.

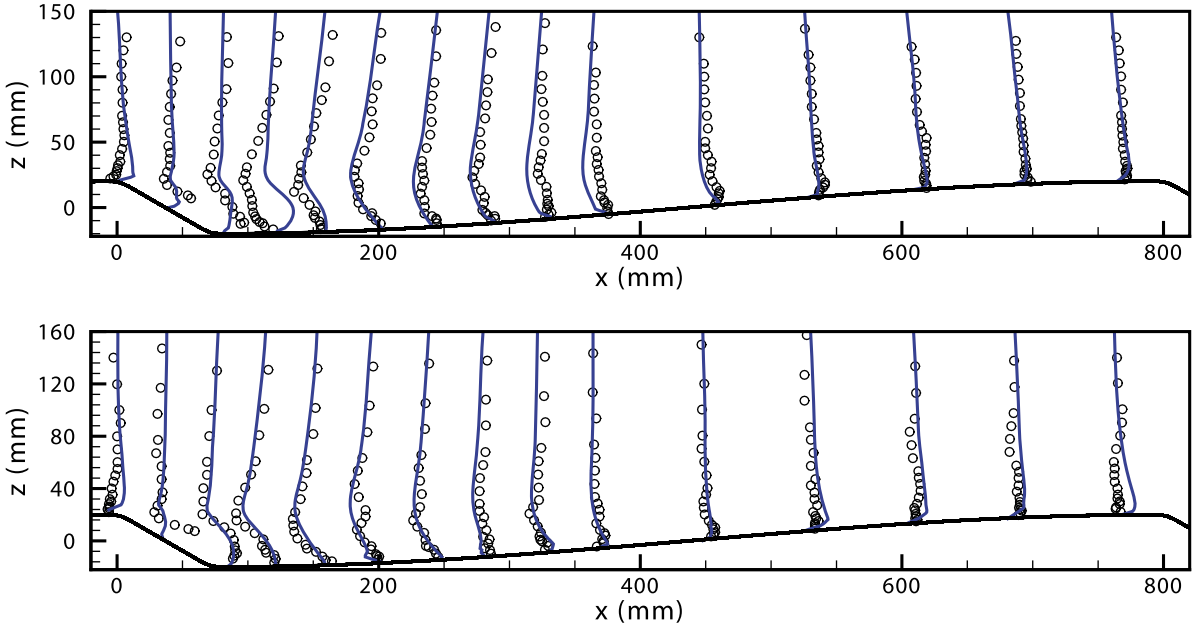

Figure 20. Comparison of nondimensional time-averaged vertical velocity profiles over 2-D dune for (top) case 2 and (bottom) case 3. The circles are the measurements, and the solid lines are the simulated profiles.

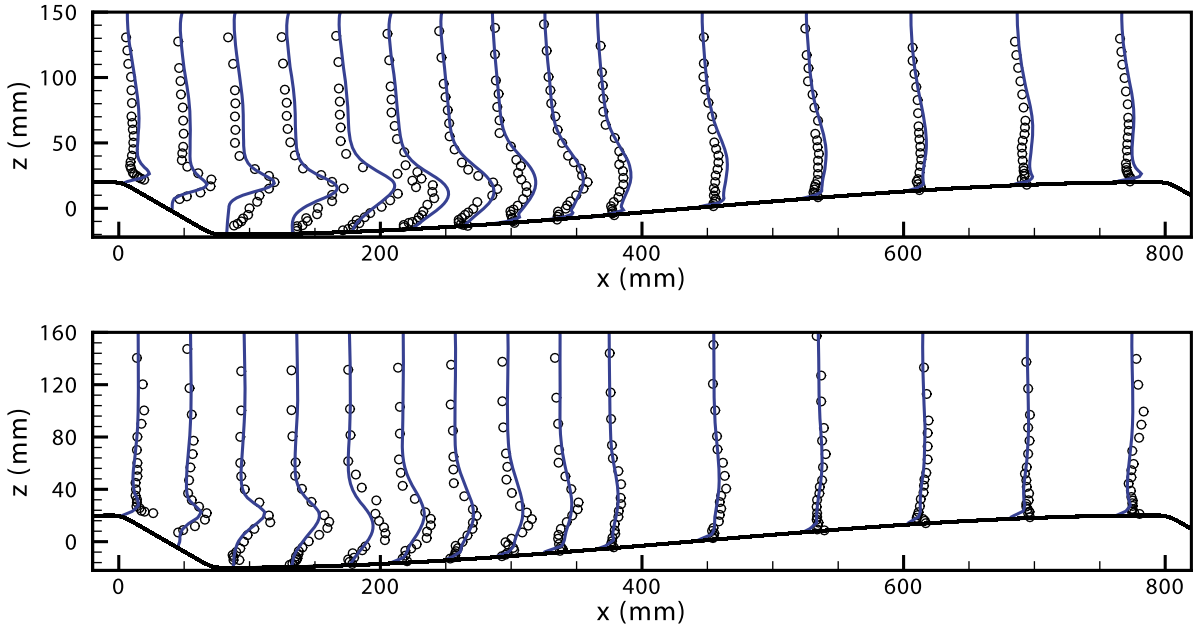

Figure 21. Comparison of nondimensional time-averaged Reynolds stress profiles of $-\overline{u'u'}$ over 2-D dune for (top) case 2 and (bottom) case 3. The circles are the measurements, and the solid lines are the simulated profiles.

