JP2.11

CFD Modeling for Urban Area Contaminant Transport and Dispersion: Model Description and Data Requirements

William J. Coirier *, Sura Kim
CFD Research Corporation, Alabama

1. INTRODUCTION

Computational Fluid Dynamics (CFD) modeling for urban area wind, turbulence and contaminant transport and dispersion is becoming more accepted by the community as a useful means to understand the complex flow and the resulting dispersion behavior in urban areas. As shown in other papers presented at this conference [Camelli et al., 2006, Coirier et al., 2006.a,b] and papers and presentations made at other conferences [Coirier et al., 2003.a, 2005.b,c], urban areas, such as New York City and many others, often exhibit localized, intense vertical mixing and lateral spreading behavior that is dependent upon building shapes, sizes, locations, and prevailing wind conditions. Small changes in contaminant source locations can produce quite different contaminant spreading behavior. This paper describes a model that has been developed to simulate the wind, turbulence and dispersion in urban areas on the building to city blocks scale, and summarizes the data typically needed to perform the simulations. In particular, the finite-volume, Reynolds-Averaged Navier-Stokes formulation of the model is described, as well as different techniques that are used to model buildings both explicitly (by resolving the building surfaces) and implicitly (by introducing momentum and turbulence source terms to represent building effects upon the flow fields). Turbulence closure using a standard k-ε model is described, including modifications to the turbulence model and boundary conditions to better represent atmospheric boundary layers. Two different cityscape CFD model generation techniques that ingest GIS data are described, which respectively generate prismatic-hexahedral and quadtree-prismatic/octree grids. Data requirements for this model are defined, including; flow and turbulence boundary conditions, building geometric descriptions and contaminant source description.

corresponding author address:
CFD Research Corporation,
215 Wynn Dr., 5th Floor, Huntsville, AL 35805
e-mail: wjc@cfdrc.com.

2. CFD-URBAN MODEL COMPONENTS DESCRIPTIONS

CFD-Urban is a suite of Computational Fluid Dynamics modeling software that is being used to simulate the wind, turbulence and dispersion fields in urban areas [Coirier et al., 2003.a, 2005.a]. CFD-Urban has been developed under a program sponsored by the Defense Threat Reduction Agency [Coirier et al., 2003.b], and has been built using parts of a commercially available software suite, CFD-ACE+ [ACE+ 2003]. It solves the Reynolds-Averaged Navier-Stokes equations using a collocated, Finite-Volume method implemented upon structured, unstructured and adaptively-refined grids using a pressure-based approach based upon the SIMPLE algorithm [Jiang, 1994, Jiang, 1999]. Turbulence closure is found by solving a variant of the standard k-ε model [Launder, 1974]. Buildings are modeled either explicitly, by resolving the buildings themselves, and/or implicitly, by modeling the effects of the buildings upon the flow by the introduction of source terms in the momentum and turbulence model equations [Coirier 2003.b]. CFD-Urban solves the steady-state and unsteady Reynolds Averaged Navier-Stokes (RANS) equations, as well as by using a Large Eddy Simulation (LES) approach. Since CFD-Urban solves the governing mass and momentum conservation laws at scales smaller than the buildings themselves, important urban aerodynamic features are naturally accounted for, including effects such as channeling, enhanced vertical mixing, downwash and street level flow energization.

The main components of CFD-Urban software suite consists of cityscape CFD model generation, wind and turbulence field solution, contaminant transport and dispersion field generation, and a collection of post-processing

tools. The sections below describe these components.

2.1 Cityscape CFD Model Generation

Model generation is typically performed using Geographic Information System (GIS) data in the ESRI two-dimensional, shp file format [ESRI]. This format stores building footprints as polygons, with height (and other) attribute data associated with the polygons in a separate file. We use the public domain library, shapelib [shapelib.maptools.org], to read and write the ESRI shp file data, and have imbedded this C-language library into various C++ applications and classes related to the model generation. For imagery and Lidar derived image-format data, we use a customized graphical user interface to interrogate the (image-encoded) height data to attribute manually drawn polygons. This data is then saved in the ESRI shp file format. Once we have either created or obtained the shp file data, model generation is made by using one of the two following techniques.

2.1.1 Prismatic-Hexahedral Mesh Generation

A specialized cityscape CFD volume mesh generator has been developed that is derived from a commercial model generator, CFD-Micromesh-C. The rapid generation of high quality, parallel processing suitable CFD volume meshes of cityscapes directly from GIS data is accomplished with customized scripts, GIS data import and manipulation and specialized model generation capability explained fully in [Coirier, 2003.b]. Briefly described here, the urban area model generator creates volume meshes by the following sequence of steps:

Step 1: Solid Model Generation: The GIS data is used to construct a voxel-based solid model. Feature suppression is controlled by changing the voxel size: Features smaller than the voxel size are not represented and hence, are not meshed.

Step 2: Vectorization of Solid Model: Once the solid model has been formed, the voxel boundaries across which material properties or voxel properties change are projected onto a z=constant plane, and these boundaries are vectorized, yielding sets of polygons that describe the projected geometry.

