LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-CONF-234276

Development of an Immersed Boundary Method to Resolve Complex Terrain in the Weather Research and Forecasting Model

K. A. Lunquist, F. K. Chow, J. K. Lundquist, J. D. Mirocha

September 5, 2007

AMS 7th Symposium on the Urban Environment
San Diego, CA, United States
September 10, 2007 through September 13, 2007

11.1 DEVELOPMENT OF AN IMMERSED BOUNDARY METHOD TO RESOLVE COMPLEX TERRAIN IN THE WEATHER RESEARCH AND FORECASTING MODEL

Katherine A. Lundquist[1]*, Fotini K. Chow [2], Julie K. Lundquist [3], and Jeffery D. Mirocha [3]
[1] Mechanical Engineering, University of California, Berkeley, CA
[2] Civil and Environmental Engineering, University of California, Berkeley, CA
[3] Atmospheric Flow, Transport, and Hazard Assessment Group,
Lawrence Livermore National Laboratory, Livermore, CA

1. INTRODUCTION

Flow and dispersion processes in urban areas are profoundly influenced by the presence of buildings which divert mean flow, affect surface heating and cooling, and alter the structure of turbulence in the lower atmosphere. Accurate prediction of velocity, temperature, and turbulent kinetic energy fields are necessary for determining the transport and dispersion of scalars. Correct predictions of scalar concentrations are vital in densely populated urban areas where they are used to aid in emergency response planning for accidental or intentional releases of hazardous substances.

Traditionally, urban flow simulations have been performed by computational fluid dynamics (CFD) codes which can accommodate the geometric complexity inherent to urban landscapes. In these types of models the grid is aligned with the solid boundaries, and the boundary conditions are applied to the computational nodes coincident with the surface. If the CFD code uses a structured curvilinear mesh, then time-consuming manual manipulation is needed to ensure that the mesh conforms to the solid boundaries while minimizing skewness. If the CFD code uses an unstructured grid, then the solver cannot be optimized for the underlying data structure which takes an irregular form. Unstructured solvers are therefore often slower and more memory intensive than their structured counterparts. Additionally, urban-scale CFD models are often forced at lateral boundaries with idealized flow, neglecting dynamic forcing due to synoptic scale weather patterns. These CFD codes solve the incompressible Navier-Stokes equations and include limited options for representing atmospheric processes such as surface fluxes and moisture. Traditional CFD codes therefore posses several drawbacks, due to the expense of either creating the grid or solving the resulting algebraic system of equations, and due to the idealized boundary conditions and the lack of full atmospheric physics.

Meso-scale atmospheric boundary layer simulations, on the other hand, are performed by numerical weather prediction (NWP) codes, which cannot handle the geometry of the urban landscape, but do provide a more complete representation of atmospheric physics. NWP codes typically use structured grids with terrain-following vertical coordinates, include a full suite of atmospheric physics parameterizations, and allow for dynamic synoptic scale lateral forcing through grid nesting. Terrain-following grids are unsuitable for urban terrain, as steep terrain gradients cause extreme distortion of the computational cells.

In this work, we introduce and develop an immersed boundary method (IBM) to allow the favorable properties of a numerical weather prediction code to be combined with the ability to handle complex terrain. IBM uses a non-conforming structured grid, and allows solid boundaries to pass through the computational cells. As the terrain passes through the mesh in an arbitrary manner, the main goal of the IBM is to apply the boundary condition on the interior of the domain as accurately as possible. With the implementation of the IBM, numerical weather prediction codes can be used to explicitly resolve urban terrain. Heterogeneous urban domains using the IBM can be nested into larger mesoscale domains using a terrain-following coordinate. The larger mesoscale domain provides lateral boundary conditions to the urban domain with the correct forcing, allowing seamless integration between mesoscale and urban scale models. Further discussion of the scope of this project is given by Lundquist et al. [2007].

The current paper describes the implementation of an IBM into the Weather Research and Fore-

* *Corresponding author address*: Katherine A. Lundquist, Mechanical Engineering, 6189 Etcheverry Hall, University of California, Berkeley, 94720-1742, email: katielundquist@berkeley.edu

casting (WRF) model, which is an open source numerical weather prediction code. The WRF model solves the non-hydrostatic compressible Navier-Stokes equations, and employs an isobaric terrain-following vertical coordinate. Many types of IB methods have been developed by researchers; a comprehensive review can be found in Mittal and Iaccarino [2005]. To the authors' knowledge, this is the first IBM approach that is able to use a pressure-based coordinate. The immersed boundary method presented here uses direct forcing, first suggested by Mohd-Yusof [1997], to impose a no-slip boundary condition. Additionally, the WRF model has been modified to include a no-slip bottom boundary condition enabling direct comparisons with the IBM solution for problems with gently sloping terrain. The accuracy and efficiency of the immersed boundary solver is examined within the context of a two-dimensional Witch of Agnesi hill. Results are also presented for two-dimensional flow over several blocks of New York City, which demonstrate the IB method's ability to handle extremely complex terrain with sharp corners and steep terrain gradients.