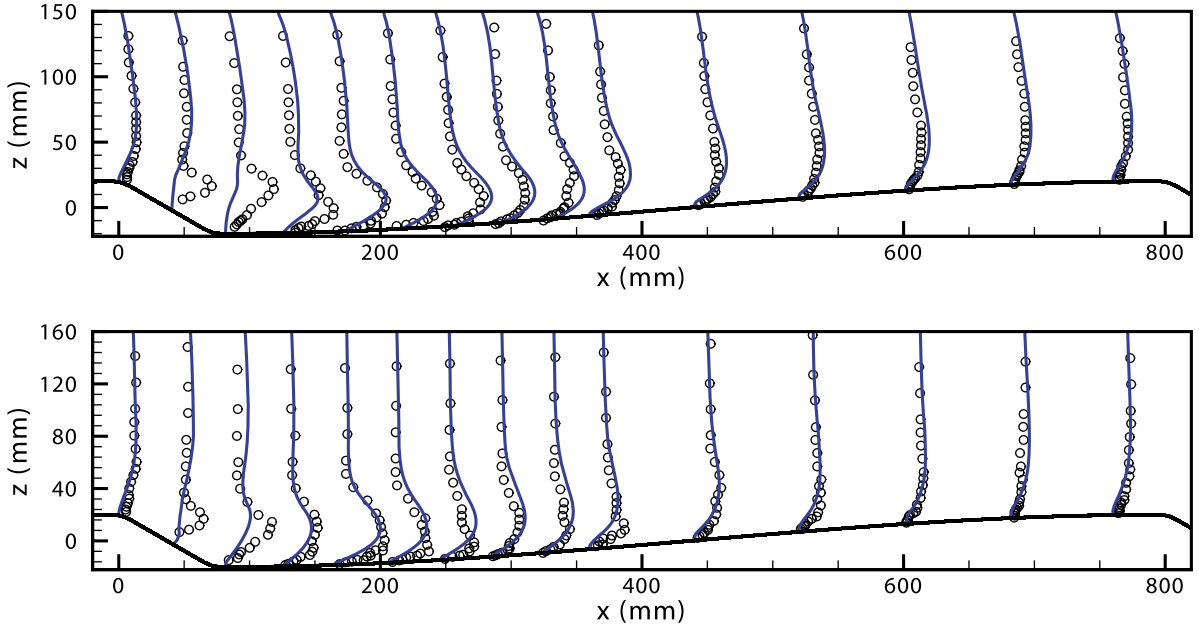

Figure 22. Comparison of nondimensional time-averaged Reynolds stress profiles of $-\overline{w'w'}$ over 2-D dune for (top) case 2 and (bottom) case 3. The circles are the measurements, and the solid lines are the simulated profiles.

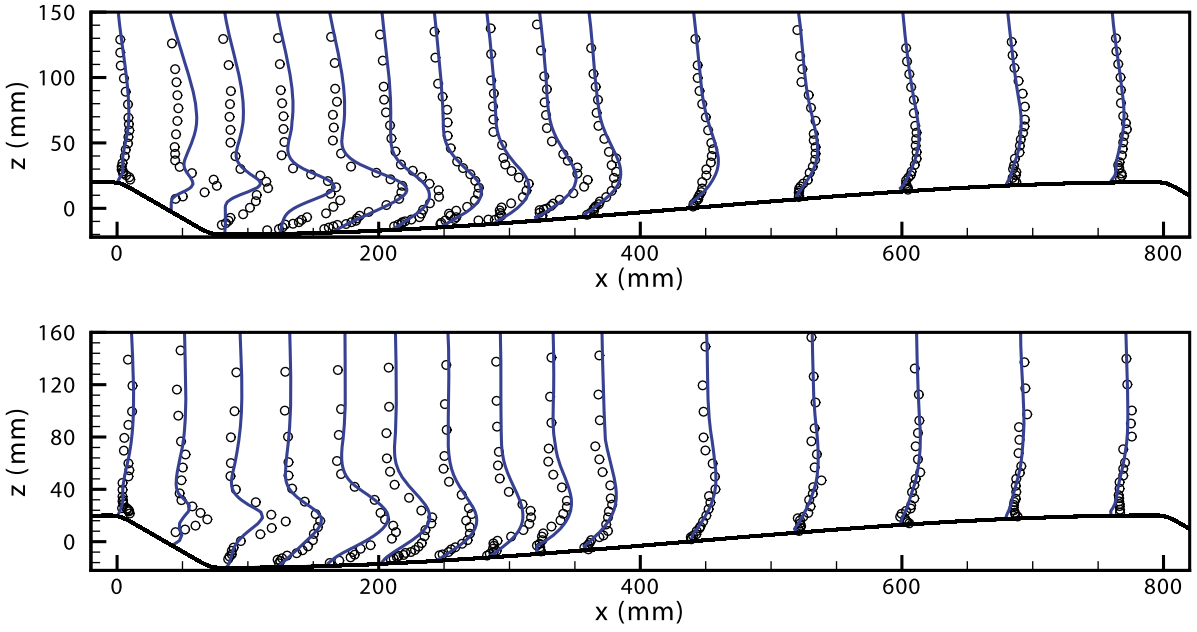

Figure 23. Comparison of nondimensional time-averaged Reynolds stress profiles of $-\overline{u'w'}$ over 2-D dune for (top) case 2 and (bottom) case 3. The circles are the measurements, and the solid lines are the simulated profiles.

formed between the legs and close to the tip, lifting fluid from the wall region. In some instances, the intensity of these vertical fluid upwellings is as strong as 30% of the bulk flow velocity [*Grigoriadis et al.*, 2009]. This phenomenon has a strong effect on the pickup of sediment and hence on the reworking of the bed.

[50] Figure 19 shows the time-averaged streamwise velocity normalized by the averaged bed shear velocity u_*^T (averaged over time and space) and compared with McLean's experiments. The normalized velocities in run 2 are generally slightly larger than those in run 3; however, considering the differences in depth and Froude number, the shapes of the individual profiles agree fairly with the experiments. Profiles of the vertical velocity for runs 2 and 3 are presented in Figure 20. An overprediction of the vertical velocity can be observed in the recirculation zone for run 2. An overprediction occurs mainly in the recirculation region, as can be expected when using wall functions that are valid only for attached boundary layers. The agreement also holds for the Reynolds stress components shown in Figures 21–23. An underprediction in Reynolds shear stress $\overline{w'w'}$ can be observed in the recirculation zone for runs 2 and 3 in Figure 22, in which w' is the fluctuation of the velocity in vertical direction.