Step 3: Tessellation of Vectorized Model: The vectorized model is then tessellated into collections of triangular and quadrilateral faces. A high quality tessellation is made using the Delaunay criteria [Shewchuk, 1996.a], combined with a circumcircle test [Ruppert, 1995] that uses adaptive precision arithmetic [Schewchuk, 1996.b]. Mesh clustering is achieved in the lateral (x-y) plane via a sequence of mesh refinement regions that increase the mesh size geometrically based on distance from the resolved region.

Step 4: Extrusion of Tessellation: The tessellated z=constant plane is then extruded in z, yielding prisms (from triangles) and hexahedra (from quadrilaterals). The Roberts transformation [Anderson, 1984] is used to cluster the mesh in the z-direction, normal to the ground plane. The clustering is found by solving a non-linear equation that relates the height of the first cell-centroid in physical space to a computational coordinate, η_0:

$$G = \frac{H}{2} F(\eta_0, \beta) - \Delta z_0 \qquad (1)$$

where, F is the Roberts transformation:

$$F(\eta, \beta) = \frac{(\beta+1) - (\beta-1)\left\{[(\beta+1)/(\beta-1)]^{(1-\eta)}\right\}}{\left\{[(\beta+1)/(\beta-1)]^{(1-\eta)}\right\}+1} \qquad (2)$$

H is the height of the upper boundary of the mesh above the ground plane, and Δz_0 is the height of the first cell center above the ground plane, and η_0 is the normalized coordinate of the first grid point, which defined by the number of z-layers. The equation G=0 is iteratively solved for β via Newton's iteration using functional derivatives of G. Once β is found, the continuous z-layer distribution is found from the Roberts transformation:

$$z_i = H * F(\eta_i, \beta) \qquad (3)$$

For parallel processing, domain decomposition is performed during the mesh generation phase by partitioning the surface triangulation (Step 3) before extrusion. This allows large meshes to be created with a low memory footprint, and requires minimal pre-processing for parallel computations by the RANS solver. The METIS

domain partitioning library [Karypis, 1998] is used to create the partitioning after being given the graph connectivity of the surface tessellation. To insure load balancing after extrusion and deletion of the in-building volume mesh, the graph partitioning strategy uses a weighted adjacency graph that is found by using the air cell-stack size for a given surface triangle. An example of a primatic-hexahedral mesh CFD model is shown in Figure 1.

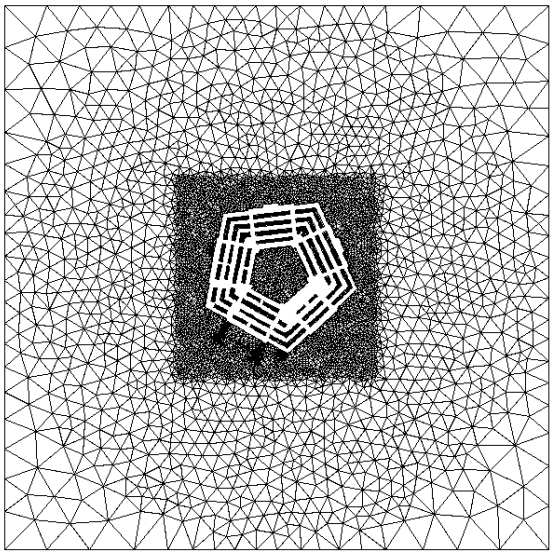

Figure 1: Prismatic-Hexahedral mesh about the Pentagon.

2.1.2 Quadtree-Prismatic/Octree Mesh Generation

As an alternative to the prismatic-hexahedral mesh generation, we are advocating the use of Quadtree-Prismatic and Octree based grids. Within the CFD-Urban suite, these "tree-based" mesh generation techniques use C++ classes that have been developed for a Hierarchical, Adaptive Mesh Refinement (HAMR) gas dynamics solver [Coirier et al., 2001, 2002.b], that was originally made to model the air blast propagation in urban areas. Hierarchically-based classes/data structures for CFD flow solver and model generation are based upon recursively subdividing a single, root, Cartesian cell into smaller cells by isotropic subdivision, and storing the cell refinement structure in a hierarchical (tree) data structure or class that grows with the refinement [Coirier, 1994, Coirier et al., 1995]. Figure 2 illustrates the octree refinement of a single Cartesian cell into eight cells.

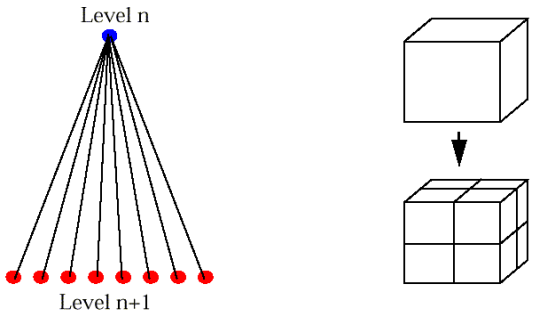

Figure 2: Octree Data Structure and corresponding Cartesian cell refinement.

