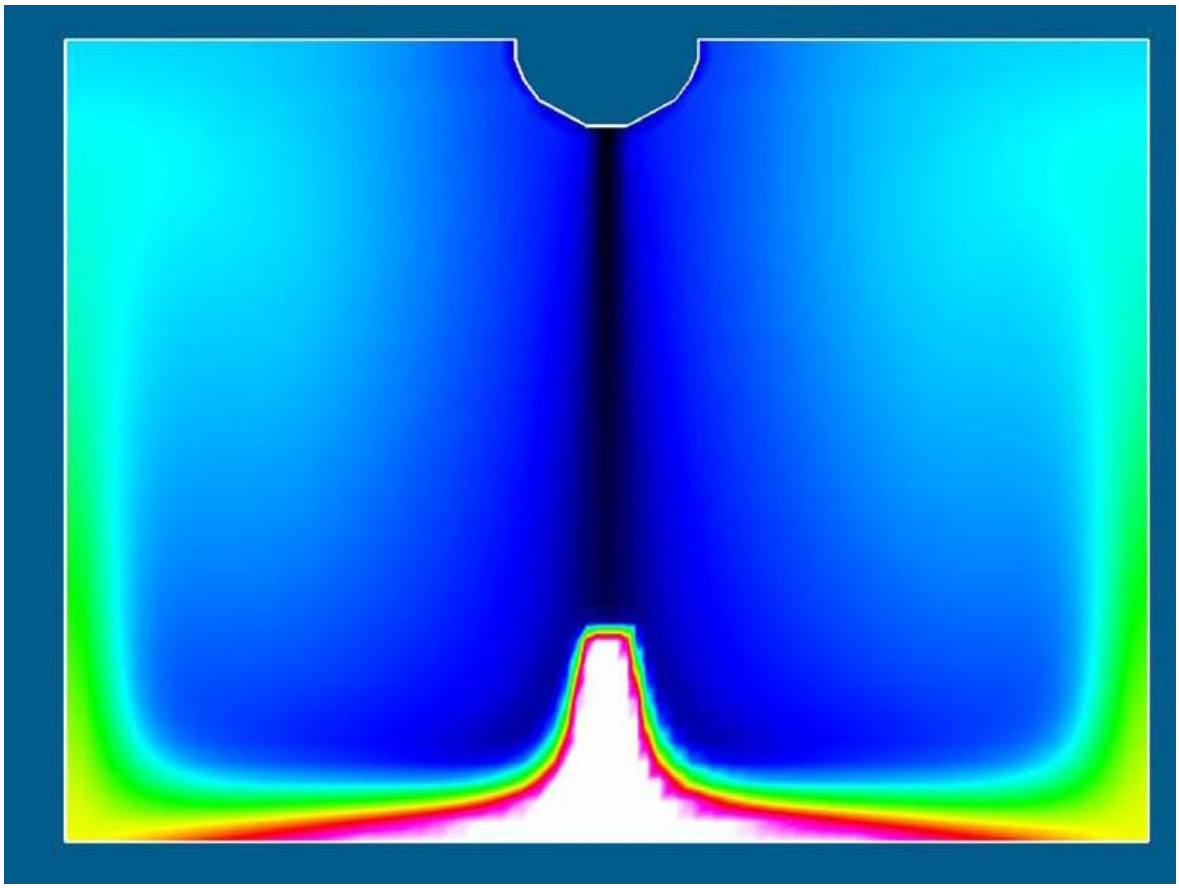
ERDC/CHL TR-07-9

US Army Corps
of Engineers®
Engineer Research and
Development Center

PAR3D: Numerical Model for Incompressible Flow with Application to Aerosol Dispersion in Complex Enclosures

Robert S. Bernard, Phu V. Luong, and Mario J. Sanchez

September 2007



Coastal and Hydraulics Laboratory

ERDC/CHL TR-07-9
September 2007

PAR3D: Numerical Model for Incompressible Flow with Application to Aerosol Dispersion in Complex Enclosures

Robert S. Bernard, Phu V. Luong, and Mario J. Sanchez

Coastal and Hydraulics Laboratory
U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Final report

Prepared for U.S. Army Corps of Engineers
Washington, DC 20314-1000

Abstract: This report documents the development of the PAR3D numerical flow model, with emphasis on modifications incorporated to facilitate simulations of contaminant dispersion in complex buildings and other enclosures. PAR3D is a general-purpose computational fluid dynamics (CFD) code for predicting three-dimensional flow and transport in air, water, and other incompressible fluids. It includes a two-equation turbulence model with adjustments for buoyancy, as well as transport equations for suspended materials (contaminants), dissolved gases, and gas bubbles (in water). The code employs parallel processing with structured curvilinear grids, which may deform to accommodate quasi-static free-surface displacement in water.

The CFD applications included here concern air flow and aerosol dispersion inside two different man-made enclosures. These consist of single chambers with complex internal geometry, but PAR3D is applicable to multi-chamber enclosures as well. The reported developments are part of a larger computational and experimental effort to characterize the dispersion of contaminants in multi-room, multi-story buildings. Comparison of PAR3D predictions with experimental data from a multi-room test facility will be presented in a separate report.

Contents

Figures

Preface

This report documents research and development conducted from October 2004 through December 2006 by personnel of the Coastal and Hydraulics Laboratory (CHL), U.S. Army Engineer Research and Development Center (ERDC). Funding for this effort was provided by the ERDC Construction Engineering Research Laboratory (CERL) under a research project entitled Chemical and Biological Protection of Facilities. The work reported here extends earlier research and development sponsored by the Chicago District, U.S. Army Corps of Engineers (USACE), from October 1994 through September 2004.

This report was prepared by Dr. Robert S. Bernard under the supervision of Earl V. Edris, Jr., Chief, Hydrologic Systems Branch; Dr. Phu V. Luong under the supervision of Dr. Robert T. McAdory, Chief, Estuarine Engineering Branch; and Mario J. Sanchez under the supervision of Dennis W. Webb, Chief, Navigation Branch. General supervision was provided by Thomas W. Richardson, Director, CHL; Dr. William D. Martin, Deputy Director, CHL; Bruce Ebersole, Chief, Flood and Storm Protection Division; and Dr. Rose M. Kress, Chief, Navigation Division.

Technical advice needed to complete this work was provided by Dr. Steven C. Wilhelms, CHL, Dr. Chang W. Sohn, CERL, Dr. John S. Gulliver, University of Minnesota, and Drs. Jeffrey Whicker and Snezana Konecni, Los Alamos National Laboratory.

Dr. William J. Croisant was the project manager for this effort, serving under the immediate supervision of Dr. Thomas Hartranft, Chief, Energy Branch, CERL. Dr. Ashok Kumar was the Program Manager, Martin J. Savoie was the Technical Director, and Dr. Ilker R. Adiguzel was the Director of CERL.

COL Richard B. Jenkins was Commander and Executive Director of ERDC. Dr. James R. Houston was Director.

Unit Conversion Factors

Multiply	By	To Obtain
cubic feet	0.02831685	cubic meters
feet	0.3048	meters
square feet	0.09290304	square meters

1 Introduction

Background

The PAR3D code (Bernard et al. 1999) represents a continuation of work that began with the MAC3D code (Bernard 1995, 1998) which was developed for simulating flow and transport (i.e., dispersion) in deep water. PAR3D retains all the capabilities of MAC3D, and it incorporates a number of improvements and extensions as well. Foremost among these is parallel processing, which reduces central processing unit (CPU) time and increases allowable resolution (problem size) by using multiple processors for executing flow-and-transport simulations. Like the MAC3D code, PAR3D employs multi-component curvilinear grids, but it also allows quasi-static grid deformation to account for free-surface displacement in water. For deep-water applications in particular, the capabilities of PAR3D include salinity transport, sediment and biomass transport, and dissolved-gas transfer and transport induced by bubble plumes and mechanical mixers.

Water and air can both be regarded as incompressible fluids at flow velocities much less than the speed of sound. Therefore, since PAR3D is formulated for incompressible flow in general, it is equally applicable for water and air at low Mach number. With this fact in mind, the code has been extended to facilitate simulations of air flow and contaminant dispersion in complex buildings and other enclosures. Internal walls are represented as having either finite or infinitesimal thickness, and dispersed contaminants can be reintroduced through supply vents after being drawn into the return vents of a heating, ventilation, and air conditioning (HVAC) system. Contaminant release can be either continuous or discontinuous (impulsive), and the boundary conditions for all variables can be altered dynamically in the course of a simulation.

Computational fluid dynamics (CFD) is now a mature branch of computational physics, which encompasses both compressible and incompressible flow. PAR3D is limited to incompressible flow in particular, which means that the fluid density is *not* affected by pressure. Even so, fluid density may be altered (slightly) by temperature, salinity, and suspended materials. In the presence of gravity, small variations in density may create buoyant forces that dramatically influence flow and transport, so PAR3D

employs the Boussinesq approximation. This means that small density variations are neglected in the continuity equation but not in the momentum equation. Whatever the case, the speed of sound is infinite for incompressible fluids, and this is the reason for the practical restriction to flow situations with low Mach number.

PAR3D uses a finite-volume scheme with structured marker-and-cell (MAC) grids to compute the transport of mass and momentum through flow regions of arbitrary shape. Structured grids employ grid cells that are numbered with integer indices (*i,j,k*), which also serve as the grid-cell coordinates (*i,j,k*) in a rectangular computational space. The MAC designation indicates that vector components are computed normal to the cell faces, whereas scalar quantities are computed exclusively at the cell centers. Face-normal (contravariant) vector components represent average values for the entire cell face, and cell-centered scalar quantities represent average values for the entire grid cell. The grid cells are all cubical in computational (*i,j,k*) space, but they may take any six-sided shape (with planar faces) in Cartesian (*x,y,z*) space. The latter feature gives PAR3D much geometric flexibility in accommodating flow regions with non-rectangular boundaries.

The grids are subdivided into multiple components that represent rectangular blocks of computational (*i,j,k*) space, and each grid block is assigned to a single processor. Shared information is communicated between processors via the standard message-passing interface (MPI) library, which is described in detail by Gropp et al. (1996).

PAR3D uses a time-marching scheme to solve the Reynolds-averaged Navier-Stokes (RANS) equations for incompressible flow. Starting with potential flow as an initial condition, the code uses discrete time-steps to advance the solution toward its final condition, which may be either steady state or time-varying. Advective terms in the RANS momentum equation are evaluated explicitly (time-lagged) to achieve a better approximation for convective transport, and diffusive terms are evaluated implicitly (time-advanced) to allow larger time-steps. A Poisson equation for pressure, obtained by combining the momentum and continuity equations, is solved iteratively during each time-step to supply the pressure gradient needed for conservation of mass. Transport equations for other variables are solved in the same manner as the momentum equation.

The influence of turbulence is incorporated via the standard κ–ε turbulence model developed by Launder and Spalding (1974). The model consists of transport equations (which include production and dissipation terms) for the turbulence energy κ and the turbulence-energy dissipation rate ε. These quantities are used to calculate eddy viscosity in the RANS momentum equation, and eddy diffusivity in the associated transport equations for other variables.

Purpose and scope

This report documents the continuing development of PAR3D and its tentative validation for applications involving particle dispersion in air. Earlier reports documenting MAC3D emphasized deep-water applications for reservoirs. In keeping with the motivation for recent developments, this report focuses on flow and dispersion in complex (man-made) enclosures. Experimental validation is limited here to single-chamber enclosures with complex internal geometry, but the PAR3D code is equally applicable to multi-chamber enclosures. The current work was part of a larger computational and experimental effort to characterize contaminant dispersion in multi-room, multi-story buildings. Comparison of PAR3D predictions with data from experiments conducted in a multi-room HVAC test facility will be documented in a separate report.