4.5. Flow Over Fixed 3-D Dune

[51] The interaction between the flow field and the bedforms is a complex one, involving as-yet poorly understood feedback mechanisms. One of these mechanisms is the effect the bedforms have on the turbulent flow field. Due to the limitations of turbulent flow measurements made over 3-D dunes, the information will be limited to measurements made over 2-D features of various types, including strongly asymmetric steep-angle dunes, relatively symmetric low-angled dunes, and spatially accelerated flows. *Maddux et al.* [2003a, 2003b] has performed a series of experiments on fixed 3-D dunes. The configurations of these experiments are identical with the experiments on 2-D dunes performed by *McLean et al.* [1994]. Maddux's dunes were qualitatively similar to real sinuous-crested 3-D dunes, as observed in the field and in flumes with mobile sediments [*Gabel*, 1993]. The dunes had a mean wavelength λ of 0.8 m and a mean height δ at the crest of 0.04 m. The stoss side of the dunes was a half-cosine wave running from its low point at the trough to its high point at the crest. The angle of the lee slope was 30°. The mean height, wavelength, and cross section of these dunes matched that of 2-D bedforms used in experiments conducted by *McLean et al.* [1994]. The three-dimensionality of the dunes was expressed as a full cosine wave in the spanwise direction, superimposed on the profile of the 2-D dunes (Figure 24). The height of the crests of the dunes above the troughs, δ, varied in the spanwise direction

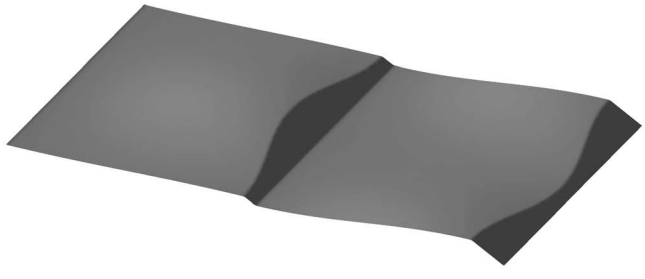

Figure 24. Schematic representation of the 3-D fixed, artificial dunes. Flow over the dunes runs from left to right.

Table 5. Experimental Conditions for 3-D Dunes from *Maddux et al.* [2003a][a]

Run	h (m)	U (m s^{-1})	Re	Fr	τ_T (Pa)	u_*^T (m s^{-1})	C_f	S ($\times 10^4$)
$T2$	0.173	0.357	0.62×10^5	0.275	1.71	0.0413	0.0134	10.6
$T3$	0.561	0.261	1.46×10^5	0.111	0.46	0.0215	0.0068	2.40

[a] h is the average water depth, U is the averaged velocity, Re is the Reynolds number, Fr is the Froude number, τ_T is the bed shear stress, u_*^T is the shear velocity, C_f is the total drag coefficient, and S is the channel slope.

from 0.02 to 0.06 m. Successive crestlines were 180° out of phase, so that a dune with a high middle and low sides was followed immediately by a dune with a low middle and high sides, as shown in Figure 24. It resulted in a crest-to-crest wavelength, λ, that varied in the cross-stream direction from 0.73 to 0.87 m. The resulting steepness of the dunes varied in the spanwise direction as well, with δ/λ values ranging from 0.025 to 0.075. These values matched well with the values of δ/λ for real dunes in the field [*Maddux et al.*, 2003a]. Although typical 3-D dunes are often more complex, with concave faces and other features, these dunes are sufficiently realistic while maintaining enough similarity to previous measurements to facilitate an understanding of the response of the flow to 3-D bed topography. *Maddux et al.* [2003a] show a significant difference in turbulence structure and bed friction between 2-D and 3-D dunes. This difference is in such a degree for which the three dimensionality of the dunes must be taken seriously into account.

[52] *Maddux et al.* [2003a, 2003b] have extended the experiments over 3-D dunes for flow of the similar depth and discharge as for experiments 2 and 3 over 2-D dunes. Table 5 shows the conditions for Maddux's experiments, and they are named T2 and T3, related to McLean's experiments 2 and 3, respectively. The water depth and the bulk velocities for T2 and T3 are slightly different than those for experiments 2 and 3.

[53] Here we simulate the flow over 3-D dunes with the same condition for the experiment T2 and we call it R2. The boundaries in streamwise direction are set to periodic. Smooth solid boundary conditions in spanwise direction are imposed to account for the glass side walls in Maddux's experiment. The free surface is considered as a rigid lid, and the flow is derived by a pressure gradient to obtain the