Hierarchically-based data structures offer some distinct advantages over traditional approaches used for unstructured solvers, such as face-based and edge-based schemes. Adaptively refined Cartesian grids exhibit outstanding mesh orthogonality with an accompanying reduction in truncation error, while allowing local mesh refinement in regions dictated by either the flow and/or the geometry of the problem. When applied to urban areas, as with CFD-Urban, high mesh resolution in an area of interest, such as the Central Business District of a city, can be maintained, yet the domain of interest cover 10's of square kilometers, while keeping the cell count of the mesh to a reasonable size.

The mesh generation process begins by the creation of a single, root, Cartesian cell spanning the entire domain. This is recursively subdivided in the x- and y- directions using a quadtree refinement, where each parent cell stores the 4 newly refined children cells. The recursive subdivision terminates when all cells within a defined refinement region are smaller than a specified threshold, and do not grow larger than a maximum size specified on a larger refinement region. During the refinement phase, the tree is "smoothed", which only allows only a 2:1 cell size change across cell and vertex boundaries. Once this single quadtree mesh is generated (which at this point has only a single cell layer in the z-direction), all the cells are refined in the z-direction (recursively) until the number of z-layers specified is reached. The z-coordinate is then "mapped" to the clustering (shown in equation 2), resulting in what may be represented (if one were to use a different, specialized data structure/class) as a quadtree-prismatic grid, but is truly generated in an octree fashion.

After the mesh is generated, it is processed by the drag processor (Section 2.2.4), which

computes the cell porosity and projected area/volume ratios in all the cells of the mesh using the GIS data. Those cells with a porosity less than a threshold (typically taken to be $\beta \le 0.001$) are removed from the mesh. Domain decomposition for parallel processing is made using the METIS library [Karypis], and the individual cells in each domain are subsequently reordered using a recursive, wavefront-like ordering scheme, to reduce the sparse matrix bandwidth in the flow solver. An example mesh using this technique is shown below, which covers the Oklahoma City Central Business District (CBD).

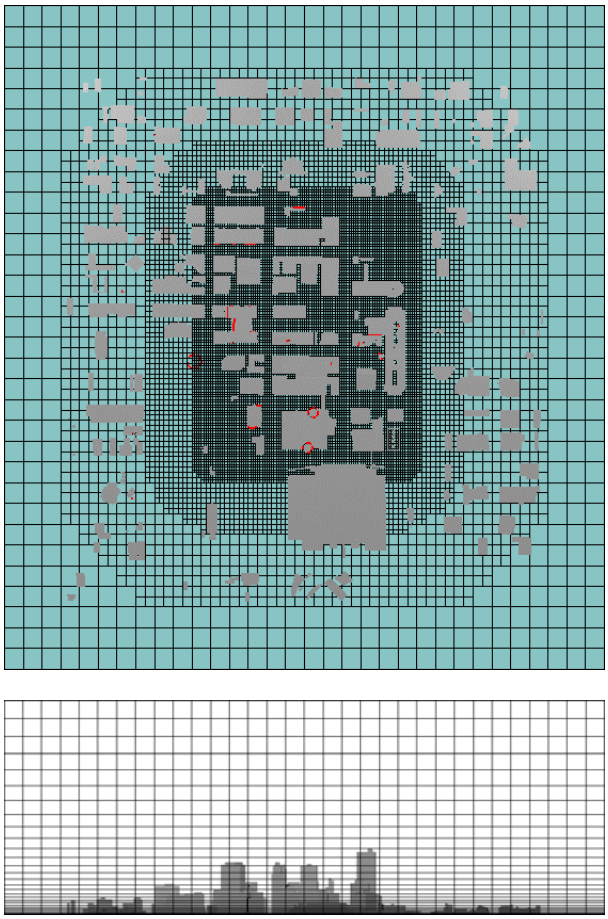

Figure 3: Quadtree-Prismatic/Octree mesh over the Oklahoma City CBD.

2.1.3 Terrain Mapping

Digital elevation data may be used to "map" the z=constant mesh to be conformal to the underlying terrain. The data may be in a number of standard digital elevation data formats including DTED and GeoTIF. The terrain mapping of the CFD mesh may be performed in two manners; a displacement or a compression approach. The displacement approach simply displaces each coordinate in the mesh in the z-direction, according to the height given by the digital elevation data at that (x,y) location. The compression approach keeps the upper boundary of the domain at a constant height, and thus compresses the coordinates between the ground surface and upper boundary. Since the compression approach can reduce mesh skew angle quality, the displacement mode is preferred.

2.2 Wind and Turbulence Field Generation

CFD-Urban solves the Reynolds-Averaged Navier-Stokes equations using a collocated, pressure-based, finite-volume method that has been generalized to arbitrarily constructed control volumes of closed polyhedra. This generalized geometric formulation includes the standard unstructured Finite Volume elements of tetrahedra, prisms, pyramids and hexahedra, as well as elements found from projection-based methods and those that use cell-based adaptive mesh refinement. Mass conservation is insured by coupling the continuity and momentum equations using an extension of the SIMPLE algorithm, formulated for unstructured and polyhedral grids [Jiang et al., 1999; Athavale et al., 1995]. A collocated approach is taken that stores all solution unknowns at the cell-centers of the mesh, where cell-centered gradients are reconstructed using a flux-formulation that performs a surface integral for each cell using values constructed at the vertices/nodes of the mesh that are found using a linearity-preserving interpolation scheme [Jiang et al., 1999]. The generalized formulation used in this model provides the framework for solving other conservation laws and transport equations, which has resulted in the CFD-ACE+ solver and system [ACE+ 2003]. Parallel processing is typically performed on clusters of personal computers, where the CFD-Urban solver uses MPI to perform message passing.