Chapter 2 outlines the physical governing equations, and Chapter 3 discusses their numerical solution. Chapter 4 enumerates the various boundary conditions that are imposed on the flow field, and Chapter 5 gives an overview of model operation. Chapter 6 presents comparisons of experimental data with PAR3D predictions for air flow and aerosol dispersion in two different enclosures. Chapter 7 offers concluding remarks, and the Appendix develops the boundary conditions proposed for turbulence in Chapter 4.

Mode of technology transfer

As part of a larger research effort, the results of this work have been applied to the simulation of air flow and contaminant transport in several facilities, including an operational military building. Some of these results will be disseminated through technical papers presented at professional conferences devoted to the modeling of fluid flow and contaminant transport. Upon publication, this report will be made immediately available to its sponsor. It will be accessible to the public through the World Wide Web at *http://itl.erdc.usace.army.mil/library/*.

2 Governing Equations

Incompressible flow

Subject to the Boussinesq approximation, which accounts for density variation only in the gravitational acceleration term, the governing equations for incompressible flow are the RANS momentum equation,

$$\rho_0 \left[\frac{\partial \vec{u}}{\partial t} + div\left(\vec{u}\vec{u}\right) \right] = div\ \vec{\vec{\tau}} - grad\ p + \rho \vec{g} \tag{2.1}$$

and the continuity equation,

$$div\ \vec{u} = 0 \tag{2.2}$$

where

ρ_o = reference density
$\vec{u}$ - Reynolds-averaged velocity vector
t = time
$\vec{\vec{\tau}}$ = Reynolds-averaged shear-stress tensor
p = Reynolds-averaged static pressure
ρ = local density
$\vec{g}$ = gravitational acceleration vector

Invoking the widely used Boussinesq hypothesis for turbulence, the individual shear-stress tensor components τ_{mn} take the form

$$\tau_{mn} = \mu \left(\frac{\partial u_m}{\partial x_n} + \frac{\partial u_n}{\partial x_m} \right) - \frac{2}{3} \rho_0\, k\, \delta_{mn} \tag{2.3}$$

where

μ = dynamic viscosity = $\mu_o + \mu_T$
μ_o = molecular viscosity
μ_T = eddy viscosity
u_m and u_n= Cartesian velocity components $(u_1, u_2, u_3) \equiv (u, v, w)$
x_m and x_n= Cartesian coordinates $(x_1, x_2, x_3) \equiv (x, y, z)$
k = turbulence energy per unit mass
δ_{mn} = unit-normal tensor components

The symbol δ_{mn} represents the familiar Kroniker delta, which is unity for $m = n$ and zero otherwise. For practical applications involving turbulent flow, one may assume that $\mu_T >> \mu_0$ so that the dynamic viscosity reduces to

$$\mu \approx \mu_T = \rho_0 \nu_T \tag{2.4}$$

in which ν_T represents the kinematic eddy viscosity. For transported quantities ϕ other than momentum, the Reynolds-averaged transport equation is

$$\frac{\partial \varphi}{\partial t} + div\left[\left(\bar{u} + \bar{u}_\varphi\right)\varphi\right] = div\left(D_\varphi \, grad\, \varphi\right) + S\left(\varphi\right) \tag{2.5}$$

where

$\bar{u}_\phi$ = settling velocity (if any) for ϕ

D_ϕ = eddy-diffusion coefficient = $\dfrac{\nu_T}{\sigma_\phi}$

σ_ϕ = Prandtl-Schmidt number for ϕ

S(ϕ) = source term for ϕ

Turbulence

The k–ε model developed by Launder and Spalding (1974) defines the kinematic eddy viscosity ν_T to be

$$\nu_T = C_\nu \frac{k^2}{\varepsilon} \tag{2.6}$$

where

C_ν = 0.09

k = turbulence energy per unit mass

ε = dissipation rate for k

The coefficient C_ν is a universal (empirical) coefficient, and the governing equations for k and ε are respectively formulated as

$$\frac{\partial k}{\partial t} + div\left(\bar{u}\, k\right) = P - \varepsilon + \frac{1}{\sigma_k} div\left(\nu_T \, grad\, k\right) \tag{2.7}$$

$$\frac{\partial \varepsilon}{\partial t} + div\left(\bar{u}\,\varepsilon\right) = \frac{\varepsilon}{k}\left(c_1 P - c_2 \varepsilon\right) + \frac{1}{\sigma_\varepsilon} div\left(\nu_T\, grad\, \varepsilon\right) \tag{2.8}$$

where P is the rate of production for the turbulence energy, and c_1, c_2, σ_k, and σ_ε are empirical coefficients.

To reproduce the observed equilibrium between production, dissipation, and diffusion for wall-bound shear flow (without a pressure gradient), as discussed by Rodi (1980), the coefficients c_1, c_2, and σ_ε must satisfy the relation,

$$c_1 - c_2 + \frac{\kappa^2}{\sigma_\varepsilon \sqrt{C_\nu}} = 0 \tag{2.9}$$

The symbol κ represents von Karman's constant, whose value lies between 0.37 and 0.44. In PAR3D this quantity is assumed to be $\kappa = 0.40$.

In the standard k–ε model, the coefficients c_2, σ_k, and σ_ε are assigned the values

$c_2 = 1.92$
$\sigma_k = 1.0$
$\sigma_\varepsilon = 1.3$

and it then follows from Equation 2.9 that the remaining coefficient c_1 is

$$c_1 = c_2 - \frac{\kappa^2}{\sigma_\varepsilon \sqrt{C_\nu}} \tag{2.10}$$

The rate of turbulence energy production, which arises from shear and buoyancy in the mean flow, is given by

$$P = \nu_T \left[2\left(u_x{}^2 + v_y{}^2 + w_z{}^2\right) + \left(u_y + v_x\right)^2 + \left(u_z + w_x\right)^2 + \left(v_z + w_y\right)^2 + \frac{\rho_z g}{\rho_0} \right] \tag{2.11}$$

in which the subscripts (x, y, z) indicate partial derivatives of the Cartesian velocity components (u, v, w) and the fluid density ρ. The symbol g denotes gravitational acceleration (downward) in the vertical (upward) z-direction. Note here that stable density stratification (ρ_z<0) *reduces*

production, whereas unstable density stratification (ρ_z>0) has the opposite effect.

The k–ε model provides approximations for all six components of the Reynolds-averaged shear stress, along with governing equations for the associated turbulence quantities k and ε; but it does *not* provide boundary conditions for the latter. These must be supplied separately, based on other considerations discussed in the appendix.

Fluid density

The fluid under consideration (air or water) is presumed to be incompressible, which means that its density is unaffected by pressure. In spite of its incompressibility, the effective density of water may be altered by temperature, salinity, gas bubbles, and suspended materials (sediments or contaminants). Likewise, the effective density of air may be altered by temperature and suspended contaminants. For air and water, PAR3D imposes a reference density defined at a predetermined absolute temperature T_0; i.e.,

$$\rho_0 = \rho(T_0) \tag{2.12}$$

For air at (or near) atmospheric pressure, the effective density is inversely proportional to absolute temperature T (in degrees Kelvin) and directly proportional to contaminant mass-concentration ϕ (in kilograms per cubic meter) such that

$$\rho \approx \rho_0 \frac{T_0}{T} + \varphi\left(1 - \frac{\rho_0 T_0}{\rho_c T}\right) \tag{2.13}$$

where

$\rho_0 \approx 1.27$ Kg/m^3
ρ_c = bulk density of contaminant in kilograms per cubic meter
$T_0 = 277$ degrees Kelvin

The effective density for water is likewise proportional to contaminant concentration ϕ and void ratio ϑ, but its dependence on temperature T and salinity θ is nonlinear; i.e.,

$$\rho \approx \rho_0 (1 - \vartheta) + \rho_1 (T) + \rho_2 (T, \theta) + \varphi \left(1 - \frac{\rho_0}{\rho_c} \right) \tag{2.14}$$

where

$\rho_0 \approx$ 1000 Kg/m3
$\vartheta =$ void ratio for gas bubbles
$\rho_1(T) =$ -0.009073$|T - T_0|$ - 0.006047 $(T - T_0)^2$
$\rho_2(T,\theta) =$ [1.0 – 0.001779$(T - T_1)$] [0.8076θ]
$T_1 =$ 273 degrees Kelvin

The term "void ratio" refers here to the local ratio of bubble volume to composite volume (water *and* bubbles). The density variation with void ratio and contaminant concentration is necessarily linear, because gas bubbles (in water) and suspended materials (in air or water) fully displace the fluids in which they are suspended. In contrast, salt goes into solution, and it does not fully displace the water in which it is dissolved. Note that the salinity θ must be specified in parts per thousand (grams per kilogram) in the definition for $\rho_2(T,\theta)$. For temperatures from 273 to 313 degrees Kelvin (0 to 40 degrees Celsius) and salinities from 0 to 40 parts per thousand, the combined functions $\rho_1(T)$ and $\rho_2(T,\theta)$ reproduce (within about 6 percent) the density variation specified by a more complicated equation of state published by Fofonoff (1985).

Curvilinear coordinates

In the PAR3D code, the governing equations (Equations 2.1, 2.2, 2.5, 2.7, and 2.8) are transformed from Cartesian coordinates (x,y,z) to general curvilinear coordinates (ξ,η,ζ) to facilitate applications with non-rectangular boundaries. The one-to-one coordinate transformation allows each of the Cartesian coordinates to be expressed directly as a function of the curvilinear coordinates, such that

$x = x(\xi,\eta,\zeta)$
$y = y(\xi,\eta,\zeta)$
$z = z(\xi,\eta,\zeta)$

It then follows that the x-, y-, and z-derivatives of any variable ϕ are related to its ξ-, η-, and ζ-derivatives by the chain rule. Thus, the transformed x-, y-, and z-components of *grad* ϕ become

$$\varphi_x = \xi_x \varphi_\xi + \eta_x \varphi_\eta + \zeta_x \varphi_\zeta \tag{2.15}$$

$$\varphi_y = \xi_y \varphi_\xi + \eta_y \varphi_\eta + \zeta_y \varphi_\zeta \tag{2.16}$$

$$\varphi_z = \xi_z \varphi_\xi + \eta_z \varphi_\eta + \zeta_z \varphi_\zeta \tag{2.17}$$

Here subscripts (x,y,z) and (ξ,η,ζ) indicate derivatives with respect to the Cartesian and curvilinear coordinates, respectively. As discussed by Anderson et al. (1984), the Cartesian derivatives of the curvilinear coordinates are given by

$$\xi_x = J^{-1}\left(y_\eta z_\zeta - y_\zeta z_\eta\right) \tag{2.18}$$

$$\xi_y = -J^{-1}\left(x_\eta z_\zeta - x_\zeta z_\eta\right) \tag{2.19}$$

$$\xi_z = J^{-1}\left(x_\eta y_\zeta - x_\zeta y_\eta\right) \tag{2.20}$$

$$\eta_x = -J^{-1}\left(y_\xi z_\zeta - y_\zeta z_\xi\right) \tag{2.21}$$

$$\eta_y = J^{-1}\left(x_\xi z_\zeta - x_\zeta z_\xi\right) \tag{2.22}$$

$$\eta_z = -J^{-1}\left(x_\xi y_\zeta - x_\zeta y_\xi\right) \tag{2.23}$$

$$\zeta_x = J^{-1}\left(y_\xi z_\eta - y_\eta z_\xi\right) \tag{2.24}$$

$$\zeta_y = -J^{-1}\left(x_\xi z_\eta - x_\eta z_\xi\right) \tag{2.25}$$

$$\zeta_z = J^{-1}\left(x_\xi y_\eta - x_\eta y_\xi\right) \tag{2.26}$$

The symbol J represents the Jacobian determinant of the coordinate transformation,

$$J = \frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)} = \begin{vmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{vmatrix} \tag{2.27}$$