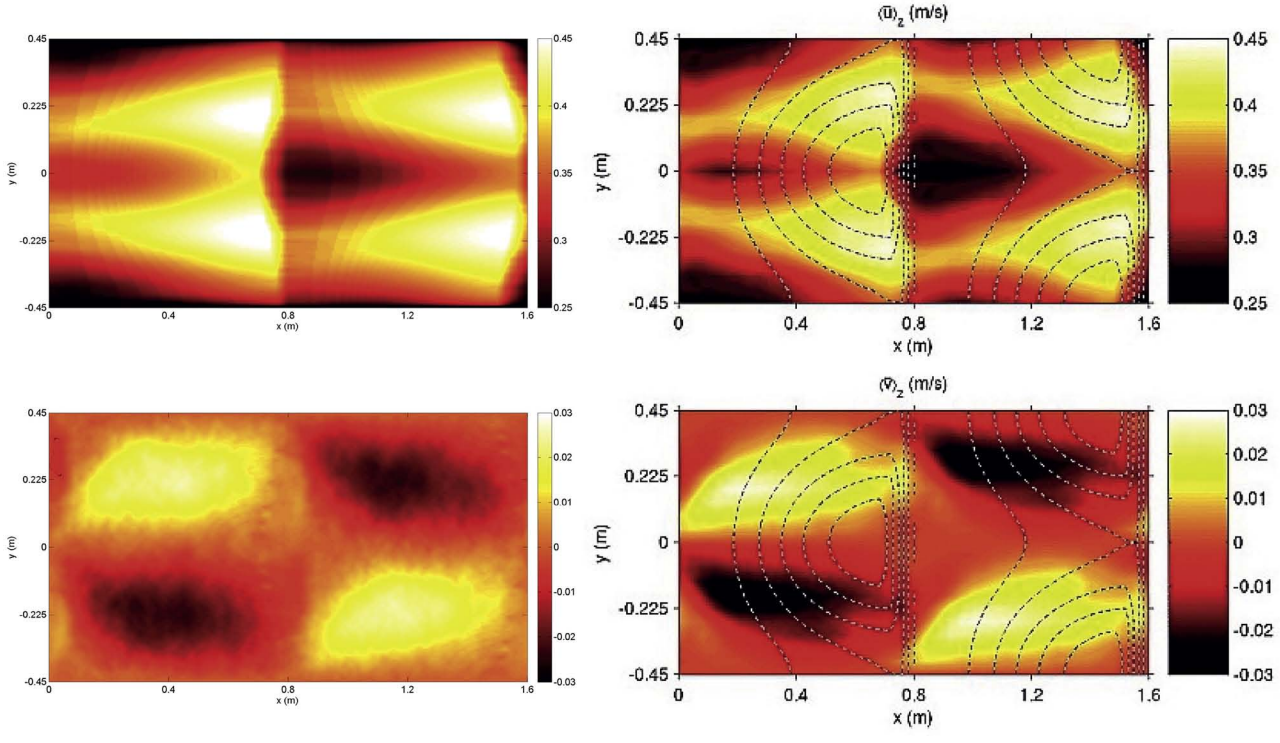

Figure 25. Computed depth-averaged (top left) streamwise and (bottom left) spanwise velocities for run R2, and comparison with T2 experiment from *Maddux et al.* [2003a] for (top right) streamwise (bottom right) and spanwise velocities.

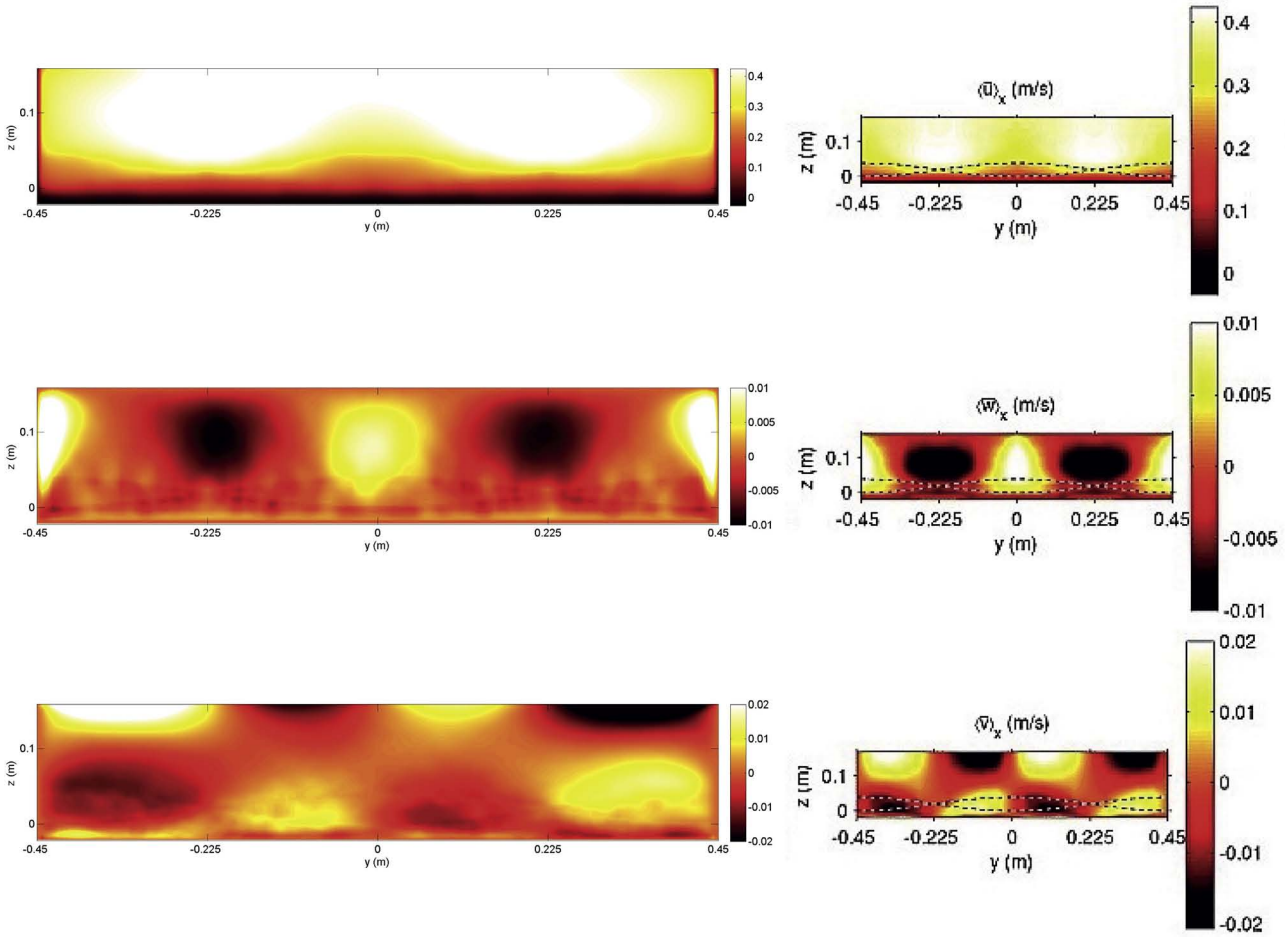

Figure 26. Computed streamwise-averaged velocities for (top left) streamwise, (middle left) normal, and (bottom left) spanwise velocities for run R2, and comparison with T2 experiment from *Maddux et al.* [2003a] for (top right) streamwise, (middle right) normal, and (bottom right) spanwise velocities.

desired bulk velocity. The bed is considered as a rough boundary with sediment diameter of 1 mm on the bed as it is imposed in T2 [*Maddux et al.*, 2003a]. The simulations are done on a base grid of 128 × 64 × 32 in streamwise, spanwise, and normal direction, with one extra level of refining beside the bed.