2. NUMERICAL METHODOLOGY

The solver for the WRF model is described in this section. Details are included for the governing equations, spatial and temporal discretiztion, and boundary conditions. Following this, details are given about the changes required to add the additional option of a no-slip bottom boundary condition to WRF. Finally, the modifications needed to include the effects of the immersed boundary are discussed.

2.1 THE WEATHER RESEARCH AND FORECASTING MODEL

WRF is an open source community model that is designed for a variety of purposes ranging from operational weather forecasting to idealized geophysical flow simulations. The software is designed to be flexible and modular, which facilitates development of the code by the broad academic community. Currently there are two dynamics solvers that will operate within the WRF framework. The core known as Advanced Research WRF (ARW) was used for these simulations, and is the focus of the following discussion. ARW has primarily developed by the National Center for Atmospheric Research (NCAR), and as of 2007, they continue to develop, maintain, and distribute the code.

ARW is a conservative finite-difference model that solves the non-hydrostatic compressible Navier-Stokes equations given by equations (1).

$$\partial_t \vec{V} + \vec{V} \cdot \nabla \vec{V} + \alpha \nabla p + \vec{g} = \vec{F} \tag{1a}$$

$$\partial_t \rho + \nabla \cdot (\rho \vec{V}) = 0 \tag{1b}$$

Here α is the specific volume, and $\vec{F}$ includes Coriolis effects and any additional forcing terms such as turbulent mixing or model physics. Additionally, WRF uses a terrain-following hydrostatic pressure coordinate allowing pressure to replace height as an independent variable. Laprise [1992] developed a transformation of the fully-compressible non-hydrostatic Euler equations into a terrain-following isobaric coordinate, forming the foundation of the governing equations found in WRF.

The vertical pressure or mass coordinate η is given in terms of the dry hydrostatic pressure P_{hs}, and defined such that it is zero at the top of the model, and unity at the surface of the terrain. The mass of the fluid in the column per unit area is then denoted by μ. This yields the coordinate definition $\eta = \frac{P_{hs} - P_{hs\,top}}{\mu}$, where $\mu(x,y) = P_{hs\,surface} - P_{hs\,top}$. Once the transformation to this coordinate system is applied, the strong conservation form of the Navier-Stokes equations takes the form given in (2). For reference, these equations are also given in the NCAR technical note on WRF [Skamarock et al., 2005, Section 2.2], where the notation differs slightly.

$$\partial_t \mu + \nabla \cdot (\mu \vec{V}) + \partial_\eta (\mu \dot{\eta}) = 0 \tag{2a}$$

$$\partial_t (\mu \vec{V}) + \nabla \cdot (\mu \vec{V}; \vec{V}) + \partial_\eta (\mu \dot{\eta} \vec{V}) - \nabla (p \partial_\eta \phi) + \partial_\eta (p \nabla \phi) = \vec{F} \tag{2b}$$

$$\partial_t (\mu w) + \nabla \cdot (\mu \vec{V} w) + \partial_\eta (\mu \dot{\eta} w) - g \left(\partial_\eta p - \mu\right) = F \tag{2c}$$

In the set of equations (2) the velocity vector $\vec{V}$ includes the horizontal velocities, and Del operates in the horizontal dimensions. The variable ϕ is the geopotential, and is defined as $\phi = gz$. Additionally the dot notation in $\dot{\eta}$ denotes differentiation with respect to time, and the semicolon notation represents the dyadic product.

In addition to the conservation of mass and momentum, an equation for potential temperature is solved. Potential temperature θ is a conserved quantity when the atmosphere is assumed to be adiabatic, so the governing equation takes the form

used for a conserved scalar:

$$\partial_t(\mu\theta) + \nabla \cdot (\mu\vec{V}\theta) + \partial_\eta(\mu\dot{\eta}\theta) = F_\theta \qquad (3)$$

Pressure is then diagnosed from the equation of state below, where γ_{dry} is the ratio of heat capacities of dry air C_p/C_v, p_o is the surface pressure, and R_{dry} is the universal gas constant:

$$p = p_o \left(\frac{R_{dry}\theta}{p_o \alpha_{dry}} \right)^{\gamma_{dry}} \qquad (4)$$

Before the discrete solver is constructed, the governing equations are recast in perturbation form for a viscous and moist atmosphere. Perturbation variables are advantageous for reducing truncation errors. A complete derivation beginning with the differential form of the compressible Navier-Stokes equations and ending with the perturbation form solved in WRF can be found in Lundquist [2006].

WRF is spatially discretized using an Arakawa-C staggered grid. Uniform grid spacing is used in the horizontal directions, and the terrain-following grid may be stretched in the vertical direction. A time split integration scheme is used to deal with the full range of frequencies admitted by the compressible Navier-Stokes equations. In this scheme a third order explicit Runge-Kutta method is used for time advancement of meterologically significant low frequency physical modes, while a smaller time step is needed to account for the higher frequency modes such as acoustic waves. Horizontally propagating acoustic modes are integrated using an explicit forward-backward scheme, and vertically propagating acoustic modes and buoyancy oscillations are treated implicitly.