2.2.1: Reynolds-Averaged Navier-Stokes Formulation

The RANS conservation equations are found by substituting into the original (un-averaged) mass and momentum equations a time mean and a fluctuating component, $f = \overline{f} + f'$. Upon time averaging ("Reynold's averaging") and evaluating for vanishing control volumes the RANS equations are found, shown below in Cartesian tensor notation.

$$\frac{\partial}{\partial x_i}\left(\rho \bar{u}_i\right) = 0 \tag{4}$$

$$\frac{\partial}{\partial x_j}\left(\rho \bar{u}_i \bar{u}_j\right) = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\bar{\tau}_{ij} - \overline{\rho u_i' u'}_j\right) + \bar{S}_i \tag{5}$$

2.2.2 Turbulence Closure via k-ε Model

The Reynold's averaging introduces additional turbulent, diffusive fluxes, typically termed the Reynolds Stresses. These are modeled by the Boussinesq assumption [Anderson, 1984], which states that the Reynold's stresses are linearly proportional to the rate of strain, where the proportionality coefficient is referred to as the eddy viscosity, μ_t.

$$-\overline{\rho u_i' u'_j} = \mu_t\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}\left(\mu_t \frac{\partial \bar{u}_k}{\partial x_k} + \rho k\right) \tag{6}$$

Turbulence closure is found by solving auxiliary field equations that yield the eddy viscosity, μ_t.

The standard k-ε model [Launder, 1974] solves evolution equations for the turbulence kinetic energy, k, and its dissipation rate, ε, shown below in the averaged, steady-state formulation:

$$\frac{\partial}{\partial x_j}(\rho \bar{u}_j k) = \rho P - \rho\varepsilon + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \tag{7}$$

$$\frac{\partial}{\partial x_j}(\rho \bar{u}_j \varepsilon) = C_{\varepsilon_1}\frac{\rho P \varepsilon}{k} - C_{\varepsilon_2}\frac{\rho\varepsilon^2}{k} + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] \tag{8}$$

The expression for the eddy viscosity relates the turbulence kinetic energy and dissipation rate:

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \tag{9}$$

The constants appearing in the standard model equations have been found by the evaluation of the model for a wide range of engineering flows [Launder, 1974]. For atmospheric flows, it has been found by Dettering [Dettering] that other coefficients may be more appropriate for this model. The coefficients used in CFD-Urban for the standard k-ε and the atmospheric boundary layer (ABL) models are shown below.

Standard	ABL
$C_\mu = 0.09$	$C_\mu = 0.0256$
$C_{\varepsilon 1} = 1.43$	$C_{\varepsilon 1} = 1.13$
$C_{\varepsilon 2} = 1.92$	$C_{\varepsilon 2} = 1.90$
$\sigma_k = 1.0$	$\sigma_k = 0.74074$
$\sigma_\varepsilon = 1.3$	$\sigma_\varepsilon = 1.298701$

Table 1: Standard [Launder] and ABL [Dettering] coefficients for the standard k-ε model.

2.2.3 Boundary Conditions

Inflow and outflow boundary conditions for the equations are applied using an approach based upon the evaluation of the sign of the mass-flux leaving the domain on each face. For faces where this is positive (outflow boundaries), first-order extrapolation is used to define the values of the velocity and turbulence field values at the face centers, while an externally supplied pressure difference from the (constant) reference pressure is applied. For faces where this is negative (inflow boundaries), a collection of functional and tabulated forms of the velocity and turbulence field may be used to specify the inflow properties, and the pressure is extrapolated from the interior. The functional velocity and turbulence profiles include logarithmic and Monin-Obukhov Similarity (MOS) profiles, using different forms for the turbulence kinetic energy and dissipation rates. Generic tabulated input is also available, which interpolates, in height above ground level, input profiles to the boundary face centers.

Wall boundary conditions are supplied using the standard wall function approach [Launder, 1974] using the meteorological form of roughness, as opposed to the equivalent sand roughness [Schlichting, 1979]. The wall function approach specifies the values of k and ε in the first cell away from the wall using the functions that relate the friction velocity, u_τ to k and ε, where z denotes distance to a wall.

$$k = \frac{u_\tau^2}{\sqrt{C_\mu}} \tag{10}$$

$$\varepsilon = \frac{C_\mu^{\frac{3}{4}} k^{\frac{3}{2}}}{\kappa z} \tag{11}$$

In the standard wall function model [Launder, 1974; Yakhot, 1992], the wall shear stress is obtained by assuming the velocity

profile is fully turbulent and that the wall is hydraulically "rough". The turbulent momentum fluxes are then consistently defined according to equilibrium theory, using the specified wall roughness and local velocity gradients along with the turbulence model exchange coefficients.