[54] Figure 25 shows the computed depth-averaged streamwise and spanwise velocities for run R2, and they are compared with the contour graphs from Maddux's experiment T2. A qualitatively good agreement can be seen between the computed results and Maddux's measurements. The highest streamwise velocity is located on the crest line of the dunes but not on the crest point. The highest streamwise velocity is located in the middle distances between the centerline and the walls, which is called as "node of the dune crest" in *Maddux et al.* [2003a]. The lowest streamwise velocity is located in the trough parts directly after the crest points because of weak velocity field in the recirculation zone behind the crest point. The largest absolute values of spanwise velocities are located on the halfway between the nodes of crest lines. The four quadrants of the spanwise velocity in Figure 25 are the results of the geometrical curvatures of the dunes.

[55] Figure 26 shows the streamwise-averaged velocities for run R2 and that for Maddux's experiment T2. It shows that the streamwise velocity has maximum values on the nodes of the crest, as it is also observed from Figure 25. For spanwise and normal velocities, a secondary current can be concluded, which can be more clearly seen in Figure 30. Further, the validations are extended to the Reynolds stresses. Figure 27 shows a comparison between Reynolds shear stresses for the current computations for run R2 and the measurement T2. All these figures show that the computed results qualitatively agree with the measurements.

[56] Figure 28 shows the instantaneous streamwise velocity in the streamwise center section and node sections and Figure 29 shows the time-averaged velocity vectors in the center section and node section for run R2. A relatively strong recirculation zone can be observed in the center

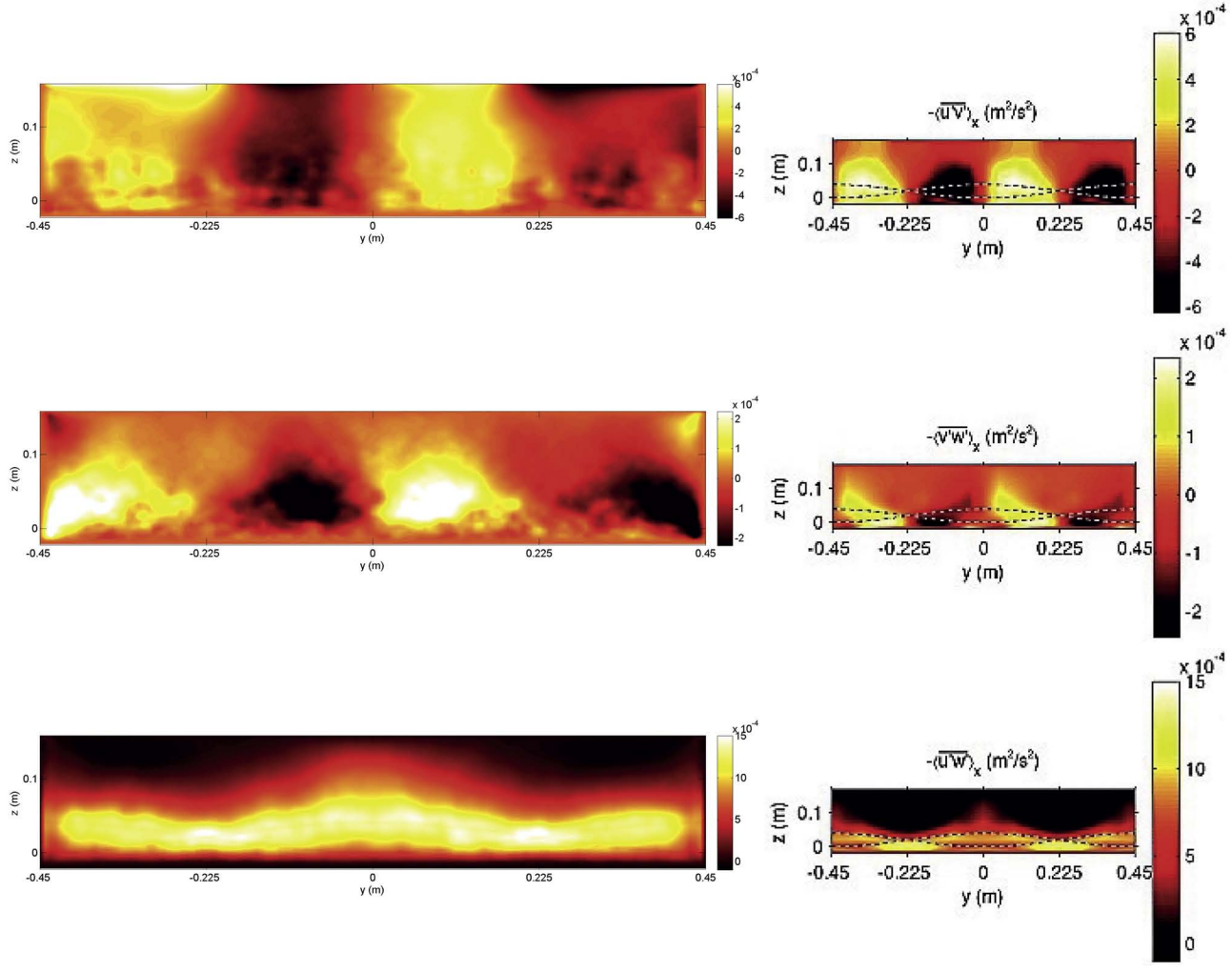

Figure 27. Computed streamwise-averaged Reynolds shear stresses (top left) $-\overline{u'v'}$, (middle left) $-\overline{v'w'}$, and (bottom left) $-\overline{u'w'}$ for run R2, and comparison with T2 experiment from *Maddux et al.* [2003a] for (top right) $-\overline{u'v'}$, (middle right) $-\overline{v'w'}$, and (bottom right) $-\overline{u'w'}$.