Several options for lateral boundary conditions are available for use in WRF. These are detailed in Skamarock et al. [2005] and include periodic, open or radiative, symmetric, and specified. The first three boundary conditions are often used in idealized cases, whereas the specified boundary conditions are common in cases with real external data.

In the vertical direction, the top boundary condition is specified to be isobaric, and the Cartesian vertical velocity w is set to zero. Additionally, gravity waves can be absorbed with a diffusive or Rayleigh damping layer. At the bottom boundary the contravariant coordinate velocity is set to zero, and a kinematic boundary condition is used for the Cartesian vertical velocity. The set of equations given by (5a) and (5b) create a free slip bottom boundary

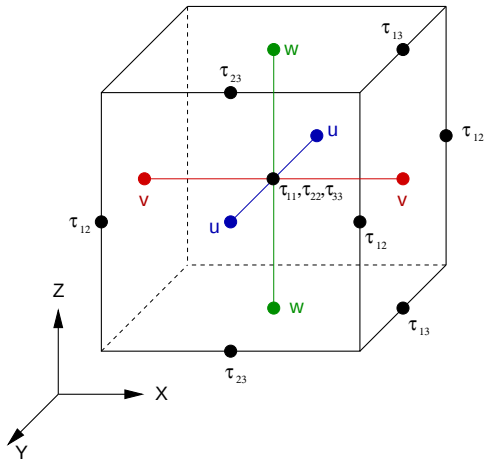


Figure 1: The Arakawa C staggered grid used in WRF.

condition.

$$\dot{\eta}_{surf} = 0 \qquad (5a)$$

$$w_{surf} = \vec{V} \cdot \nabla h \qquad (5b)$$

In equation (5b) h is a function specifying the terrain height. The horizontal velocities are extrapolated to the surface using a quadratic Lagrange polynomial. The shear stress at the boundary is implicitly set to zero, unless the effects of friction are taken into account by specifying a user determined coefficient of drag.

2.2 DEVELOPMENT OF THE NO-SLIP BOUNDARY CONDITION

As part of this work, a no-slip bottom boundary condition has been added to WRF as an additional option beyond the standard boundary conditions available in the code. To add the no-slip option, several modifications were made to the original WRF boundary conditions, which are given in the previous section as (5). The first equation (5a) is still appropriate, as it sets the contravariant velocity of the bottom coordinate to zero. This ensures that the bottom coordinate follows the terrain, and does not change position in time. The kinematic boundary condition given by (5b) is no longer valid, and is replaced with the requirement that the Cartesian vertical velocity $w = 0$ on the boundary. The need to extrapolate the horizontal velocities to the surface is eliminated, therefore no approximations are made in our formulation of the no-slip boundary condition. With these changes, the no-slip boundary condition is satisfied for the advective fluxes in

the continuity and momentum equations. The use of an Arakawa C staggered grid (illustrated in Figure 1) eliminates the need to explicitly set the u and v velocities equal to zero on the surface. In the case of the vertical derivative in the horizontal momentum equations (the third term in equation (2b)) it can be seen that because $\dot{\eta}$ and $\vec{V}$ are both zero at the surface, a one-sided vertical difference is sufficient for accurately calculating this term at the bottom boundary of the domain.

If turbulent mixing is included in the model, then additional boundary conditions must be imposed on the diffusive flux terms in the momentum equation in order to achieve a no-slip condition. The WRF model includes options for calculating the diffusive terms within the framework of the terrain-following coordinate or in physical space. The physical space option is compatible with all of the turbulence parameterizations, whereas limited options for turbulence modeling are available within the coordinate metric framework. For this reason, the no-slip boundary condition has been implemented in the physical space formulation.

WRF employs eddy viscosity type turbulence models so that the turbulent mixing terms take the form given by (6), and the stress tensor is (7).

$$F_i = -\frac{\partial \tau_{ij}}{\partial x_j} + \frac{\partial}{\partial z}(\nabla z \cdot \tau_{ij}) \tag{6}$$

$$\tau_{ij} = -\mu \nu_t D_{ij} \tag{7}$$

In the diffusive flux equation the second term arises as the Jacobian of the coordinate transformation, and z is the physical height of the coordinate so that $\nabla z = \nabla \phi / g$. Again, the Del operates in the horizontal dimensions only. The variable μ is the column mass per unit area, ν_t is the turbulent eddy viscosity, and D_{ij} is twice the strain rate or deformation tensor and is defined as

$$D_{ij} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - (\nabla z)_j \frac{\partial u_i}{\partial z} - (\nabla z)_i \frac{\partial u_j}{\partial z} \tag{8}$$

The locations of the components of the deformation and stress tensors are shown in Figure 1. Diagonal elements of each tensor are located at the cell center, and off-diagonal elements are located at vorticity points which are centered on the cell edges.