2.2.4 Drag Processing

Buildings that are not explicitly resolved (meshed) by the model generator are treated using a distributed drag model, which adds source terms to the x- and y-momentum equations, as well as the turbulence kinetic energy and dissipation rate equations, according to the geometrical relationships between the volumes occupied by the building shapes and the CFD volume mesh. A polygon clipping operation based upon the Sutherland-Hodgman clipping [Sutherland et al., 1974] is used within a specialized post processing procedure that computes the projected (Cartesian axes) frontal areas and porosities of all building shapes contained within a cell. Fast, hashed container class algorithms are used to speed up this processing, resulting in the projected sub-cell resolved projected areas and porosities that are stored as separate data within the mesh data file. The core drag preprocessing algorithm performs polygon clipping operations to determine the (possibly multiple) polygon intersections with a given cell, by clipping the (non-convex) polygons representing the buildings against the lowest z-constant face(s) of the cell. The clipped areas are used with the attribute data to determine clipped volumes and hence porosity, which is included in the drag terms. The geometric terms found by this approach are:

$$A_x = \frac{\sum_p L_y^p \Delta z\, p}{1 - \sum_p \frac{V_{clip}^p}{V_{cell}}} = \frac{A_{frontal,x}}{porosity} \qquad A_y = \frac{\sum_p L_x^p \Delta z\, p}{1 - \sum_p \frac{V_{clip}^p}{V_{cell}}} = \frac{A_{frontal,y}}{porosity}$$

The polygons created in working with the building and drag data are used for visualization purposes as well. The non-convex polygons read into the drag modeling pre-processor are properly tessellated into triangles, and these triangles form the basis for extrusion into prisms, which are then output as separate entities for visualization of the drag modeled buildings.

Two different source term formulations are available for use within the RANS solver: A "Mesoscale" approach [Lacser] and source terms which contain turbulence kinetic energy source and sink terms[Liu] which were found in simulating the flow over forest canopies. Both approaches add source terms to the momentum and turbulence kinetic energy equation, while the Liu approach adds a source term to the kinetic energy dissipation rate equation. The source terms are shown in table 2, where $U = \sqrt{u^2 + v^2 + w^2}$.

The treatment of foliage upon the flow may also be simulated, but due to lack of reliable data and programmatic focus, this feature is seldom used.

2.3 Contaminant Dispersion Field Generation

The contaminant field is found by solving an unsteady, Eulerian, contaminant transport equation. The contaminant mass fraction is evolved in time using the supplied velocity and turbulence fields (either steady, quasi-steady or transient) according to:

$$\frac{\partial \rho Y_k}{\partial t} + \frac{\partial \left(\rho u_i Y_k\right)}{\partial x_i} = \frac{\partial(-\overline{\rho u'_j Y'_k})}{\partial x_j} + \rho S_k \qquad (12)$$

where the ρ is the mixture density, Y_k is the mass fraction of the k-th component, and the turbulent species flux is found using the RANS (Boussinesq) closure as:

$$-\overline{\rho u'_j Y'_k} = \frac{\mu_t}{\sigma_k}\frac{\partial Y_k}{\partial x_j} \qquad (13)$$

Mesoscale Model [Lacser]	Liu's Model [Liu]
$S_x = -A_x \frac{\rho}{2} u\|u\|$	$S_x = -A_x \frac{\rho}{2} u\|U\|$
$S_y = -A_y \frac{\rho}{2} v\|u\|$	$S_y = -A_y \frac{\rho}{2} v\|U\|$
$S_k = -\left(A_x^2 + A_y^2\right)^{y_2} \frac{\rho}{2}\left(u^2 + v^2\right)^{3/2}$	$S_k = \frac{\rho}{2}\left(A_x^2 + A_y^2\right)^{1/2}\left[\|U\|^3 - 4k\|U\|\right]$
None	$S_\varepsilon = \frac{\rho}{2}\left(A_x^2 + A_y^2\right)^{1/2}\left[\frac{3}{2}\frac{\varepsilon}{k}\|U\|^3 - 2.4\|U\|\varepsilon\right]$

Table 2: Drag Model Source Terms

Continuous releases are modeled by solving the steady form of equation 11, while unsteady, time varying releases are found by solving the transient form. Moving sources, and sources with time varying strengths, as well as multiple sources and multiple tracer gases are all available.

For contaminant dispersion calculations, we typically use what we term as a "frozen hydrodynamics" approach, where the steady-state, equilibrium flow field is held constant in equation 11, which makes the solution of the contaminant dispersion equation very efficient. Like the other transport equations in CFD-Urban, we use an implicit sparse matrix method that employs either a Conjugate Gradient solver using Incomplete LU preconditioning with zero fill in, or an Algebraic Multigrid Solver. Since the frozen hydrodynamics solver is not tied to any particular wind field, we can use wind field libraries that may be blended in time and parametric space. This disconnection of the wind and turbulence field from the contaminant transport field was first developed and used in the Phase I portion of the work [Coirier et al., 2003.b], and has subsequently provided the basis for an operational coupling strategy to couple CFD-Urban with Numerical Weather Prediction (NWP) models, as shown in a paper at this conference [Coirier et al., 2006.b]. The unified frozen hydrodynamic solver may blend wind and turbulent fields in time, and use these fields during the unsteady solution of the contaminant transport equations. An alternative use of this approach is to blend steady state flowfields in parametric space, such as prevailing condition angle, and use these fields in the contaminant transport equation.