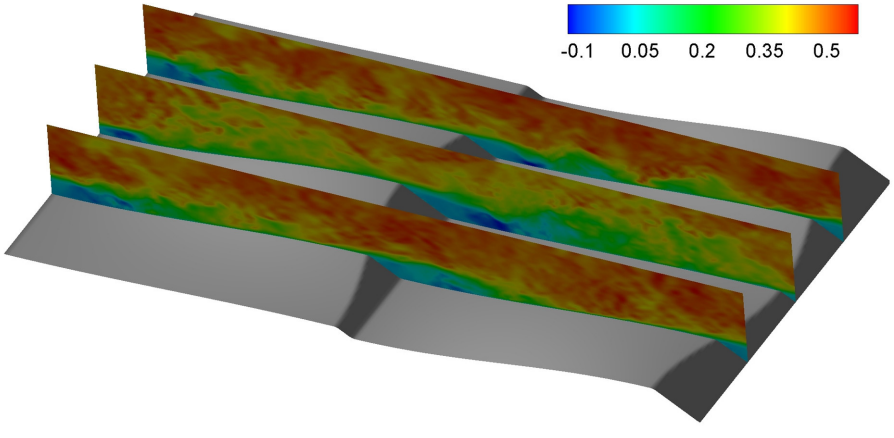

Figure 28. Instantaneous streamwise velocity (m s^{-1}) at the center section and node sections of the 3-D dune.

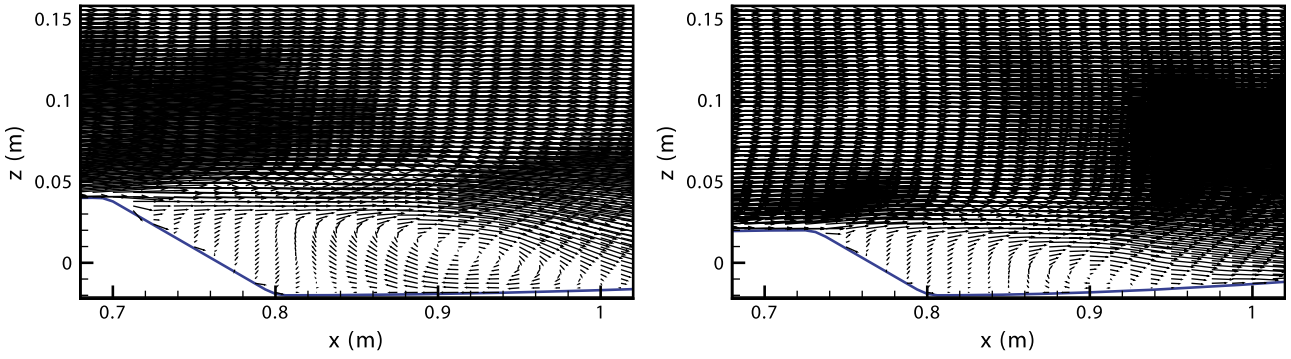

Figure 29. Time-averaged velocity vectors in the (left) streamwise central section and (right) streamwise node section.

section, whereas a weak recirculation exists in the node section. Although the streamwise velocity is stronger on the node of the crest line, the height of this point is much less than the crest point, and it makes the recirculation zone less strong. The size of recirculation zone is more affected by the height of the dune rather than the velocity distribution on the dune crest line.

[57] Figure 30 shows the time-averaged velocity vectors in the trough of the dune (in $x = 0.8$ m). Four counterrotating secondary circulation vectors, with secondary current velocities, can be observed. The loops are also observed in the experiment T2. These secondary currents appear to be generating a net downward momentum flux. This flux could be carrying momentum that is not being carried by low levels of turbulence over the 3-D dunes [*Maddux et al.*, 2003a].

[58] Figure 31 shows the instantaneous vorticity in the center section and on the node sections of the domain. The boiling effect can be less observed on the crest node, and the vorticity structures are pulled down. Although on the crest point, the vorticity is also in some contents pulled toward the bed, after the recirculation zone, the vorticity is separated again and goes toward the water surface. This trend is observed in all instantaneous vorticity profiles of the current run.

[59] Figure 32 shows the time-averaged TKE in the center section and node sections of the domain. A larger region of TKE can be observed behind the crest point than the node because of larger recirculation zone. As the recirculation zone gets smaller, the TKE gets concentrated in a smaller region, although the turbulence intensity of these two regions are comparable.

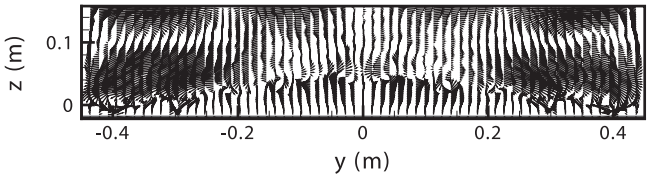

Figure 30. Time-averaged velocity vectors in the trough of the dune in $x = 0.8$ m. Four loops can be observed, which rotate in the direction opposite to their surrounding loops.

5. Conclusions

[60] We have developed an efficient numerical model for hydrodynamics in a cost-efficient way, to be applied for the simulation of morphodynamics. This model uses unstructured Cartesian grids with local refining to be able to capture the small scales turbulence with efficient computational costs. The grid can be refined in high-gradient regions as well as in the vicinity of the bed, in the form of an adaptive grid. The adaptivity can follow the movement of the bed to yield a high-resolution solution for the flow in complex moving bed topology. A ghost-cell immersed-boundary method is applied for the cells that intersect the boundaries. LES is employed to resolve the turbulence.