To include the effects of the no-slip bottom boundary, the calculation of the deformation tensor is modified. The presence of the Jacobian terms complicates the implementation of the no-slip boundary condition by adding partial derivatives in the vertical direction to each term. The native WRF boundary condition (which is free slip) uses a quadratic Lagrange polynomial to estimate the u and v velocity at the surface for the purpose of calculating D_{11}, D_{22}, and D_{12}. This step is eliminated for the no-slip boundary condition, and a value of zero is used for u and v on the surface. In the native WRF boundary condition D_{13} and D_{23} are zero because they are located on the surface. For the no-slip boundary condition, the surface values are calculated using a one sided difference. These new values of the deformation tensor are used for calculating the turbulent stresses. Finally, the equation for the diffusion terms includes additional terms arising from the Jacobian of the coordinate transformation. With a free slip boundary condition τ_{ij} takes a value of zero on the surface. To satisfy the no-slip boundary condition, certain elements of τ_{ij} must now be calculated at various locations on the surface, such as below the cell center, below u points, etc. The calculation of these additional surface stress terms is consistent with the procedure previously described for evaluating deformation at the surface, and the methods employed are one-sided differences, setting velocities to zero at the surface, and averaging surface values to intermediate locations.

2.3 FORCING AND RECONSTRUCTION AT THE IMMERSED BOUNDARY

Immersed boundary method functionality has been added to the Weather Research and Forecasting model through modification to the source code and the addition of a FORTRAN module. IBM is a technique used to represent the effects of solid boundaries on a non-conforming structured grid. The effects of the external forcing of the fluid by the boundaries are represented by the addition of a body force term F_B in the equation for the conservation of momentum. The forcing term takes a non-zero value at computational nodes in the vicinity of the immersed boundary, but has no effect away from the boundaries.

$$\partial_t \vec{V} + \vec{V} \cdot \nabla \vec{V} = -\alpha \nabla p + \nu \nabla^2 \vec{V} + \vec{g} + \vec{F}_B \tag{9}$$

Treatment of the forcing term has varied among researchers since IBM was introduced by Peskin in 1972. The method used in this work falls into a category commonly referred to as discrete or direct forcing which first appeared in Mohd-Yusof [1997], and was subsequently used by Fadlun et al. [2000], Iaccarino and Verzicco [2003], and others. Direct forcing is especially adept at handling rigid boundaries,

and produces a sharp representation of the fluid-solid interface. This property is desirable for viscous high Reynolds number flows where accurately resolving boundary layers is extremely challenging. In direct forcing, the forcing term takes the value given by (10), however, it is not explicitly calculated. Instead, the desired Dirichlet boundary condition $\vec{v}$ is directly imposed on the boundary. $\vec{V}^n$ is the velocity of the node where forcing will be imposed.

$$\vec{F}_B = \frac{\vec{v} - \vec{V}^n}{\Delta t} + \vec{V} \cdot \nabla \vec{V} + \alpha \nabla p - \nu \nabla^2 \vec{V} - \vec{g} \quad (10)$$

In the case that the boundary is coincident with computational nodes, it is clear that the Dirichlet boundary condition can be imposed by assigning the value of $\vec{v}$ to the node. However when the boundary passes through the grid in an arbitrary manner, the discrete grid points are not generally aligned with the boundary. In particular this is the case on staggered grids, like the one used in WRF. An interpolation method must be used to determine the forcing needed at actual computational nodes; this procedure is often called boundary reconstruction.

The first step in boundary reconstruction is to specify the terrain independently of the grid. For the implementation into the WRF model we have allowed specification of terrain height at twice the resolution of the grid, as illustrated in Figure 2. Specifying the terrain elevation in the $x - y$ plane is consistent with raw urban lidar data. Lidar data is often processed in an effort to extract three-dimensional features like vertical walls and 90-degree angles from the raw two-dimensional elevation data, creating a shape file. Future efforts are planned to allow the immersed boundary to be represented by either type of data.

The next step is the determination of cells that are cut by the immersed boundary. With a staggered grid, cut cells must be determined for each flow variable that will have a boundary condition imposed. Each node for each variable (u, v, w) is marked as interior (solid nodes) or exterior (fluid nodes) to the terrain to define the cut cells. Flow variables such as velocity can be reconstructed at fluid nodes as in Fadlun et al. [2000], or at solid nodes as in Mohd-Yusof [1997]. In this work the velocity is reconstructed at solid nodes. This reconstruction technique is called a ghost cell method, and has been used for incompressible flows by Majumdar et al. [2001], Tseng and Ferziger [2003], and others. Ghost points (depicted in Figure 3) are identified as the layer of nodes belonging to cut cells that are within the interior or solid region of the domain.

Now the value of the variable at the ghost cell which will enforce the Dirichlet boundary condition must be computed. Several different interpolation methods have been employed by researchers for the purpose of making this calculation, ranging from linear interpolation to inverse distance weighting schemes [Iaccarino and Verzicco, 2003]. For the compressible flow equations used in WRF, we have developed a unique bilinear reconstruction scheme for two-dimensional terrain. This method is similar to the linear interpolation scheme used by Tseng and Ferziger [2003], but has additional favorable properties. Tseng and Ferziger used the location of the boundary point and the two nearest computational nodes to the ghost point for determining the weighting coefficients used in calculating the value at the ghost point. Large extrapolation coefficients result when the nearest neighbors are close to the boundary point and the ghost point is relatively far away. Our method eliminates the large weighting coefficients, which cause the IBM to assign unphysical velocity values to the ghost points, often leading to numerical instabilities.