2.3.1: Meandering Wind Effects using a Wind Library

As an example of the use of wind field libraries, we demonstrate a technique that may be used to simulate the effect of a meandering wind component upon the contaminant dispersion field, yet use only three steady state flow solutions. If we represent the meandering component as a single periodic function that varies the approaching flow angle with time (such as shown in Figure 4), one approach to the problem is to perform an unsteady RANS calculation, and solve the contaminant transport equations while we solve the RANS equations.

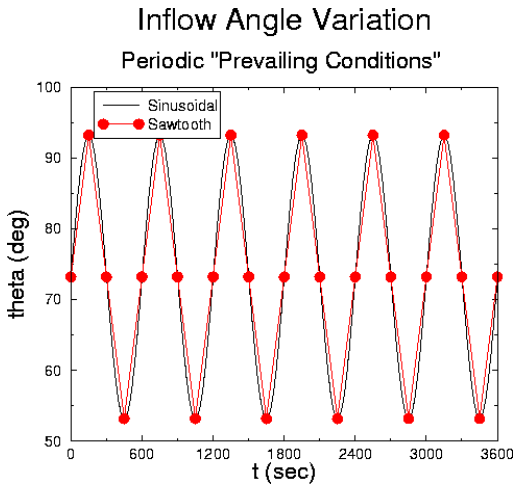

Figure 4: Periodic and Sawtooth Inflow Angle Variations for Meandering Wind Simulations

The meandering effect may be simulated more effectively using the wind library by performing three steady state RANS calculations, corresponding to the mean,

minimum and maximum approaching flow angles, and then blending these solutions in time using a saw-tooth, periodic function, for the contaminant field evolution. Figure 5 shows the maximum ground level concentrations obtained using an unsteady RANS and contaminant transport calculation where the temporal variation of the approaching flow angle is shown in Figure 4. Figure 6 shows the $\log_{10}$ maximum ground level concentration using the wind library meandering wind approach. The wind library approach required significantly less computational time than the unsteady case.

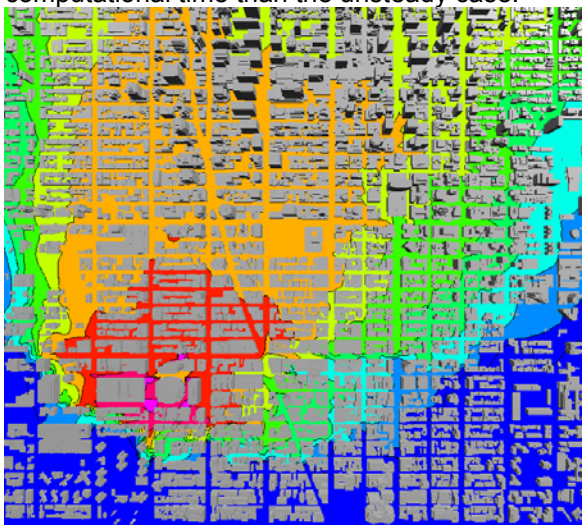

Figure 5: Unsteady calculation results, showing $\log_{10}$ of maximum ground level concentrations.

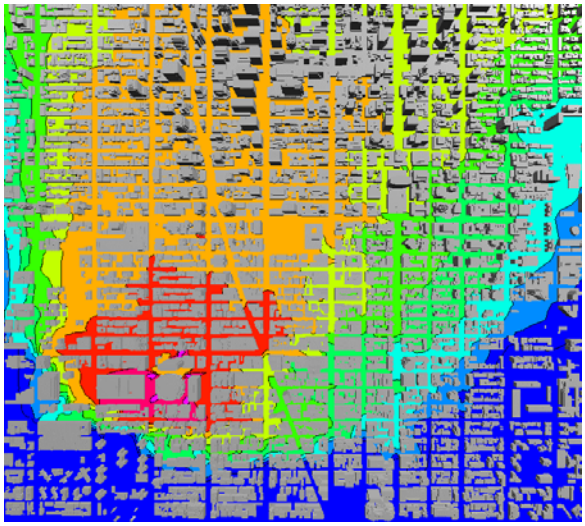

Figure 6: Wind library results, showing $\log_{10}$ of maximum ground level concentrations.

2.4 Post Processing Tools

Post processing tools are needed to convert the steady and unsteady flow, turbulence, and contaminant transport data into graphical images, plots and other forms. For visualization, we use specialized versions of CFD-VIEW, which is a visualization package for CFD models developed by ESI Group, NA. A collection of C++ and Python applications have been developed for CFD-Urban data processing that read the DTF common file format [Coirier et al., 1998] are available for nearly all needed post processing tasks.

3. DATA REQUIREMENTS

The data required for CFD-Urban is best described according to the tasks that use the data. To that end, we will define the data needed to generate the cityscape model, to supply as boundary condition data for the flow solution, and the source terms for the transport and dispersion solver.