[61] At the first step, the hydrodynamics model is compared with an analytical solution of flow (Wannier flow) to validate the applied multilevel techniques as well as the applied ghost-cell immersed boundary technique. Comparing with Wannier's analytical solution, the results show a second order of accuracy for the velocities. Moreover, the flow fields around a 2-D cylinder at Reynolds numbers of 20, 40, and 100 are simulated, and they are compared with the published results, and a good agreement is found.

[62] The flow has been simulated on fixed 2-D dunes. A comparison with experimental measurements from previous studies shows a good agreement. The model has sufficient capability to calculate the physical phenomena such as turbulence coherent structures and boiling vortices. Moreover, the flow on 3-D dunes is solved, and the contours for velocities and Reynolds shear stresses are compared with the previous experimental studies from literature. The contours show qualitatively good agreement with the experimental measurements. Furthermore, the model captured the recirculation zone and the current of turbulence in a physics-based way. The model has a potential to be employed for simulation of sediment transport and morphodynamic evolution of an alluvial bed.

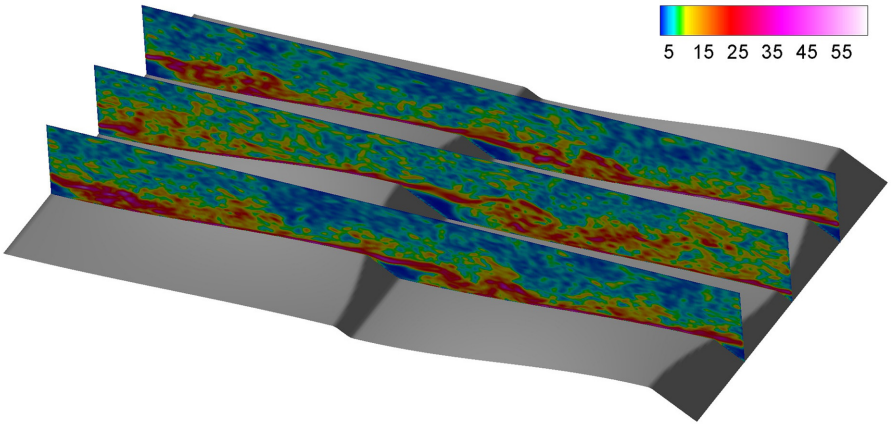

Figure 31. Instantaneous vorticity contours at the central and the node sections. For the central section, the vorticity attaches to the bed and is separated from the bed behind the recirculation zone, moving toward the free surface, whereas for the node sections, the vorticity sticks to the vicinity of the bed.

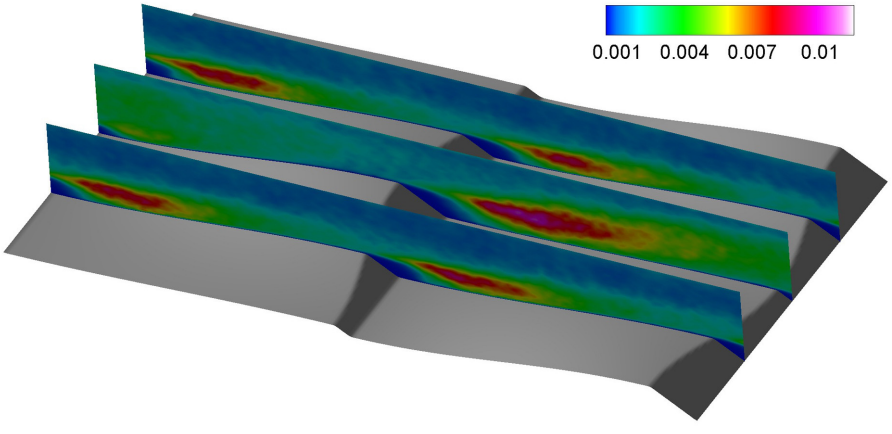

Figure 32. Time-averaged TKE ($m^2\ s^{-2}$) at the center section and node sections. A large region of TKE is formed behind the crest point, because of a larger recirculation zone.

[63] **Acknowledgments.** The work presented herein was carried out as part of the work package "River Morphology" of Delft Cluster project 4.30 "Safety against flooding," financed by the Netherlands government and supported by Rijkswaterstaat Waterdienst. The authors are grateful to Sanjay Giri for his fruitful suggestions.

References

Aftosmis, M. J., M. J. Berger, and J. E. Melton (1998), Robust and efficient Cartesian mesh generation for component-based geometry, *AIAA J.*, *36*(6), 952–960, doi:10.2514/2.464.

Balaras, E. (2004), Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations, *Comput. Fluids*, *33*, 375–404.

Bayyuk, S. (1996), Euler flows with arbitrary geometries and moving boundaries, Ph.D. dissertation, University of Michigan, Ann Arbor, Mich.

Berger, M. J., M. J. Aftosmis, and G. Adomavicius (2000), Parallel multigrid on Cartesian meshes with complex geometry, *Proceedings of the 8th Int. Conf. on Parallel CFD*, Norwegian Univ. of Sci., Trodheim, Norway.

Best, J. (2005), The fluid dynamics of river dunes: A review and some future research directions, *J. Geophys. Res.*, *110*, F04S02, doi:10.1029/2004JF000218.

Braza, M., P. Chassaing, and H. Ha Minh (1986), Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, *J. Fluid Mech.*, *165*, 79–130.

Calhoun, D. (2002), A cartesian grid method for solving the two-dimensional streamfunctionvorticity equations in irregular regions, *J. Comput. Phys.*, *176*(2), 231–275.

Clarke, D., M. Salas, and H. Hassan (1986), Euler calculations for multielement airfoils using Cartesian grids, *AIAA J.*, *24*(3), 353–358.

Coirier, W. J., and K. G. Powell (1996), Solution-adaptive Cartesian cell approach for viscous and inviscid flows, *AIAA J.*, *34*, 938–945.