Expressed in expanded form, J becomes

$$J = x_\xi \left(y_\eta z_\zeta - y_\zeta z_\eta \right) - x_\eta \left(y_\xi z_\zeta - y_\zeta z_\xi \right) + x_\zeta \left(y_\xi z_\eta - y_\eta z_\xi \right) \tag{2.28}$$

Under the coordinate transformation outlined above, the curvilinear expression for the divergence of any Cartesian vector $\vec{u} \equiv (u,v,w)$ is given by

$$div\, \vec{u} = J^{-1} \left(\tilde{u}_\xi + \tilde{v}_\eta + \tilde{w}_\zeta \right) \tag{2.29}$$

where

$$\tilde{u} = J\left(\xi_x u + \xi_y v + \xi_z w \right) \tag{2.30}$$

$$\tilde{v} = J\left(\eta_x u + \eta_y v + \eta_z w \right) \tag{2.31}$$

$$\tilde{w} = J\left(\zeta_x u + \zeta_y v + \zeta_z w \right) \tag{2.32}$$

In PAR3D the Cartesian velocity components (u,v,w) are retained (for convenience) as dependent variables in the transformed RANS momentum equation, but the transformed velocity components $(\tilde{u},\tilde{v},\tilde{w})$ are used in the transformed continuity equation. The latter represent the proper components for the time-dependent velocity field in the curvilinear coordinate system, but their time increments are constructed from the more conveniently computed increments of the Cartesian components, as discussed in Chapter 3.

3 Numerical Considerations

Explicit upwind scheme for advection

All transport equations in the PAR3D code retain the conservative form $div(\vec{u}\,\phi)$ for advective (i.e., convective) terms. In principle the continuity equation (Equation 2.2) reduces these to $\vec{u} \bullet grad\,\phi$; but in practice the conservative form achieves a better representation of advective transport on non-uniform grids. In any case, a one-dimensional (1-D) example is used here for illustrating the conservative discretizaton of advective terms in general. Consider the following 1-D equation for transport by advection:

$$\frac{\partial \varphi}{\partial t} + \frac{\partial}{\partial \eta}(\upsilon\varphi) = 0 \tag{3.1}$$

where

ϕ = transported quantity
t = time
υ = transporting velocity in the η-direction

Let the time and space increments be Δt and $\Delta\eta$, respectively; and let the integer subscript j indicate discrete values ϕ_j defined at stations j-1, j, j+1, etc., along the η-axis. Furthermore, let the integer superscript n indicate discrete values of ϕ at time levels n-1, n, n+1, etc. Imposing a forward approximation for the time derivative of ϕ, along with a central approximation for the space derivative of $\upsilon\phi$, one obtains a discrete approximation for Equation 3.1 at time level n and position j,

$$\frac{\Delta\varphi_j^n}{\Delta t} + \frac{\upsilon_{j+1/2}^n \varphi_{j+1/2}^n - \upsilon_{j-1/2}^n \varphi_{j-1/2}^n}{\Delta\eta} = 0 \tag{3.2}$$

If position j is regarded as the center of a 1-D grid cell, with positions $j \pm 1/2$ as the corresponding cell faces, then Equation 3.2 represents a finite-volume approximation for Equation 3.1 in 1-D space. Inconveniently, however, the dependent variable ϕ has been discretized only at integer positions j, and its intermediate (cell-face) values at positions $j \pm 1/2$ can be obtained only by interpolation or extrapolation.

PAR3D obtains these values by second-order upwind extrapolation. For example, when $\upsilon^n_{j+1/2} > 0$ then

$$\varphi^n_{j+1/2} \equiv \frac{3}{2}\varphi^n_j - \frac{1}{2}\varphi^n_{j-1} \tag{3.3}$$

But when $\upsilon^n_{j+1/2} < 0$ then

$$\varphi^n_{j+1/2} \equiv \frac{3}{2}\varphi^n_{j+1} - \frac{1}{2}\varphi^n_{j+2} \tag{3.4}$$

In the parlance of numerical analysis, this is an explicit upwind scheme, which is first-order correct in time, second-order correct in space, and marginally stable for $C < 2/3$, where C is the Courant number,

$$C = \frac{\upsilon \Delta t}{\Delta \eta} \tag{3.5}$$

PAR3D uses second-order upwind extrapolation (Equations 3.3 and 3.4) for all dependent variables *except k* and ε, for which it uses first-order upwind extrapolation instead. The nonlinear production term P renders Equations 2.7 and 2.8 more prone to instability than the other governing equations, but first-order extrapolation keeps their discrete solutions numerically stable.

Implicit central scheme for diffusion

When expressed in expanded form, the shear-stress terms arising from *div* $\vec{\tau}$ in Equation 2.1, and likewise the diffusion terms in Equations 2.5, 2.7, and 2.8, all include terms of the form D_ϕ *div grad* ϕ. These terms are dissipative, and they contribute to numerical stability when $C << 1$; but they can also hamper computational speed by reducing the allowable size of the time-step Δt. Consider, for example, the 1-D advection-diffusion equation (ADE):

$$\frac{\partial \varphi}{\partial t} + \upsilon \frac{\partial \varphi}{\partial \eta} = D \frac{\partial^2 \varphi}{\partial \eta^2} \tag{3.6}$$

Here the coefficient D represents the diffusion coefficient, and the other quantities are the same as in Equation 3.1. Using a three-point upwind

approximation (assuming $\upsilon > 0$) for the advective term, and a central approximation for the diffusion term, the explicit approximation for Equation 3.6 becomes

$$\frac{\Delta\varphi_j^n}{\Delta t} + \frac{\upsilon}{2}\left[\frac{3\varphi_j^n - 4\varphi_{j-1}^n + \varphi_{j-2}^n}{\Delta\eta}\right] = D\left[\frac{\varphi_{j+1}^n - 2\varphi_j^n + \varphi_{j-1}^n}{(\Delta\eta)^2}\right] \tag{3.7}$$

As a rough rule of thumb, one can expect Equation 3.7 to be numerically stable as long as

$$\frac{3}{2}C + 2\Gamma < 1 \tag{3.8}$$

where C is given by Equation 3.5, and

$$\Gamma = \frac{D\Delta t}{(\Delta\eta)^2} \tag{3.9}$$

The ADE stability rule (Equation 3.8) depends on both C and Γ, but the advective component is inversely proportional to $\Delta\eta$, whereas the diffusive component is inversely proportional to $(\Delta\eta)^2$. This means that the time-step allowed by the advective term decreases with the grid spacing, but the time-step allowed by the diffusive term decreases with the *square* of the grid spacing. For turbulent flow simulations, where diffusion can dominate advection (locally) by an order of magnitude or more, the diffusive time-step limit may be far more confining than the advective (Courant) limit. While it may be desirable to use an explicit (time-lagged) approximation for advection terms, to achieve better accuracy for advective transport, it is likewise expedient to include an implicit (time-advanced) contribution from diffusion terms, to circumvent the diffusive time-step limitation. The discretization employed by PAR3D is equivalent to the following two-level representation for the ADE:

$$\frac{\Delta\varphi_j^n}{\Delta t} + \frac{\upsilon}{2}\left[\frac{3\varphi_j^n - 4\varphi_{j-1}^n + \varphi_{j-2}^n}{\Delta\eta}\right] = D\left[\frac{\varphi_{j+1}^{n+1} - 2\varphi_j^{n+1} + \varphi_{j-1}^{n+1}}{(\Delta\eta)^2}\right] \tag{3.10}$$

where $\varphi_j^{n+1} = \varphi_j^n + \Delta\varphi_j^n$.

Since implicit treatment of the diffusion term introduces $\Delta\phi_j^n$ on both sides of Equation 3.10, it follows that multiple iterations are required for $\Delta\phi_j^n$ in each time-step. For this purpose, PAR3D uses a symmetric Gauss-Seidel scheme, each iteration of which updates $\Delta\phi_j^n$ (cell-by-cell) in a forward sweep from the origin to the uppermost corner of the grid, followed by a reverse sweep from the uppermost corner back to the origin. The aim of this process is numerical stability rather than iterative convergence for $\Delta\phi_j^n$ at time level n. Therefore, only a few Gauss-Seidel iterations are needed within each time-step.

Velocity interpolation on MAC grids

PAR3D uses MAC grids that are staggered in their discrete placement of vector and scalar quantities. This means that the discrete values for variables such as pressure and temperature are formally defined only at grid-cell centers, whereas the normal (contravariant) components of vectors such as velocity are formally defined only on the cell faces from which they emanate. The most convenient locations for computing velocity increments are the cell centers; but this requires an additional closure relation that defines cell-centered velocity components in terms of face-centered velocity components, and face-centered velocity increments in terms of cell-centered velocity increments. The following 1-D example illustrates the closure relation that is used in PAR3D.

Let the integer locations j-1, j, j+1 represent cell centers along the η-axis, and let the half-integer locations $j \pm 1/2$ represent cell faces. The discrete velocities $v^n_{j\pm1/2}$ are formally defined only on cell faces; but when they are needed at the cell centers, they are computed by simple averaging; i.e.,

$$v_j^n \equiv \frac{1}{2}\left[v_{j+1/2}^n + v_{j-1/2}^n \right] \tag{3.11}$$

The velocity increments Δv_j^n are formally defined only at the cell centers; but when they are needed to update face-centered velocities, they are projected onto the cell faces by fourth-order interpolation; e.g.,

$$\Delta v_{j+1/2}^n = \frac{1}{16}\left[9\Delta v_{j+1}^n - \Delta v_{j+2}^n + 9\Delta v_j^n - \Delta v_{j-1}^n \right] \tag{3.12}$$

The interpolated increments defined by Equation 3.12 are the ones that are used to advance the face-centered velocities from one time level to the next; i.e,

$$\upsilon^{n+1}_{j+1/2} = \upsilon^{n}_{j+1/2} + \frac{1}{16}\left[9\Delta\upsilon^{n}_{j+1} - \Delta\upsilon^{n}_{j+2} + 9\Delta\upsilon^{n}_{j} - \Delta\upsilon^{n}_{j-1}\right] \tag{3.13}$$

The closure relation (Equation 3.12) between face-centered and cell-centered velocity increments retains the convenience of cell-centered time-discretization for all variables. That is, the time-dependent governing equations (Equations 2.1, 2.5, 2.7, and 2.8) are solved numerically as though they were *all* discretized only at the cell centers. In the PAR3D analog for Equation 3.12, increment $\Delta\upsilon^{n}_{j+1/2}$ gives way to the face-centered curvilinear increments $(\Delta\tilde{u},\Delta\tilde{v},\Delta\tilde{w})$ computed from the cell-centered Cartesian increments (Δu, Δv, Δw). In the transformed continuity equation, curvilinear velocity components $(\tilde{u},\tilde{v},\tilde{w})$ are formally defined only on the cell faces, and *these* are the velocity components that represent the evolving flow field. In any case, the formal definition of velocity components *only* on cell faces precludes numerical oscillations that may arise from the continuity equation when velocity components are formally defined at the cell centers instead.