The bilinear interpolation method used in this work is illustrated in Figure 3, and the equation for a generic variable ϕ is given by (11).

$$\phi = c_1 + c_2 x + c_3 z + c_4 x z \quad (11)$$

First, the location of the ghost point is reflected across the boundary in the surface normal direction, and this is labeled an image point. The (x, z) location of the image point is used in equation (11). When a no-slip boundary condition is used, the ghost point and image point will be equal in magnitude with opposite signs $\phi_G = -\phi_I$. The weighting

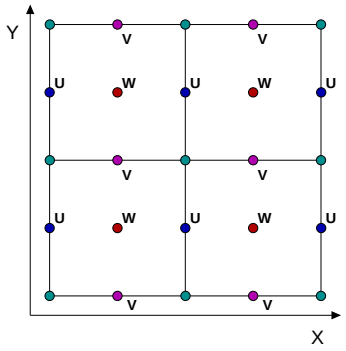


Figure 2: Terrain heights are specified at each the marked locations marked by a dot. The terrain data may have up to twice the resolution of the computational grid.

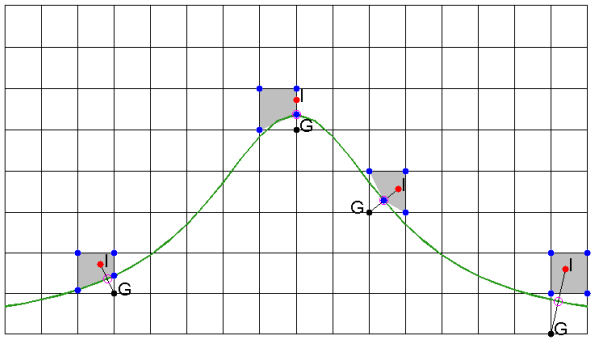


Figure 3: Illustration of the boundary reconstruction method developed for use in WRF. Black ghost points and red image points are labeled with a 'G' and 'I' respectively. Open magenta circles represent boundary points, and closed blue circles are the nearest neighbors of the image point. The interpolation region bounded by the neighbors is shaded in grey.

coefficients $\vec{c}$ are determined by solving the system of equations (12) for each ghost point.

$$\vec{c} = \mathbf{V}^{-1}\vec{\phi} \tag{12}$$

The Vandermonde matrix (13), contains the physical location of four neighboring points.

$$\mathbf{V} = \begin{bmatrix} 1 & x_1 & z_1 & x_1 z_1 \\ 1 & x_2 & z_2 & x_2 z_2 \\ 1 & x_3 & z_3 & x_3 z_3 \\ 1 & x_4 & z_4 & x_4 z_4 \end{bmatrix} \tag{13}$$

The nearest neighbors for (13) are now chosen as those nodes or boundary points that are nearest to the image point (unlike previous methods which choose neighbors closest to the ghost point). This is advantageous because the image point lies within the interpolation region bounded by the neighbors, and a realistic and physical velocity value will result from the interpolation. The velocity values at the neighbors make up the vector $\vec{\phi}$.

The neighbors can be chosen as computational nodes or as boundary points. Examples of the choices made are shown in Figure 3, but this illustration is not exhaustive. It can be seen that if all four nodes surrounding the image point belong to the fluid domain, then these are chosen as the neighboring points. The variable values calculated by the solver enter the vector $\vec{\phi}$. If one of the computational nodes lies within the solid domain, then the boundary point in the surface normal direction is used instead. The value of this point is known from the Dirichlet boundary condition. If two neighboring computational nodes are solid points, then two boundary points are chosen instead. Solid nodes are never used in the interpolation scheme. As a last step, the variable value calculated from this scheme is assigned to the ghost node.

It should be noted that the hydrostatic pressure coordinate in WRF is not time invariant. Therefore, this procedure, starting with the determination of fluid and solid nodes and ending with imposing the boundary condition by assigning values at the ghost point, must be repeated for each iteration of the solver. The computational penalty for repeating these steps has not been prohibitive, however, the authors are investigating ways to improve efficiency. The simulations presented in Section 3 are a direct comparison between WRF with the no-slip boundary conditions, and WRF with the immersed boundary method. On average, the simulations using IBM required 30% more computational time when compared to the non-IBM solution. The advantage of IBM, as demonstrated in Section 4, is that it can handle highly complex terrain. A quantitative comparison of resources could not be made for the cases where WRF is unable to handle the terrain with its original terrain-following grid. Additionally, although the WRF grid is structured, it is not Cartesian. We found that special care must be exercised in determining the interpolation neighbors by accounting for the horizontal gradients in the vertical coordinate.