3.1 Cityscape Model Generation Data Requirements

The data represented by the ESRI shp file format [ESRI] is sufficient to perform cityscape model generation. When performing the prismatic-hexahedral mesh generation, the voxel size is usually taken to be approximately 1 meter, so resolution below this size will not have any effect upon the mesh generation. Since both mesh generation approaches do not require the generation of a watertight surface representation, the requirements for model generation are less stringent than for most CFD model generators. Various components of the drag processing and model generation need to perform a point in polygon test, so we reorient all polygons from the shp file data to be positively oriented. Terrain and elevation data are needed on a fine resolution to be useful. We recommend that height attributed polygons, similar to the ESRI data, be stored in a community database, and furthermore, recommend that in addition to the data, a software library be provided to perform database queries of the cityscape related data. These queries should return building height, point in building test, as well as return aerodynamically relevant data, such as planform density and frontal areas [Grimmond et al., 1999].

3.2 Flow and Turbulence Boundary Conditions

The representation of proper flow and turbulence boundary conditions will continue to be a subject of research, since there is still much to be learned regarding the turbulent mass and momentum exchange processes in the urban areas. From our experiences we have found that it is critical that whatever velocity and turbulence profiles have been supplied, that they are in turbulence equilibrium in a manner consistent with the turbulence model and the governing flow equations. By assuming that the lateral gradients are negligible (which is clearly not valid inside an urban area), and that the production due to buoyancy is also negligible (which is only valid in the absence of temperature gradients) turbulence equilibrium states that the turbulence kinetic energy production by shear is balanced by the dissipation rate:

$$\mu_t \left(\frac{\partial U}{\partial z} \right)^2 = \rho \varepsilon \qquad (14)$$

This relation, using the velocity profile gradients, coupled with the definition of the eddy viscosity (9) is sufficient to supply turbulence kinetic energy and dissipation rates in equilibrium with the flow field. Wall boundary conditions using the wall function must also be consistent with the turbulence model, and for maintaining logarithmic profiles, the wall roughness length must be the same as that used in the inflow profile.

3.3 Contaminant Source Term Descriptions

Source location and strength as a function of time are typically all that is needed to solve the contaminant transport equation. For highly resolved calculations, it may be necessary to model the source term momentum fluxes. Multicomponent gases may be modeled, and thus, individual production/source rates must be specified. The modeling of a dense gas release near the source will require a multi-phase flow solution (which is not addressed here), but farther away from the source, where the particle loading is sufficiently low, the contaminant may be modeled as a dilute gas using the Eulerian approach (11).

4. CONCLUSIONS

We have described the CFD-Urban modeling tools that may be used to simulate the flow, turbulence and contaminant transport fields in urban areas. The finite-volume, Reynolds-Averaged Navier-Stokes formulation of the model have been described, as well as different techniques that are used to model buildings both explicitly (by resolving the building surfaces) and implicitly (by introducing momentum and turbulence source terms to represent building effects upon the flow fields). Turbulence closure using a standard k-ε model has been detailied, including modifications to the turbulence model and boundary conditions to better represent atmospheric boundary layers. Two different cityscape CFD model generation techniques that ingest GIS data have been outlined which respectively generate prismatic-hexahedral and quadtree-prismatic/octree grids. Data requirements for this model were defined, including; flow and turbulence boundary conditions, building geometric descriptions and contaminant source description.

5. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support for this work from the Defense Threat Reduction Agency/TDOC, Technical Monitors, Rick Fry, John Pace and Kibong Kim. The authors would also like to thank Jim Bowers of US Army/Dugway Proving Ground for his continued support.

6. REFERENCES

ACE+: CFD-ACE+, V2003 Users Manual, Volumes 1 and 2.

Anderson, D.A., Tannehill, J.C. and Pletcher, R.H.: 1984, *Computational Fluid Mechanics and Heat Transfer,* Hemisphere, New York.

Athavale, M.M., Jiang, Y., Przekwas, A.J.: 1995, Application of an unstructured grid solution methodology to turbomachinery flows, AIAA-95-0174.

Camelli, F., Coirier, W.J., Hansen, O.R., Huber, A., Kim, S., Hanna, S., Brown, M., An intercomprison of Four Computational Fluid Dynamics Models: Transport and Dispersion around Madison Square Garden,American Meteorological Society, 6th Symposium on the Urban Environment, 2006

Coirier, W.J., 1994: "An Adaptively Refined Euler and Navier-Stokes Solution on an

Unstructured Quadtree-Based Grid," Ph.D. Thesis, Aerospace Engineering Department, The University of Michigan, 1994.

Coirier, W.J., and Powell, K., 1995: "An Accuracy Assessment of Cartesian-Mesh Approaches for the Euler Equations," Journal of Computational Physics, Volume 117, pp. 121-131.

Coirier, W.J., et al., 1998: CFD-DTF: A data transfer facility for CFD and multidisciplinary analysis, AIAA-98-0125.

Coirier, W.J., Bayyuk, S.A., 2001: Simulation of Blast Waves in Urban Areas using a Hierarchical Solution-Adaptive Mesh Refinement Technique", DTRA SBIR Phase I Final Report: Contract: DTRA01-01-P-0161

Coirier, W.J., Furmancyzk, M., Przekwas, A.J., 2002.a: Development of an Image- and CFD-Based Urban Scale Wind Field and Dispersion Simulator, American Meteorological Society, 4th Symposium on the Urban Environment, p. J17, J1.11

Coirier, W.J., Bayyuk, S.A. 2002.b: "Urban Area Blast Wave Modeling using Hierarchical,Adaptive Mesh Refinement", American Institute of Aeronautics and Astronautics, Annual Meeting, AIAA-2002-2749.