Application of the pressure gradient

In the manner discussed in previous reports (Bernard 1995, 1998) the discrete pressure gradient is omitted from the discretized momentum equation, only to be computed and imposed *after* provisional (cell-centered) velocity increments have been projected onto the cell faces. This pressure gradient enforces mass conservation, and it lags the velocity by one time-step. For illustration, consider the inviscid Euler equation with unit density,

$$\frac{\partial\vec{u}}{\partial t} + div(\vec{u}\vec{u}) = -grad\, p \tag{3.14}$$

and let $\vec{u}$ be constrained by the continuity equation (Equation 2.2). The existing velocity $\vec{u}^{\,n}$ at time level n is used first to compute a provisionally updated velocity $\vec{u}^{\,*}$, given by

$$\vec{u}^{\,*} = \vec{u}^{\,n} - \Delta t\, div(\vec{u}^{\,n}\vec{u}^{\,n}) \tag{3.15}$$

where $div\ \vec{u}^* \neq 0$

The updated (mass-conserving) velocity $\vec{u}^{n+1}$ is then computed as

$$\vec{u}^{n+1} = \vec{u}^* - \Delta t\ grad\ p^n \tag{3.16}$$

where the constraint $div\ \vec{u}^{n+1} = 0$ requires that p^n satisfies a discrete Poisson equation for pressure,

$$div\ grad\ p^n = \frac{div\ \vec{u}^*}{\Delta t} \tag{3.17}$$

PAR3D employs a conjugate-gradient iteration scheme developed by Kapitza and Eppel (1987) to solve the Poisson equation for p^n. The same iteration scheme is used to solve a similar equation,

$$div\ grad\ \Phi = div\ \vec{u}^* \tag{3.18}$$

where $\vec{u}^*$ now represents some arbitrarily specified initial velocity, and the required mass-conserving initial velocity for time level $n = 0$ is given by

$$\vec{u}^0 = \vec{u}^* - grad\ \Phi \tag{3.19}$$

Computational geometry

To facilitate the implementation of curvilinear coordinates with structured MAC grids, PAR3D identifies the coordinates (ξ,η,ζ) with the ***integer*** indices (i,j,k) such that

$$\begin{aligned} \xi &\Rightarrow i \\ \eta &\Rightarrow j \\ \zeta &\Rightarrow k \end{aligned}$$

These grids may have almost any shape in Cartesian space, with arbitrary spacing $(\Delta x,\Delta y,\Delta z)$ between the grid nodes. Regardless of shape, however, they have *unit* spacing in computational (i,j,k) space; i.e.,

$$\begin{aligned} \Delta\xi &= \Delta i = 1 \\ \Delta\eta &= \Delta j = 1 \end{aligned}$$

$$\Delta\zeta = \Delta k = 1$$

Figure 1 shows a single non-rectangular grid cell in Cartesian space and the same cell in computational space, where it is perfectly cubical. To further illustrate the correspondence between Cartesian and computational spaces, Figure 2 shows a grid surface with non-rectangular edges in the Cartesian (x, y) plane and the same grid surface projected onto the computational (i, j) plane.

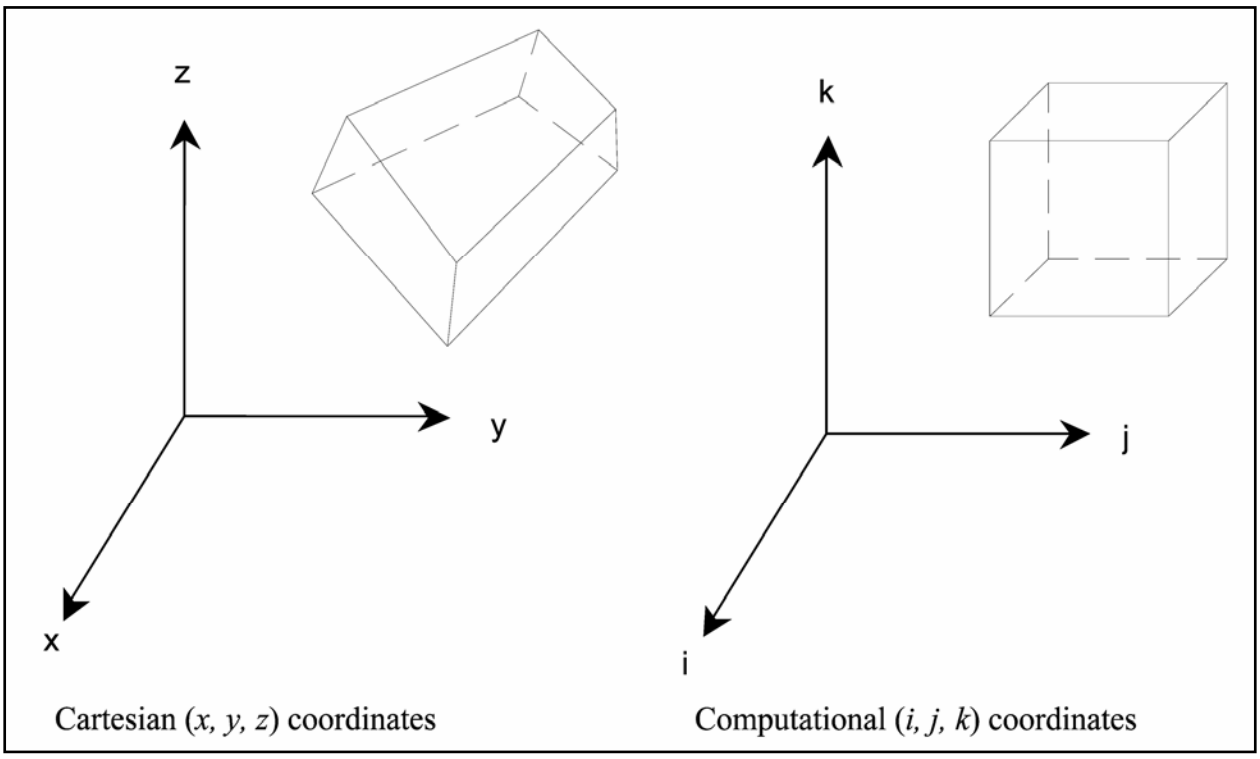


Figure 1. Grid cells in Cartesian and computational spaces.

Grids may be subdivided into two or more contiguous components, which may have any shape in Cartesian space, but which have rectangular shapes in computational space. These components are called grid blocks, and grids containing more than one block are called multi-block grids. Each block occupies a distinct region of Cartesian space, which resides on a separate computer processor with its own (i, j, k) coordinate system. Figure 3 shows two (contiguous) grid blocks with cutouts that represent internal solid obstacles.

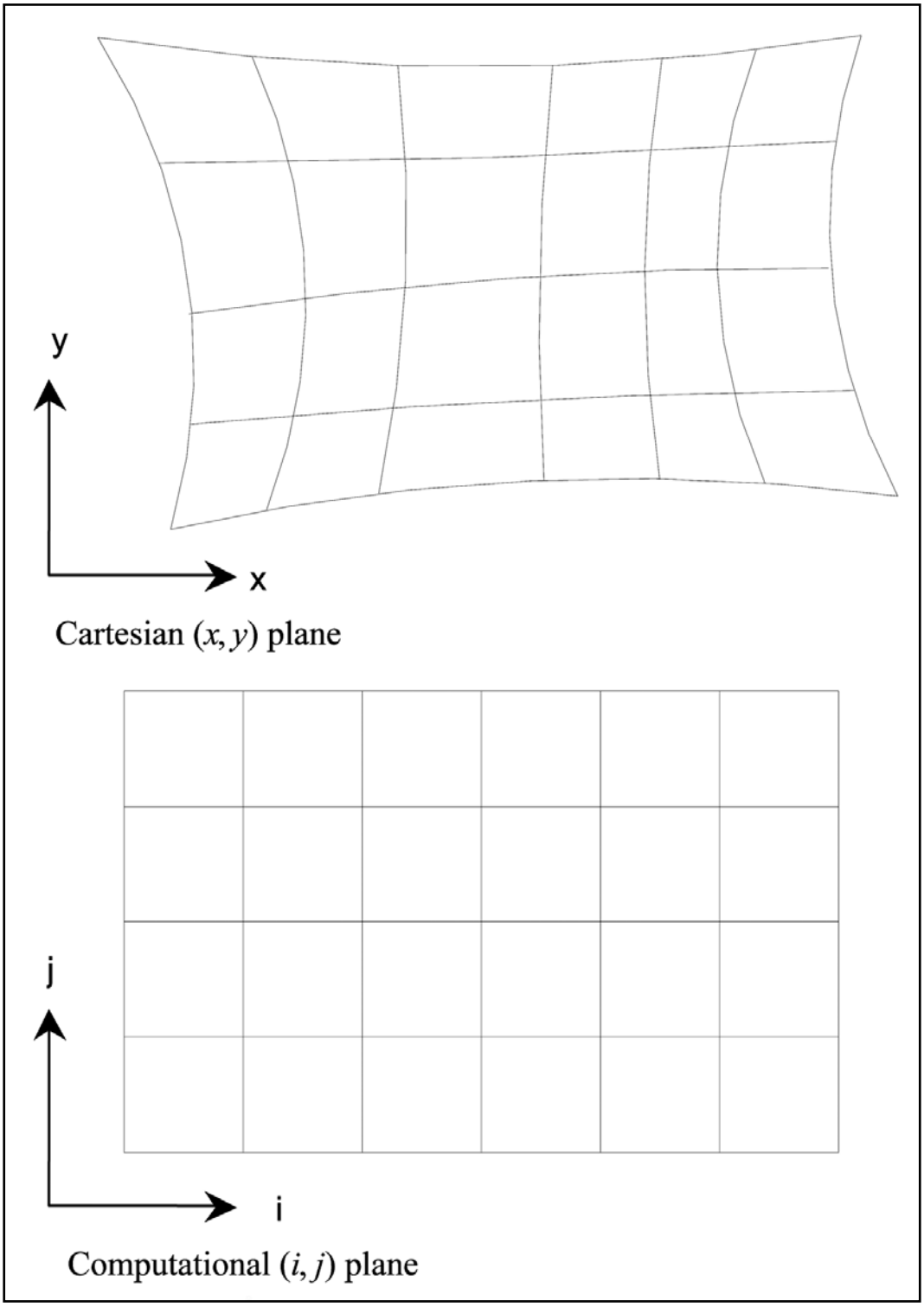


Figure 2. Grid surface projection onto Cartesian and computational planes.

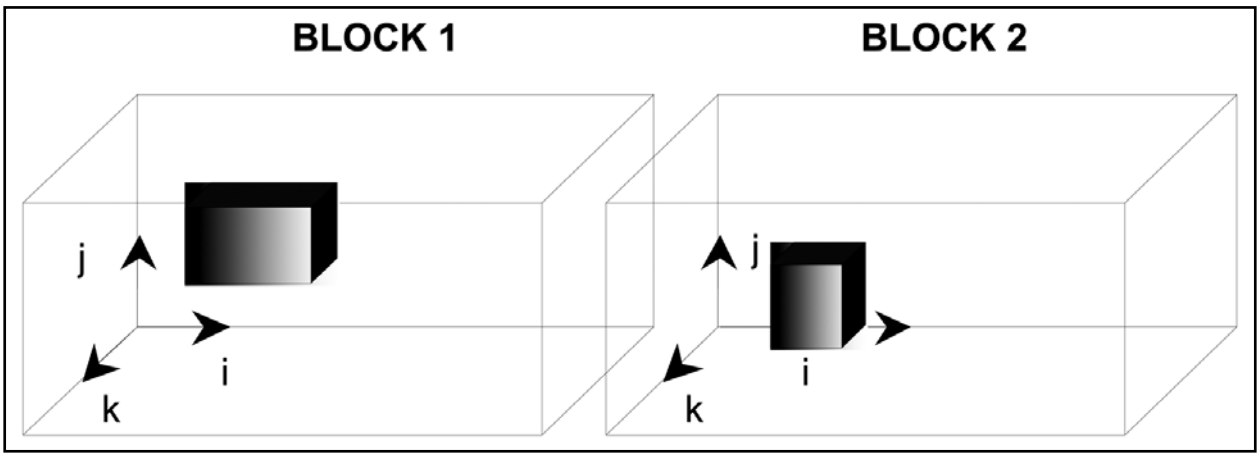


Figure 3. Two contiguous grid blocks with cutouts representing solid obstacles.