3. WITCH OF AGNESI HILL SIMULATIONS

The performance of the IBM method is examined in this section by considering flow over a two-dimensional hill defined by the Witch of Agnesi curve. This case is chosen because a direct comparison can be made between WRF with the no-slip modifications detailed in Section 2.2 and using the IBM method described in Section 2.3. Previous simulations have demonstrated that our IBM implementation is able to reproduce results compared with the original WRF coordinate system for turbulent flow over flat terrain in three dimensions using a variety of surface boundary conditions [Lundquist, 2006]. This previous work includes simulations with no-slip boundary conditions along with two IBM implementations that use approximate boundary conditions based on the log-law. These are referred to as log-law velocity reconstruction [Senocak et al., 2004] and shear stress reconstruction [Lundquist, 2006]. Future work will extend the results with com-

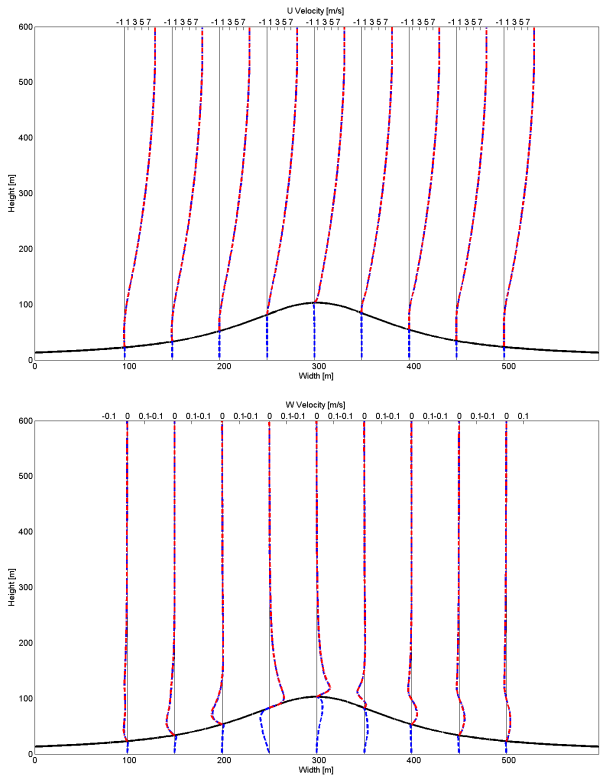


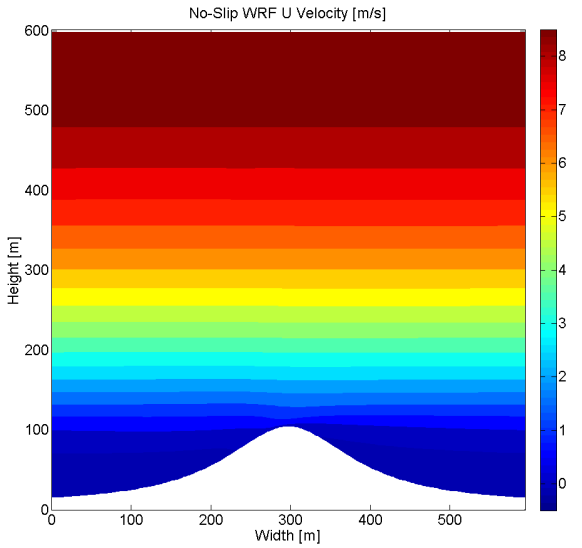
Figure 4: Profiles of u and w velocity [m/s] are shown for start up flow over a hill at 72 hours. The blue dashed line is the IBM solution, and the red dash-dot line is the no-slip WRF solution.

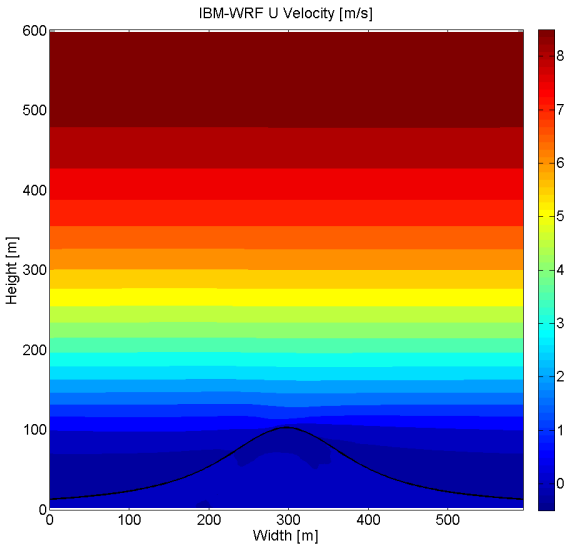
plex terrain presented here to three dimensions, and will include the ability to use a wall model parameterization.