Coirier, W., Reich, A., Fricker, D., Furmanczyk, M., Przekwas, A., 2003.a: *Application of CFD-Urban For Support of the Joint URBAN 2003 Field Test*, 7th Annual George Mason University Conference on Transport and Dispersion, Fairfax, Virginia.

Coirier, W.J., Fricker, D.M., and Furmanczyk, M.: 2003.b, Development of a high fidelity PC based simulator for modeling the atmospheric transport and dispersion of nuclear, chemical, biological and radiological substances in urban areas, DTRA SBIR Phase II Final Report: Contract: DTRAA01-01-0079

Coirier, W.J., Fricker, D.M., Furmanczyk, M., and Kim, S.: 2005.a. "A Computational Fluid Dynamics Approach for Urban Area Transport and Dispersion Modeling", in print, Environmental Fluid Mechanics, Vol. 15(5).

Coirier, W.J., Kim, S., 2005.b: Recent developments and applications of the cfd-urban transport and dispersion model, 9th Annual George Mason University Conference on Transport and Dispersion Modeling, July 18-20, Fairfax, Virginia.

Coirier, W.J., Kim, S., 2005.c: Detailed transport and dispersion calculations in support of the MSG05 field test, 9th Annual George Mason University Conference on Transport and Dispersion Modeling, July 18-20, Fairfax, Virginia.

Coirier, W.J., Kim, S. 2006.a. Summary of CFD-Urban Results in Support of the Madison Square Garden and Urban Disperison Program Field Tests, American Meteorological Society, 6th Symposium on the Urban Environment.

Coirier, W.J., Kim, S., Chen, F., Tuwari, M.,: 2006.b. Evaluation of urban scale contaminant transport and dispersion modeling using loosely coupled CFD and mesoscale models, American Meteorological Society, 6th Symposium on the Urban Environment.

Dettering, H.W., Etling, D., 1985: Application of the E-ε Turbulence Model to the Atmospheric Boundary Layer, Boundary Layer Meteorology, 33, pp 113-133.

ESRI, ESRI Shapefile Technical Description: An ESRI White Paper, available on the web at www.**esri**.com/library/whitepapers/pdfs/**shapefile**.pdf

Grimmond: Grimmond, C.S.B., Oke, T.R., "Aerodynamic Properties of Urban Areas Derived from Analysis of Surface Form", Journal of Applied Meteorology, Volume 38, pp 1262-1291, 1999.

Jiang, Y., Fricker, D.M., Coirier, W.J.: 1999, Parallelization of a fully implicit unstructured pressure-based flow solver using MPI, AIAA-99-3274.

Jiang, Y., Przekwas, A.J.: 1994, Implicit, pressure-based incompressible Navier-Stokes solver for unstructured meshes, AIAA-94-0303.

Karypis, G. and Kumar, K.: 1998, Multilevel K-way partitioning scheme for irregular graphs", *Journal of Parallel and Distributed Computing*, **48**(1):96-129.

Launder, B.E. and Spalding, D.B.: 1974, The numerical computation of turbulent flows, *Computer Methods in Applied Mechanics and Engineering*. **3**, 269-289.

Lacser, A., Otte, T.L., 2004: Implementation of an Urban Canopy Parameterization in a Mesoscale Meteorological Model, Journal of Applied Meteorology, Vol 43, Issue 11.

Liu, J., Chen, J.M., Black, T.A., Novak, M.D., 1996: "E-Epsilon Modeling of Turbulent Air Flow Downwind of a Model Forest Edge", Boundary-Layer Meteorology, V 77, 1996, pp 21-44

Ruppert, J.: 1995, A Delaunay “Refinement Algorithm for Quality 2-Dimensional Mesh Generation, *Journal of Algorithms*, Vol. 18, Issue 3, 548-585.

Schlichting, H., 1979, Boundary Layer Theory, 7th Edition, MacGraw-Hill Series in Mechanical Engineering

shapelib.maptools.org: Available on the world wide web at http://shapelib.maptools.org/

Shewchuk, J.R.: 1996.a, Triangle: engineering a 2D quality mesh generator and Delaunay triangulator”, First Workshop on Applied Computational Geometry, ACM.

Shewchuk, J.R.: 1996.b, Adaptive precision floating-point arithmetic and fast robust geometric predicates, Journal of Discrete & Computational Geometry, Vol. 18, No. 3, 305-363.

Sutherland, I.E., Hodgman, G.W., 1974: Reentrant Polygon Clipping, Communications of the ACM, Graphics and Image Processing, 17(1):32-42.

Yakhot, V., Orszag, S.A., Thangam, S., Gatski, T.B., And Speziale, C.G.: 1992, Development Of Turbulence Models For Shear Flows By A Double Expansion Technique, *Phys. Fluids A* **4,** 1510-1520.

.