Finite-volume discretization

As a result of the unit grid spacing in computational space, the volume of each grid cell in Cartesian space is equal to the value of the Jacobian J for the coordinate transformation, which is given by Equation 2.28. For each cell face, one can construct an area vector $\delta\vec{s}_m$ normal to that face. This vector is the product of the (scalar) cell-face area with a unit vector normal to the face, and the subscripts m = 1, 2, 3 correspond to the ξ-, η-, and ζ-faces, respectively. The Cartesian components of the area vectors are δs_{mn}, in which the subscripts n = 1, 2, 3 correspond to the x-, y-, and z-directions, respectively. Using condensed notation, the definition for these components is

$$\delta s_{mn} = J \frac{\partial \xi_m}{\partial x_n} \tag{3.20}$$

where $(\xi_1, \xi_2, \xi_3) \equiv (\xi, \eta, \zeta)$ and $(x_1, x_2, x_3) \equiv (x, y, z)$.

For example, the x-component for area vector $\delta\vec{s}_1$ (normal to a face of constant ξ) is

$$\delta s_{11} = J \frac{\partial \xi}{\partial x} = y_\eta z_\zeta - y_\zeta z_\eta \tag{3.21}$$

Scalar quantities (pressure, temperature, density, etc.) are formally evaluated at the cell centers, but their values actually represent cell-averaged values. Vector fluxes (mass flux, momentum flux, etc.) represent face-integrated quantities evaluated on each cell face. Subject to the finite-

volume discretization outlined above, the components of volumetric flux normal to faces of constant ξ, η, ζ, respectively, are

$$\tilde{u} = \delta s_{11}\, u + \delta s_{12}\, v + \delta s_{13}\, w \tag{3.22}$$

$$\tilde{v} = \delta s_{21}\, u + \delta s_{22}\, v + \delta s_{23}\, w \tag{3.23}$$

$$\tilde{w} = \delta s_{31}\, u + \delta s_{32}\, v + \delta s_{33}\, w \tag{3.24}$$

Note that Equations 3.22, 3.23, and 3.24 are equivalent to Equations 2.30, 2.31, and 2.32, respectively. With $\vec{u}_\varphi \equiv 0$, the curvilinear (finite-volume) form of the transport equation (Equation 2.5) now becomes

$$J\frac{\partial \varphi}{\partial t} + J\, div\left(\vec{u}\,\varphi\right) = J\, div\left(D_\varphi\, grad\,\varphi\right) + J\, S\left(\varphi\right) \tag{3.25}$$

The curvilinear form for the advective term is

$$J\, div\left(\vec{u}\,\varphi\right) = \frac{\partial}{\partial \xi}\left(\tilde{u}\,\varphi\right) + \frac{\partial}{\partial \eta}\left(\tilde{v}\,\varphi\right) + \frac{\partial}{\partial \zeta}\left(\tilde{w}\,\varphi\right) \tag{3.26}$$

For a grid cell centered at $(\xi, \eta, \zeta) \equiv (i, j, k)$, with unit volume in computational space, the evaluation of Equation 3.26 is straightforward; i.e.,

$$\frac{\partial}{\partial \xi}\left(\tilde{u}\,\varphi\right) = \left(\tilde{u}\,\varphi\right)_{i+1/2} - \left(\tilde{u}\,\varphi\right)_{i-1/2} \tag{3.27}$$

$$\frac{\partial}{\partial \eta}\left(\tilde{v}\,\varphi\right) = \left(\tilde{v}\,\varphi\right)_{j+1/2} - \left(\tilde{v}\,\varphi\right)_{j-1/2} \tag{3.28}$$

$$\frac{\partial}{\partial \zeta}\left(\tilde{w}\,\varphi\right) = \left(\tilde{w}\,\varphi\right)_{k+1/2} - \left(\tilde{w}\,\varphi\right)_{k-1/2} \tag{3.29}$$

The diffusion term in Equation 3.25 is evaluated in expanded form; i.e.,

$$J\, div\left(D_\varphi\, grad\,\varphi\right) = J\, grad\, D_\varphi \bullet grad\,\varphi + J\, D_\varphi\, div\, grad\,\varphi \tag{3.30}$$

The calculation of the dot product in Equation 3.30 is straightforward, using Equations 2.15 to 2.17. The second term is evaluated as follows:

$$J D_{\varphi}\, div\left(grad\,\varphi\right) = D_{\varphi}\left(\frac{\partial \tilde{\varphi}_1}{\partial \xi} + \frac{\partial \tilde{\varphi}_2}{\partial \eta} + \frac{\partial \tilde{\varphi}_3}{\partial \zeta}\right) \tag{3.31}$$

where

$$\tilde{\varphi}_1 = \delta s_{11}\,\varphi_x + \delta s_{12}\,\varphi_y + \delta s_{13}\,\varphi_z \tag{3.32}$$

$$\tilde{\varphi}_2 = \delta s_{21}\,\varphi_x + \delta s_{22}\,\varphi_y + \delta s_{23}\,\varphi_z \tag{3.33}$$

$$\tilde{\varphi}_3 = \delta s_{31}\,\varphi_x + \delta s_{32}\,\varphi_y + \delta s_{33}\,\varphi_z \tag{3.34}$$

and Equations 2.15 to 2.17 are used to compute the *x*-, *y*-, and *z*-derivatives in Equations 3.29 to 3.31. Bernard and Kapitza (1992) explain in detail the finite-difference approximations needed for the ξ-, η-, and ζ-derivatives that occur in *grad* ϕ via Equations 2.15 to 2.17 and Equations 3.32 to 3.34. Suffice it here to say that, in these equations, the computed derivatives must employ certain combinations of (interpolated) face-centered and cell-centered coordinates (*x, y, z*) and dependent variables ϕ to ensure that

$$curl\,grad\,\varphi = 0 \tag{3.35}$$

The foregoing comments and constraints concerning the *div* and *grad* operators apply wherever these operators occur in the governing equations outlined in Chapter 2.

4 Boundary Conditions

Solid boundaries

In the context of PAR3D flow simulation, any solid boundary that fully confines the flow is considered impermeable. It might be a wall, floor, or ceiling of a room in a building; or it might be a bottom or sidewall of a reservoir or an open channel. Nothing except heat can pass through a boundary of this kind, and the component of velocity normal to the boundary is identically zero. In computational (i, j, k) space, where (ξ, η, ζ) ≡ (i, j, k), the normal-velocity constraint is easily stated as

$\tilde{u} \equiv 0$ for i-boundaries (constant ξ, constant i)

$\tilde{v} \equiv 0$ for j-boundaries (constant η, constant j)

$\tilde{w} \equiv 0$ for k-boundaries (constant ζ, constant k)

Subject to the constraint that nothing can pass through the boundary, the gradient flux of any transported quantity ϕ (other than heat) must also be zero normal to the boundary, such that

$$\vec{n} \bullet grad\, \varphi \equiv 0 \tag{4.1}$$

where $\vec{n}$ is a vector normal to the boundary, *as defined* by Equation 3.20. Note that the foregoing constraints apply to *slip* (frictionless) and *no-slip* (frictional) boundaries alike. In the case of heat flux, however, which is assumed to be proportional to the gradient of temperature T in the (solid) material adjacent to the boundary, Equation 4.1 gives way to

$$q = -C_T\, \vec{n} \bullet grad\, T \tag{4.2}$$

where q is the heat flux through the boundary (*into* the fluid) and C_T is the thermal conductivity for the material adjacent to the boundary.

In principle, a *slip* wall should impose no shear stress on the flow adjacent to the wall. In PAR3D this condition is idealized with the tangential-velocity constraint,

$$\vec{n} \bullet grad\, \upsilon \equiv 0 \tag{4.3}$$

where υ represents here the component of velocity tangent to the wall. For a *no-slip* wall adjacent to laminar flow in particular, PAR3D approximates the shear stress τ_w tangent to the wall as

$$\tau_w \approx \frac{\mu_0 \vec{n}}{|\vec{n}|} \bullet grad\, \upsilon \tag{4.4}$$

Subject to the no-slip condition, which imposes zero velocity at the wall, Equation 4.4 gives way to the first-order discrete approximation,

$$\tau_w \approx \frac{2\mu_0 \upsilon_1}{\delta_1} \tag{4.5}$$

where

μ_0 = molecular viscosity
υ_1 = average tangential velocity in grid cell adjacent to wall
δ_1 = thickness of adjacent grid cell normal to wall

For a *no-slip* wall adjacent to turbulent flow, the shear stress at the wall is proportional to the square of the friction velocity; i.e.,

$$\tau_w = \rho_0 \upsilon_*^2 \tag{4.6}$$

For "hydraulically rough" walls, the friction velocity υ_* can be computed directly from the roughness height δ_* of the wall surface material. PAR3D uses the following profile for υ near the wall, based on experimental data cited by Kuethe and Schetzer (1967):

$$\frac{\upsilon}{\upsilon_*} \approx \delta_+ \quad \text{for} \quad \delta_+ \le 5.25 \tag{4.7a}$$

$$\frac{\upsilon}{\upsilon_*} \approx \frac{1}{\kappa} + 0.524\,\delta_+ \quad \text{for} \quad 5.25 < \delta_+ < 20 \tag{4.7b}$$

$$\frac{\upsilon}{\upsilon_*} \approx \frac{1}{\kappa} \ln\left(9\,\delta_+\right) \quad \text{for} \quad \delta_+ \ge 20 \tag{4.7c}$$

in which von Karman's constant is taken to be $\kappa = 0.4$.

For rough walls, the non-dimensional distance δ_+ is obtained from the roughness height δ_* and the normal distance δ from the wall, such that

$$\delta_+ = \frac{10\,\delta}{3\,\delta_*} \tag{4.8}$$

For "hydraulically smooth" walls, however, the friction velocity v_* must be computed from Equation 4.7 (a, b, and c) using molecular viscosity μ_0 with an alternate definition for δ_+; i.e.,

$$\delta_+ = \frac{\rho_0\, v_*\, \delta}{\mu_0} \tag{4.9}$$

Also needed for turbulent flow are boundary conditions for the turbulence energy k and its dissipation rate ε. As discussed by Rodi (1980), near-wall values for these variables (adjacent to no-slip boundaries) are commonly obtained from the "wall" functions,

$$k = \frac{v_*^2}{C_\nu^{1/2}} \quad \text{for} \quad \delta \le \delta_1 \tag{4.10}$$

$$\varepsilon = \frac{v_*^3}{\kappa\,\delta_1} \quad \text{for} \quad \delta \approx \delta_1 \tag{4.11}$$

Equations 4.10 and 4.11 do not represent true boundary conditions because they are not imposed at the wall itself. They might better be described as off-the-wall surrogates for near-wall modeling with authentic (properly formulated) boundary conditions. In principle, they are applicable only for fully developed, unidirectional shear flow without a pressure gradient; but they are typically employed without regard for flow geometry or pressure variation. When used with coarse grids, which cannot resolve the near-wall velocity profile, Equations 4.10 and 4.11 may even contaminate the numerical solution with inappropriate values for k and ε in the flow field away from the wall.