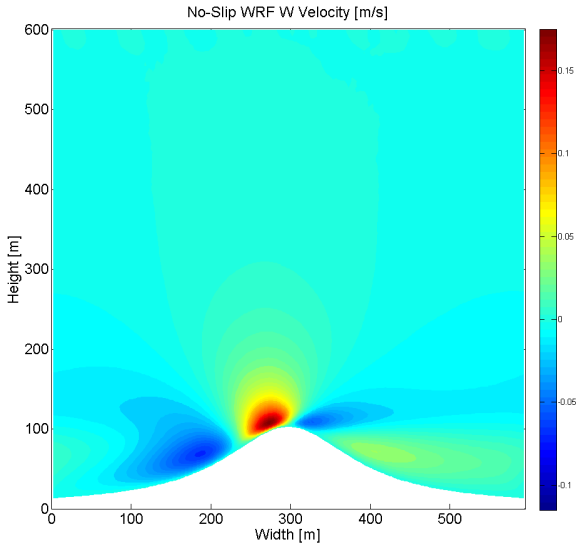
The test flow case is startup flow over a two-dimensional hill. The hill is defined by the Witch of Agnesi curve using a height and mountain half-width of 100m each. The flow is initialized with a neutral and quiescent sounding, and driven with a constant pressure gradient in the x direction. The number of grid points used in each direction is $(n_x, n_y, n_z) = (120, 3, 172)$ for a total domain size of $(X, Y, Z) = (595m, 10m, 600m)$. In the horizontal dimensions a constant 5m grid spacing is used. In the vertical direction the grid is not stretched in η, however, the pressure coordinate naturally results in stretching in physical space. In the case with terrain-following coordinates, the vertical grid spacing is $\Delta z_{min} \approx 2.85m$ and $\Delta z_{max} \approx 3.52m$ at the initialization time. In the IBM domain the vertical grid spans the entire 600m domain height, giving a more uniform (and slightly coarser) grid resolution, with $\Delta z_{min} \approx 3.42m$ and $\Delta z_{max} \approx 3.6m$. A constant eddy viscosity is used in the turbulent diffusion terms. For more sophisticated turbulence models (such as those solving a turbulent kinetic energy equation), boundary conditions are needed for the turbulent scalars. Using a constant eddy viscosity allows us to verify our IBM implementation independently of the turbulence model. Future plans include the extension of the IBM method to include the treatment of turbulent scalars.

Periodic boundary conditions are used at the lateral boundaries. The no-slip boundary condition at the bottom of the domain is achieved either through the no-slip modifications in the terrain-following WRF, or with IBM; both methods were previously described in Section 2. At the top of the domain, the native WRF boundary condition is used (isobaric and zero vertical velocity). A diffusive damping layer with a thickness of 50m is used at the top of the domain to prevent wave reflection off of the top boundary.

Results are shown for the hill case after 72 hours of simulation. Figure 4 shows u and w velocity profiles at several locations across the domain. It can be seen that the IBM solution matches the no-slip WRF solution almost exactly. At this instance in time, the maximum u velocity at the top of the domain is 8.9296m/s in the IBM solution and 8.9294m/s in the no-slip WRF solution. Given the different vertical grid spacing in the two simulations, the excellent agreement is remarkable. Contours of the same data are shown in Figures 5 and 6. The u velocity increases as it crests the hill, and areas of weak recirculation are observed in the troughs. The profiles of u and w show that separation occurs on the front of the hill between 200 and 250m (due to the periodic boundary conditions), and again on the lee of the hill between 300 and 350m. The figures also illustrate the behavior of the flow in the interior, or solid, portion of the IBM domain. A weak recirculation is generated in the opposite sense to the flow above the hill. While the flow beneath the hill is of no practical interest in this case, it is interesting to note the features that develop and useful to verify that the boundary conditions on hill surface are indeed being set correctly.







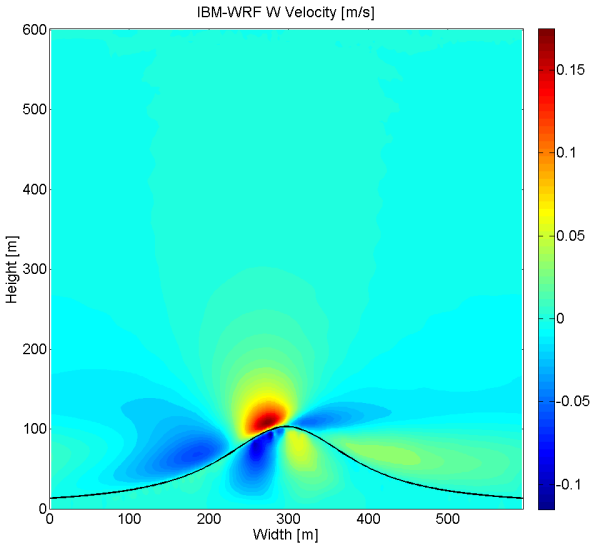


Figure 5: Contours of u and w velocity [m/s] are shown for no-slip WRF for start up flow over a hill at 72 hours.

Figure 6: Contours of u and w velocity [m/s] are shown for IBM-WRF for start up flow over a hill at 72 hours.

4. TWO-DIMENSIONAL URBAN TERRAIN SIMULATIONS

The IBM allows resolution of urban terrain, enabling WRF to simulate flows which cannot be computed using a standard terrain-following coordinate. To demonstrate this capability, we have modeled flow over a two-dimensional slice of downtown New York City. The terrain is featured in Figure 7, and the tallest building that is included has a height of 248m. The domain size is $(X, Y, Z) = (698m, 4m, 600m)$ with $\Delta x = 2m$, $\Delta y = 2m$, and $\Delta z \approx 2.4m$. The total number of grid points in each direction is $(n_x, n_y, n_z) = (350, 3, 250)$. At initialization the fluid is at rest, and the flow is driven by a constant horizontal pressure gradient. Boundary conditions are identical to those described in Section 3.