Coutanceau, M., and R. Bouard (1977), Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation, *J. Fluid Mech.*, *79*(2), 231–256.

Dennis, S. C. R., and G. Z. Chang (1970), Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100, *J. Fluid Mech.*, *42*(3), 471–489.

Fadlun, E. A., R. Verzicco, P. Orlandi, and J. Mohd-Yusof (2000), Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *J. Comput. Phys.*, *161*, 35–60.

Fornberg, B. (1980), A numerical study of steady viscous flow past a circular cylinder, *J. Fluid Mech.*, *98*(4), 819–855.

Gabel, S. L. (1993), Geometry and kinematics of dunes during steady and unsteady flows in the Calamus River, Nebraska, USA, *Sedimentology*, *40*, 237–269.

Gilmanov, A., F. Sotiropoulos, and E. Balaras (2003), A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids, *J. Comput. Phys.*, *191*, 660–669.

Grigoriadis, D. G. E., E. Balaras, and A. A. Dimas (2009), Large-eddy simulations of unidirectional water flow over dunes, *J. Geophys. Res.*, *114*, F02022, doi:10.1029/2008JF001014.

Ham, F. E., F. S. Lien, and A. B. Strong (2002), A Cartesian grid method with transient anisotropic adaptation, *J. Comput. Phys.*, *179*, 469–494, doi:10.1006/jcph.2002.7067.

Hunt, J. C. R., A. A. Wray, and P. Moin (1988), Eddies, streams and convergence zones in turbulent flows, in *CTR Annual Research Briefs*, Rep. CTR-S88, NASA Ames, Stanford University, Stanford, Calif.

Iaccarino, G., and R. Verzicco (2003), Immersed boundary technique for turbulent flow simulations, *Appl. Mech. Rev.*, *56*(3), 331–347.

Kim, J., D. Kim, and H. Choi (2001), An immersed-boundary finite-volume method for simulations of flow in complex geometries, *J. Comput. Phys.*, *171*, 132–150.

Kirkpatrick, M. P., S. W. Armfield, and J. H. Kent (2003), A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional Cartesian grid, *J. Comput. Phys.*, *184*, 1–36.

Liu, C., X. Sheng, C. H. Sung, and J. H. Kent (1998), Preconditioned multigrid methods for unsteady incompressible flows, *J. Comput. Phys.*, *139*, 35–57.

Maddux, T. B., J. M. Nelson, and S. R. McLean (2003a), Turbulent flow over three-dimensional dunes: 1. Free surface and flow response, *J. Geophys. Res.*, *108*(F1), 6009, doi:10.1029/2003JF000017.

Maddux, T. B., S. R. McLean, and J. M. Nelson (2003b), Turbulent flow over three-dimensional dunes: 2. Fluid and bed stresses, *J. Geophys. Res.*, *108*(F1), 6010, doi:10.1029/2003JF000018.

Martin, D. F., P. Colella, and D. Graves (2008), A cell-centered adaptive projection method for the incompressible Navier-Stokes equations in three dimensions, *J. Comput. Phys.*, *227*(3), 1863–1886.

McLean, S. R., J. M. Nelson, and S. R. Wolfe (1994), Turbulence structure over two-dimensional bed forms: Implications for sediment transport, *J. Geophys. Res.*, *99*(*C6*), 12729–12747.

Mittal, R., H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, and A. von Loebbecke (2008), A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *J. Comput. Phys.*, *227*, 4825–4852.

Mohd-Yusof, J. (1997), Combined immersed boundaries/B-splines methods for simulations of flows in complex geometries, *CTR Annual Research Briefs*, pp. 317–327, NASA Ames/Stanford University, Stanford.

Pan, D. (2006), An immersed boundary method for incompressible flows using volume of body function, *Int. J. Numer. Methods Fluids*, *50*, 733–750.

Quirk, J. J. (1994), An alternative to unstructured grids for computing gas dynamic flows around arbitrary complex two-dimensional bodies, *Comput. Fluids*, *23*, 125–142.

Saad, Y. (2003), *Iterative Methods for Sparse Linear Systems*, 531 pp., Soc. for Ind. and Appl. Math., Philadelphia, Pa.

Russell, D., and Z. J. Wang (2003), A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow, *J. Comput. Phys.*, *191*, 177–205.

Smagorinsky, J. (1963), General circulation experiments with the primitive equations, *Mon. Weather Rev.*, *91*(3), 99–164.

Tritton, D. J. (1959), Experiments on the flow past a circular cylinder at low Reynolds numbers, *J. Fluid Mech.*, *6*(4), 547–567.

Tseng, Y. H., and J. H. Ferziger (2003), A ghost-cell immersed boundary method for flow in complex geometry, *J. Comput. Phys.*, *192*, 593–623.

Udaykumar, H. S., R. Mittal, P. Rampunggoon, and A. Khanna (2001), A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, *J. Comput. Phys.*, *174*, 345–380.

Wannier, G. H. (1950), A contribution to the hydrodynamics of lubrication, *Q. Appl. Math.*, *8*, 1–32.

Waymel, F., F. Monnoyer, and M. J. P. William-Louis (2006), Numerical simulation of the unsteady three-dimensional flow in confined domains crossed by moving bodies, *Comput. Fluids*, *35*, 525–543.

Xu, S., and Z. J. Wang (2006), An immersed interface method for simulating the interaction of a fluid with moving boundaries, *J. Comput. Phys.*, *216*, 454–493.

Yang, J., and E. Balaras (2006), An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, *J. Comput. Phys.*, *215*, 12–40.

Ye, T., R. Mittal, H. S. Udaykumar, and W. Shyy (1999), An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *J. Comput. Phys.*, *156*, 209–240.

Zeeuw, D. D., and K. G. Powell (1993), An adaptive refined Cartesian mesh solver for the Euler equations, *J. Comput. Phys.*, *104*, 56–68.