Instead of wall functions imposed *in* the flow adjacent to no-slip boundaries, PAR3D uses *boundary* conditions that reproduce the off-the-

wall values given by Equations 4.10 and 4.11, but *only* for the circumstances under which they are valid. Specifically, in the wall-adjacent grid cells, the near-wall turbulence production rate P is approximated as

$$P \approx P_0 + \frac{3}{2}\frac{\upsilon_*^3}{\kappa \delta_1} \tag{4.12}$$

in which P_0 is the production rate otherwise given by Equation 2.11. The flux components for k and ε normal to the wall are respectively approximated as

$$\frac{\nu_T \vec{n}}{\sigma_k} \bullet grad\, k \approx \frac{\kappa \upsilon_*}{C_\nu^{1/2} \sigma_k} |\vec{n}| (k - k_*) \tag{4.13}$$

$$\frac{\nu_T \vec{n}}{\sigma_\varepsilon} \bullet grad\, \varepsilon \approx \frac{\kappa \upsilon_*}{C_\nu^{1/2} \sigma_\varepsilon} |\vec{n}| (\varepsilon - \varepsilon_*) \tag{4.14}$$

where

$$\kappa = 0.40$$
$$C_\upsilon = 0.09$$
$$\sigma_k = 1.0$$
$$\sigma_\varepsilon = 1.3$$
$$k_* = \frac{\upsilon_*^2}{C_\nu^{1/2}}$$
$$\varepsilon_* = \frac{7\upsilon_*^3}{\kappa \delta_1}$$

The near-wall production rate and the wall-normal flux components for k and ε were developed in an unpublished supplementary effort, which involved the 1-D numerical solution of the RANS momentum equation (Equation 2.1) and the turbulence model equations (Equations 2.7 and 2.8) for unidirectional shear flow without a pressure gradient, parallel to a no-slip wall. These developments are discussed in detail in the appendix.

When supplemented with Equations 4.6, 4.7, and 4.12 to 4.14, the numerical solutions for Equations 2.1, 2.7, and 2.8 exhibit decreasing wall influence with increasing wall-normal grid spacing δ_1. Moreover, as υ_* approaches zero, Equations 4.13 and 4.14 reduce to null conditions for the wall-normal *flux* of k and ε. This effect is very different from that of

Equations 4.10 and 4.11, which impose null *values* for k and ε as v_* approaches zero.

PAR3D designates all grid-block interfaces and internal cutout surfaces (Figure 3) as *slip* boundaries by default. Model users may designate these boundaries individually as slip or no-slip, and they may also change the default to no-slip, via the PAR3D input files. To facilitate coarse-grid simulations for complex enclosures with many internal partitions, the code also allows *any* grid surface to be designated as an impermeable slip boundary. In addition, the grid-block interfaces may be individually designated as fully permeable connecting surfaces (called "grid cuts") between contiguous flow regions.

Inflow and outflow boundaries

Boundaries through which flow may enter or leave the computational grid are designated either as flux boundaries or as open boundaries. These may consist of groups of one or more cell faces on any one of the six bounding surfaces for a particular grid block or internal cutout (Figure 3). On a *flux* boundary, the total rate of flow and the distribution of flow are fixed for the duration of the simulated time-interval. On an *open* boundary, the total rate of flow is fixed, but the distribution of flow varies with conditions just upstream of the boundary in question.

Phantom grid cells that lie *outside* the grid or *inside* internal cutouts are called "out" cells. Every grid block is completely surrounded by *out* cells that may contain *either* updated values for adjacent variables in neighboring grid blocks *or* updated values for variables entering the grid via local inflow. When *flux* faces reside on the surface of a cutout, the adjacent *out* cells (inside the cutout) may likewise contain updated values for variables entering the grid from that location.

At designated *flux* boundaries, the velocity component normal to the boundary may be specified for each flux face, along with the three Cartesian velocity components for the *out* cell just upstream. In addition, fixed inflow values may be specified for all the other transported variables in the same out cell. For suspended materials (sediments or contaminants) the concentration in the out cell may correspond to a previous (or current) concentration at some other location in the flow. This is discussed later in the section on reentrant boundaries.

Open boundaries are subject to an extrapolation procedure developed by Orlanski (1976) to eliminate spurious reflections in situations (such as buoyant flow) that may generate internal waves. For normal components of velocity, required *on* an open boundary, and for cell-centered variables, required in an *out* cell just outside an open boundary, the Orlanski procedure replaces the governing equation for the dependent variable with

$$\frac{\partial \varphi}{\partial t} + c\frac{\partial \varphi}{\partial \eta} = 0 \tag{4.15}$$

where

c = effective "wave" speed for dependent variable ϕ
η = coordinate direction normal to boundary in computational space

Equation 4.15 is first rearranged to obtain an expression for the wave speed; i.e.,

$$c = -\frac{\varphi_t}{\varphi_\eta} \tag{4.16}$$

Using superscript n to indicate time level, and subscript j to indicate η-position, the value of c is then approximated (for time level $n - 1$ and position j) as

$$c_j^{n-1} \approx -\frac{\Delta \eta}{\Delta t}\frac{\varphi_j^n - \varphi_j^{n-1}}{\varphi_j^{n-1} - \varphi_{j-1}^{n-1}} \tag{4.17}$$

According to Equation 4.15, for an *out* cell at position $j + 1$ with an *open* boundary at position $j + 1/2$, the value of ϕ at time level $n + 1$ can be approximated as

$$\varphi_{j+1}^{n+1} \approx \varphi_{j+1}^n - c_j^{n-1}\frac{\Delta t}{\Delta \eta}\left(\varphi_{j+1}^n - \varphi_j^n\right) \tag{4.18}$$

In the PAR3D version of the Orlanski procedure, however, this result is replaced with

$$\varphi_{j+1}^{n+1} \approx \varphi_j^n \quad \text{for} \quad c_j^{n-1} > 0 \tag{4.19a}$$

$$\varphi_{j+1}^{n+1} \approx \varphi_{j+1}^{n} \quad \text{for} \quad c_j^{n-1} \le 0 \tag{4.19b}$$

For an *out* cell at position $j - 1$ with an *open* boundary at position $j - 1/2$, the same procedure gives

$$\varphi_{j-1}^{n+1} \approx \varphi_{j}^{n} \quad \text{for} \quad c_{j+1}^{n-1} < 0 \tag{4.20a}$$

$$\varphi_{j-1}^{n+1} \approx \varphi_{j-1}^{n} \quad \text{for} \quad c_{j+1}^{n-1} \ge 0 \tag{4.20b}$$

Equations 4.19 and 4.20 ensure that outward propagating disturbances exit the grid, without reflection, through the designated *open* boundaries.

When used for cell-centered variables, this extrapolation procedure is straightforward, and it can be applied without further adjustment. When used for the face-centered flux components ($\tilde{u}, \tilde{v}, \tilde{w}$) defined by Equations 3.22 to 3.24, however, an additional step is necessary. Specifically, after the flux extrapolation has been completed, the resulting (total) flow rate is computed for the contiguous cell faces that define each (separately designated) *open* boundary. This flow rate may be different from the specified (fixed) flow rate for that boundary, and the ratio of fixed rate to extrapolated rate is used as a correction factor for the individual fluxes. Thus, the distribution of flow may change with time, but the rate of flow remains constant for the open boundary in question.

Note that the *flux* designation can be used for inflow and outflow boundaries alike, but the *open* designation is generally limited in practice to outflow boundaries. If an open designation *is* to be used successfully for inflow, the boundary in question should lie far upstream of any perturbing obstacles in the flow.

Reentrant boundaries

Inflow boundaries with a *flux* designation may also serve as reentrant boundaries for suspended materials (sediments or contaminants). *Out* cells adjacent to an inflow may be referred to a single (user-designated) grid cell, from which a current or previous material concentration is extracted to serve as the updated out-cell concentration. The aim of this option is to facilitate the modeling of transport in closed systems that recycle suspended materials through networks of pipes or ducts. The reference (intake) grid cell might lie next to an intake for a pump or an air

handler, where a record of the concentration is kept from one time-step to the next. To account for travel time through connecting pipes or ducts, the model user can activate a *delay* option, which computes return time t_R using the return velocity v_R and the return distance l_R from the intake cell; i.e.,

$$t_R = \frac{l_R}{v_R} \tag{4.21}$$

The return velocity v_R is assumed to be the local inflow velocity, and the return distance l_R is taken to be the *sum* of the Cartesian return-distance components (x_R, y_R, z_R) from the intake to the out cell, such that

$$l_R = x_R + y_R + z_R \tag{4.22}$$

Equation 4.22 is intended to approximate return distances associated with connecting pipes and ducts that follow straight (x, y, z) lines except when changing direction via right-angle turns. Once the return time has been calculated, the out-cell concentration at time t is equated with the intake concentration at time $t - t_R$.

PAR3D also allows out cells to be referred individually to nearby intake cells from which the suspended-material concentration is transferred *without* accounting for travel time. The combination of idealized transfers, with and without delay, allows the code to model the role of pipes and ducts in closed systems without computing the internal details of the pipe or duct flow.

Air-water interfaces

For applications in deep water, the overhead boundary may be designated a *free* surface. At user-specified time-intervals, PAR3D displaces the free surface by an incremental distance $\Delta z = \hat{w}\,\Delta t$, where $\hat{w}$ is the z-component of the flow velocity *normal* to the surface. To accommodate this displacement, the z-coordinates of all the grid nodes are adjusted proportionately in the grid block adjacent to the surface. When the computed flow reaches steady state, the fluid velocity is tangent to the free surface ($\hat{w} = 0$) and there is no further displacement thereof.

The PAR3D treatment of free-surface deformation for water is decidedly quasi-static. Each incremental displacement of the surface represents an iteration toward a final (steady state) configuration in which the surface is perfectly aligned with the flow. The main limitation is that dynamic surface motion cannot be simulated in this manner, and standing waves are the only surface waves that can be accommodated.

The transported constituents that may cross a free surface are heat, dissolved gas, and undissolved gas (released from rising bubbles). The boundary condition for heat flux is the same as that given by Equation 4.2 for slip boundaries. The gases under consideration here are oxygen and nitrogen, and these are transported separately in distinct (dissolved and undissolved) phases. For dissolved gases in particular, the surface flux q into the water is given by

$$q = K_L \left|\vec{n}\right| \left(\chi_S - \chi \right) \tag{4.23}$$

where χ is the dissolved-gas concentration, and χ_s is the concentration at saturation. The transfer coefficient K_L is based on the concept of "penetration" as discussed by Azbel (1981) in which

$$K_L \approx \sqrt{D_0 \, \omega} \tag{4.24}$$

where

D_0 = molecular diffusivity of dissolved gas in water
ω = exchange frequency

The formula used for the free-surface transfer coefficient in PAR3D is equivalent to one developed by Fortesque and Pearson (1967),

$$K_L \approx 2.15 \sqrt{\frac{\varepsilon D_0}{k}} \tag{4.25}$$

with ω related to the turbulence energy k and dissipation rate ε by

$$\omega \approx 4.62 \frac{\varepsilon}{k} \tag{4.26}$$

Note that the gas-transfer equation (Equation 4.23) also applies at the surface of a bubble, albeit with a different penetration-based formula for the transfer coefficient. In this case, the exchange frequency is assumed to be

$$\omega \approx \frac{w_B}{\lambda} \tag{4.27}$$

where

w_B = bubble rise velocity, relative to water

The mean (local) separation distance between the centers of the rising bubbles is

$$\lambda = d_B \left(\frac{6\vartheta}{\pi} \right)^{-1/3} \tag{4.28}$$

where

d_B = effective bubble diameter
ϑ = local void ratio for bubbles

Accordingly, the formula for the bubble-surface transfer coefficient becomes

$$K_L \approx \left(\frac{D_0 w_B}{d_B} \right)^{\frac{1}{2}} \left(\frac{6\vartheta}{\pi} \right)^{\frac{1}{6}} \tag{4.29}$$

Note that the term "local void ratio" refers here to the ratio of bubble volume to fluid volume in each grid cell.