Figure 7 shows flow streamlines and the velocity magnitude at three different snapshots in time. After 20 minutes of flow spin up, recirculation regions form behind several of the buildings. By 6 hours there is a large area of recirculation between the buildings, and the flow velocities are fairly weak in these regions. Although not shown in the figure, the peak velocity in the domain at 6 hours is 3.9m/s at a height of 600m.

5. CONCLUSIONS

This work has demonstrated that the immersed boundary method is a viable option in NWP, and specifically WRF, for removing the barriers to modeling complex geometries created by the use of terrain-following coordinates. A new IBM suitable for viscous compressible flows has been developed and implemented in WRF. The IBM features a boundary reconstruction method that utilizes the concept of image points. By choosing the nearest neighbors to the image points, we have alleviated the problems associated with large weighting coefficients which have been noted in previous IBM implementations. A no-slip bottom boundary condition has also been added to WRF as an additional option with terrain-following coordinates. For flow over a two-dimensional hill, the velocity profiles using IBM show excellent agreement with the solution calculated on a terrain-following grid. Finally, applicability to realistic, highly-complex urban flows was demonstrated by modeling flow over buildings in New York City. Future work will extend these results to three dimensions, ultimately allowing for seamless grid nesting from the meso-scale to the urban scale.

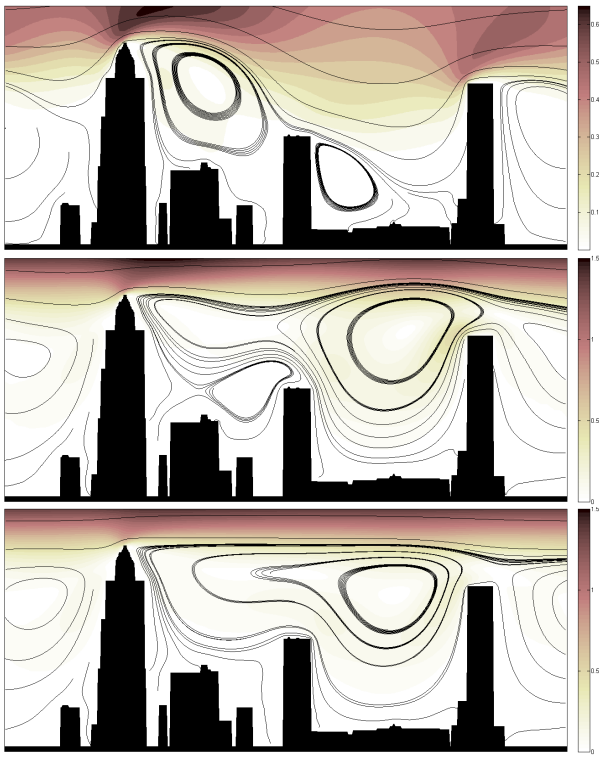


Figure 7: Velocity contours (m/s) and streamlines of startup flow over two-dimensional building data from NYC using IBM-WRF. (a) Time = 20 min (b) Time = 3hrs 20min (c) Time = 6hrs

6. ACKNOWLEDGEMENTS

The first author is grateful for the support of a Student Employee Graduate Research Fellowship from Lawrence Livermore National Laboratory (LLNL). This project was funded by the LLNL Laboratory-Driven Research and Development program. This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

REFERENCES

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

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

R. Laprise. The euler equations of motion with hydrostatic pressure as an independent variable. *Mon. Weather Rev.*, 120(7):197–207, July 1992.

J.K. Lundquist, F.K. Chow, J.D. Mirocha, and K.A. Lundquist. An improved WRF for urban-scale and complex-terrain ap-

plications. In *7th Symposium on the Urban Environment*. American Meteorological Society, 2007.

K.A. Lundquist. Implementation of the immersed boundary method in the weather research and forecasting model. Master's thesis, University of California, Berkeley, 2006.

S. Majumdar, G. Iaccarino, and P. Durbin. RANS solvers with adaptive structured boundary non-conforming grids. Technical report, Center for Turbulence Research, NASA Ames/Stanford Univ., Palo Alto, CA, 2001.

R. Mittal and G. Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261, 2005.

J. Mohd-Yusof. Combined immersed boundary/b-spline methods for simulations of flow in complex geometry. Technical report, Center for Turbulence Research, NASA Ames/Stanford Univ., Palo Alto, CA, 1997.

C.S. Peskin. Flow patterns around heart valves: A numerical method. *J. Comp. Phys.*, 10(2):252–271, October 1972.

I. Senocak, A.S. Ackerman, D.E Stevens, and N.N Mansour. Topography modeling in atmospheric flows using the immersed boundary method. Technical report, Center for Turbulence Research, NASA Ames/Stanford Univ., Palo Alto, CA, 2004.

W.C. Skamarock, J.B. Klemp, J. Dudia, D.O. Gill, D.M. Barker, W. Wang, and J.G. Powers. A description of the advanced research wrf version 2. Technical Report NCAR/TN-468+STR, National Center for Atmospheric Research, Boulder, CO, June 2005.

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