Solid obstacles

Solid obstacles may be incorporated in the grid as internal cutouts, as shown in Figure 3, or they may be included as empty regions of Cartesian (x, y, z) space surrounded by active grid blocks. Since each grid block bears its own computational (i, j, k) space, the latter option avoids the creation of unused (cutout) grid. The obstacle surfaces are *slip* (or *no-slip*) boundaries by default, as discussed previously; but they can be individually designated as *flux* boundaries or as *open* boundaries.

When obstacles are included inside grid blocks as cutouts, the obstacle surfaces conform to internal grid lines of the blocks in question. When included as empty regions between blocks, however, their surfaces are formed by the corresponding external surfaces of the bounding blocks. The latter option affords more geometric flexibility, because it allows the computational (i, j, k) axes of each bounding block to be oriented differently with respect to the fixed Cartesian (x, y, z) axes.

Grid-block interfaces

Interfaces between contiguous grid blocks (Figure 3) are treated as solid boundaries unless designated otherwise in the model input. When a grid is divided into multiple blocks to facilitate parallel processing, however, block interfaces may also represent connecting surfaces ("grid cuts") across which flow and transport are continued without hindrance or interruption.

Since input designations for grid cuts can be rather lengthy for grids with many blocks, PAR3D input (other than the grid itself) is divided into two input files. The primary input file contains everything *except* grid-cut designations, whereas the secondary input file contains *only* grid-cut designations. To reduce the drudgery associated with the preparation of grid-cut input by hand, a companion code (SPLICE) has been developed, which locates the connecting interfaces between grid blocks, and creates a (provisional) secondary input file in which all block interfaces are grid cuts by default. This file can be edited by the model user to eliminate individual interfaces (or portions thereof) that are *not* intended as grid cuts. A second companion code (GINPUT) has been developed that creates primary input for other boundary types from user-specified surface coordinates.

Sources and sinks

PAR3D allows user-selected groups of grid cells to be designated as sources from which (or into which) heat, undissolved gas, and suspended material may emanate (or vanish). These sources (or sinks) operate for the duration of the simulated time-interval, after which they can be deactivated or moved to another location for a subsequent (hot start) time-interval. In the case of undissolved gas (bubbles) in water, a free surface also acts as a sink, removing the gas *from* the water at the rate that it arrives *at* the surface.

5 Model Operation

Cold start

Initial (mass-conserving) flow is computed by PAR3D from user-specified velocity data via Equations 3.18 and 3.19. Ordinarily these velocity data are specified exclusively along inflow and outflow boundaries, in which case the computed initial flow is potential flow. Any user-specified velocity field is admissible, however, and the resulting initial flow preserves inherent vorticity while eliminating continuity violations.

Starting with the initial flow posed at time zero, PAR3D uses Equations 2.1, 2.2, 2.7, and 2.8 (in discrete form) to advance the developing flow, through time, toward steady state. If no steady state exists for the situation under consideration, the computed flow may reach a periodic or quasi-periodic condition (e.g., vortex shedding) that persists forever.

For passive constituents emanating from a fixed source, with null concentrations initially, transport via Equation 2.5 may be delayed until the source itself is activated. Likewise, for constituents introduced through inflow boundaries, transport may be delayed until the initiation of introduction.

Hot start

Flow simulations are rarely completed in a single run. They are usually executed in a sequence of runs, called hot starts, which continue any flow development that was left incomplete by the previous run.

Before executing a hot start, PAR3D reads an output file from the previous run, which contains current values for the grid coordinates (which may be different from the initial grid) and all the dependent variables. It then reads the primary and secondary input files, which may direct the code to continue execution with the same boundary conditions or with new boundary conditions. The latter option affords users the freedom to change boundary conditions (and source terms) at will from one hot start to the next.

Source code

PAR3D is written in FORTRAN 77 with the MPI Library included during compilation. MPI commands embedded in the source code mediate the transfer of shared information between processors. Thus, PAR3D can be executed only on computers equipped with FORTRAN compilers and MPI libraries. User-prepared input is specified in NAMELIST format, and grids are accepted in PLOT3D format.

6 Applications

Previous applications

The present work is the first ERDC report documenting PAR3D in particular, but the model has been used since 1999 in a number of research efforts. These include studies involving secondary flow in curved channels (Bernard et al. 1999), buoyant flow and gas transfer induced by bubble plumes (Bernard 2003), and prevention of salinity intrusion in the Panama Canal (LaBonte and Spani 2005). Since past work has been concerned with flow and transport exclusively in water, the applications presented here deal with flow and transport in air.

Air flow and aerosol dispersion in a simple enclosure

Shimada et al. (1996) have conducted experiments concerning the forced-air transport of fine particles in a ventilated room. The room under consideration (Figure 4) was designed to simulate flow patterns inside a train car, with a supply vent running the length of the ceiling, and a return vent in one of the walls. The nominal room dimensions were 270 cm in the x-direction, 330 cm in the y-direction, and 200 cm in the z-direction. The supply vent was a slot, 5 cm wide, in the elliptical supply duct; and the return vent was a square opening, 40 cm across, centered at x = 135 cm, y = 0, and z = 35 cm. The flow rate was 257 L/sec, with the inflow velocity profile shown in Figure 4.

Figure 5 shows the bounding surfaces for the grid used in the PAR3D representation of the ventilated room. Figures 6 and 7 show the central xz-plane and the central yz-plane, respectively, for the same grid. The grid spacing ranged from 2.5 to 5.0 cm, and the grid was divided into four blocks with cutting planes perpendicular to the y-axis. Figures 8 and 9 depict computed (steady-state) velocity vectors for the central xz-plane and the central yz-plane, respectively. Figures 10, 11, and 12 show comparisons of computed and measured z-components of velocity in the central xz-plane.

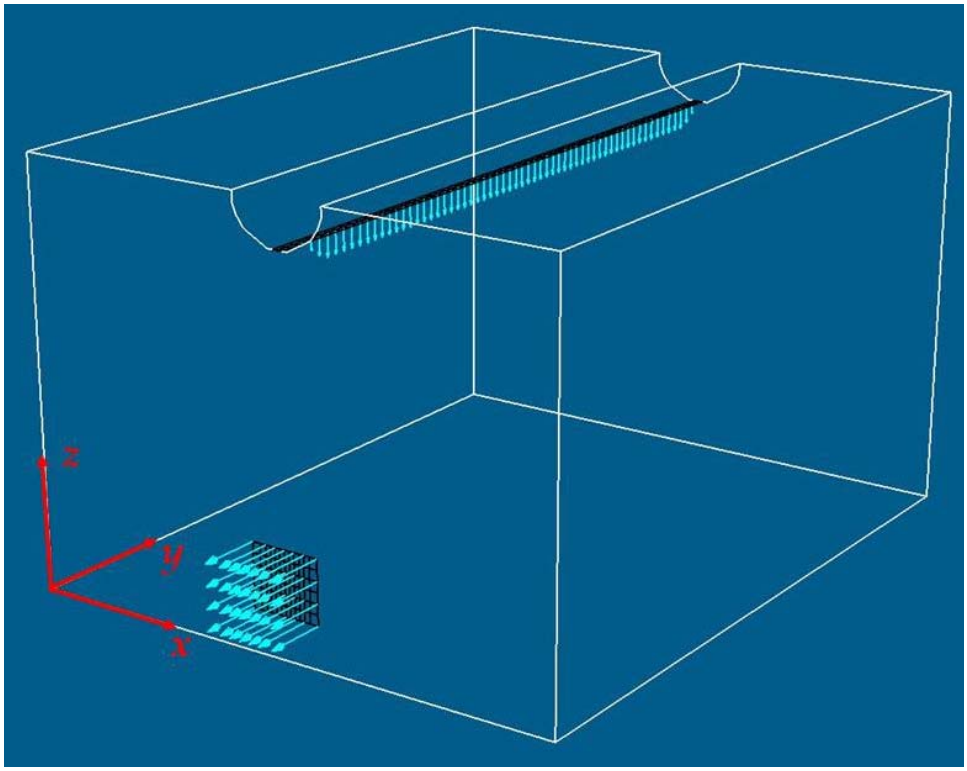


Figure 4. Outline of ventilated room used in experiments conducted by Shimada et al. (1996) showing overhead supply vent and sidewall return vent.

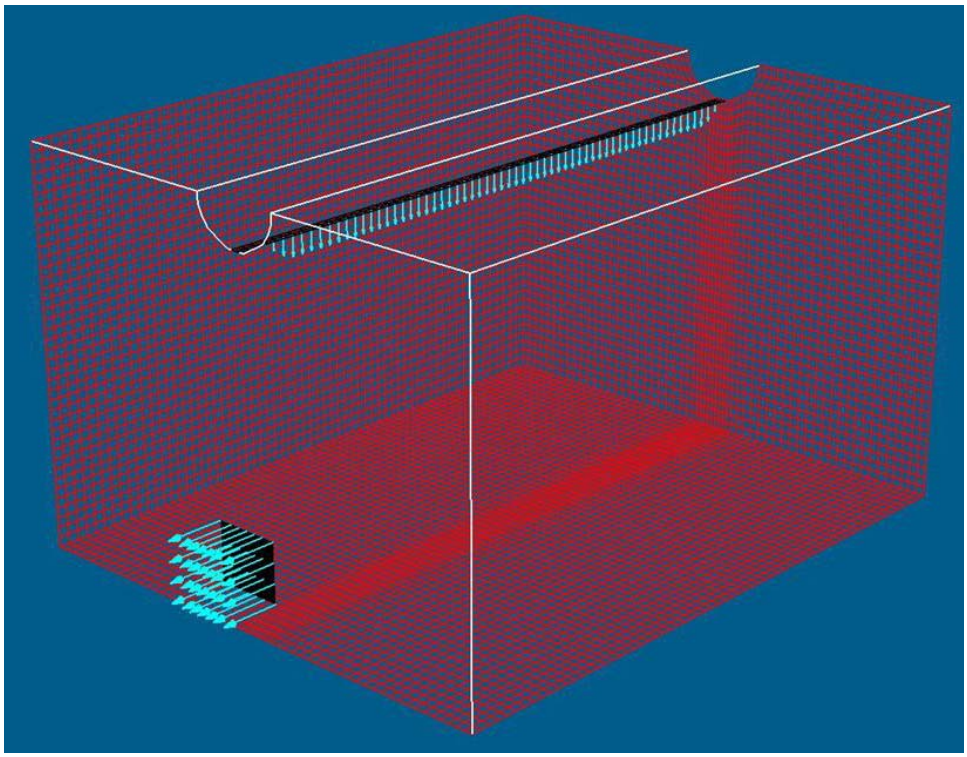


Figure 5. Bounding grid surfaces used in PAR3D representation of ventilated room.

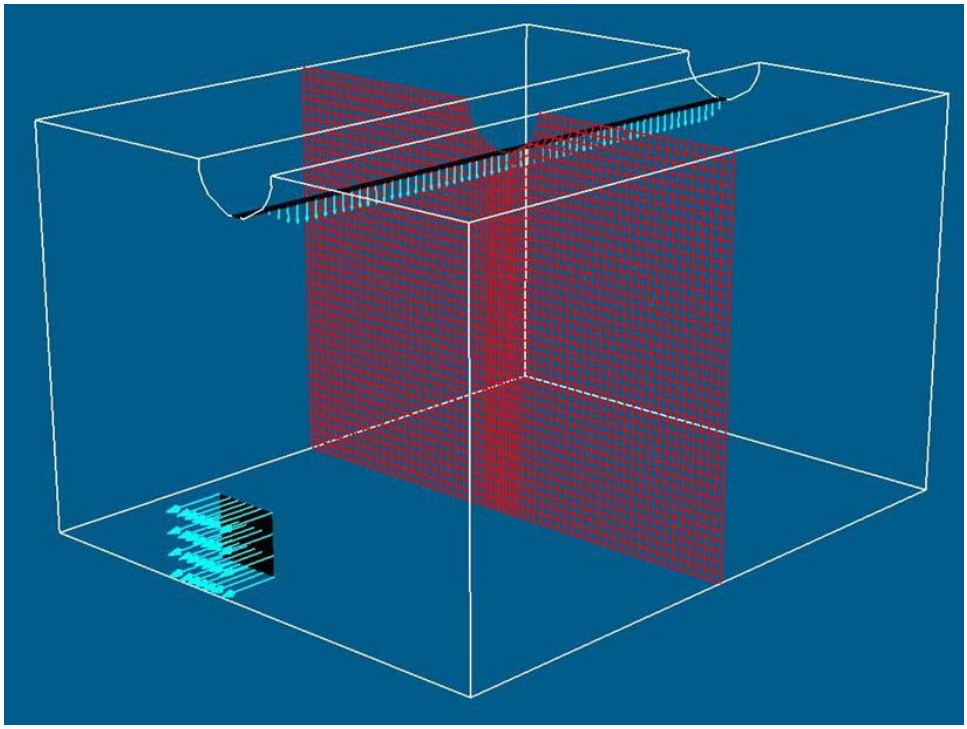


Figure 6. Central *xz*-plane for PAR3D representation of ventilated room.

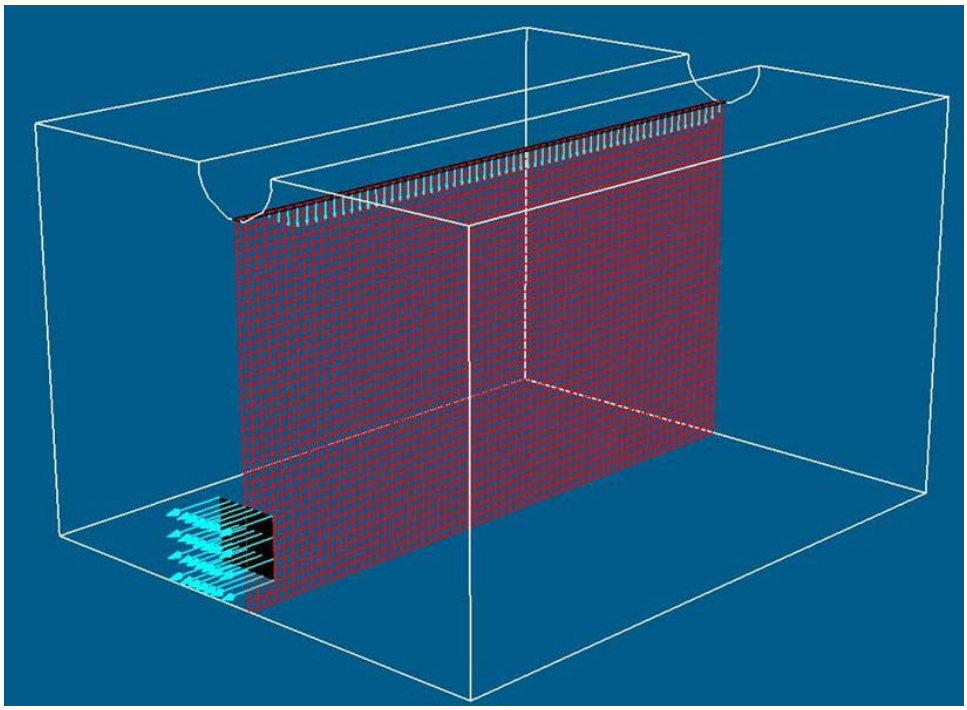


Figure 7. Central *yz*-plane for PAR3D representation of ventilated room.

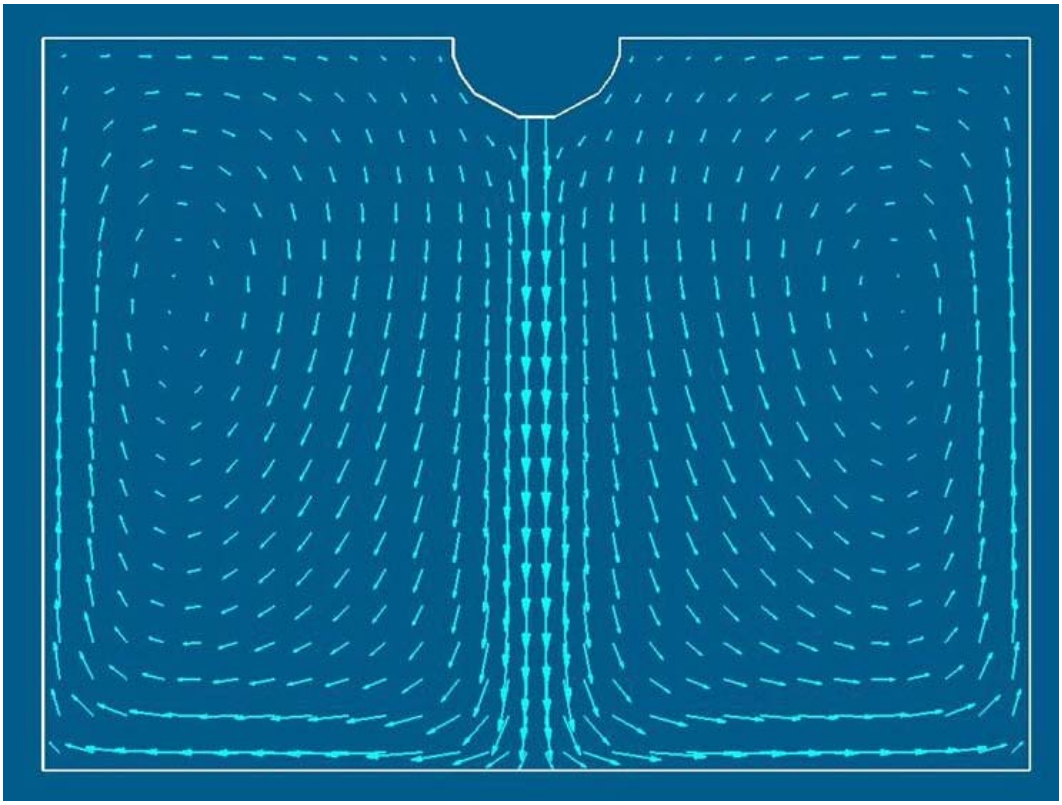


Figure 8. Computed velocity vectors for central *xz*-plane of ventilated room.

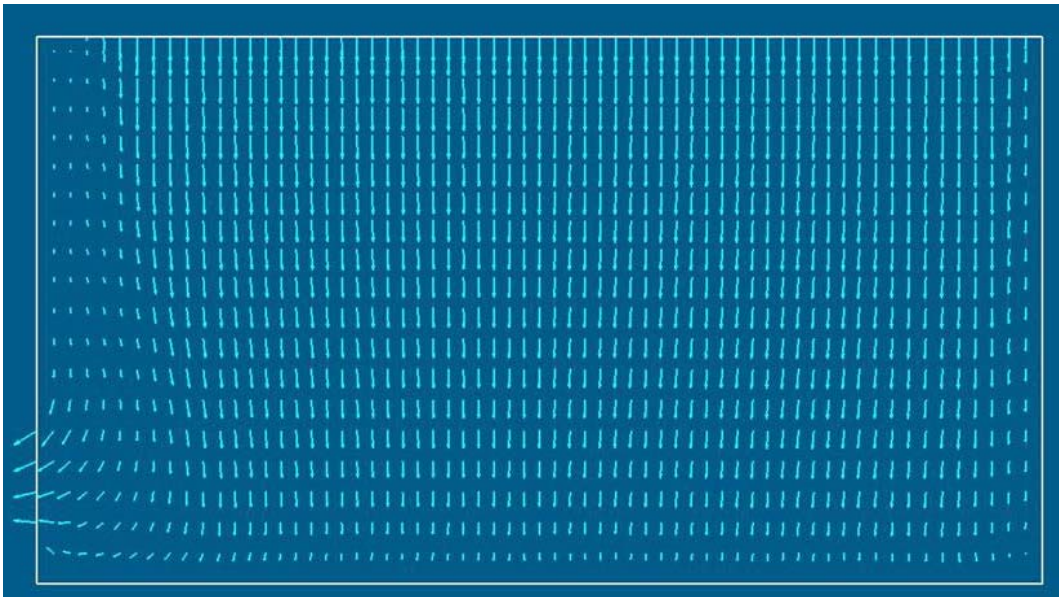


Figure 9. Computed velocity vectors for central *yz*-plane of ventilated room.

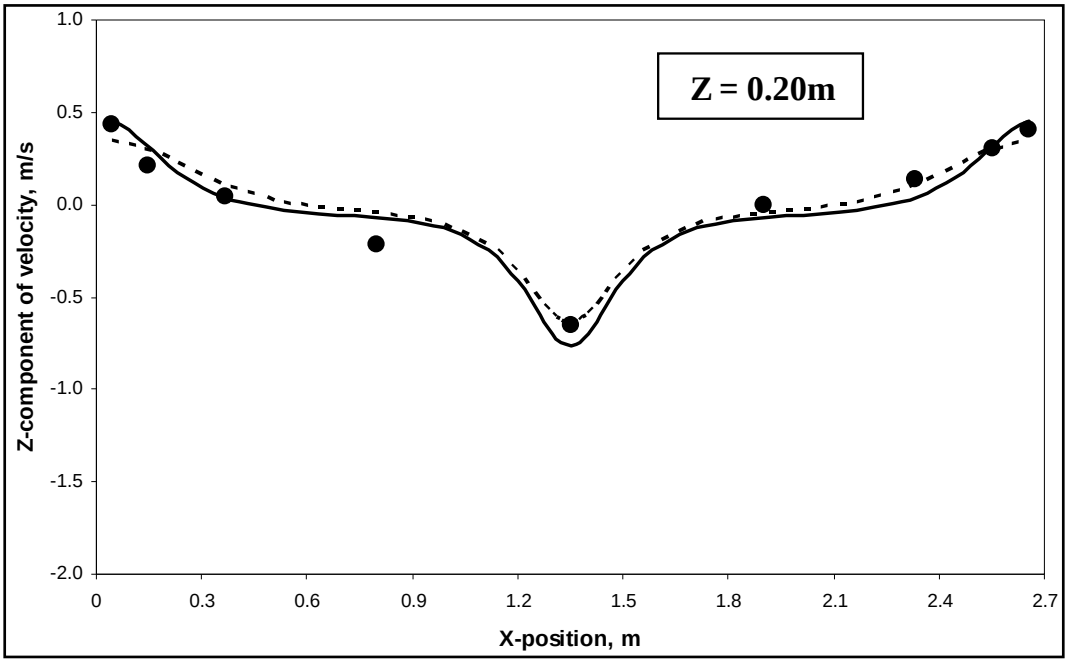


Figure 10. Comparison of experimental data for velocity (•) with PAR3D results computed at z = 20 cm in central *xz*-plane, using slip condition (- - - -) and no-slip condition (—).

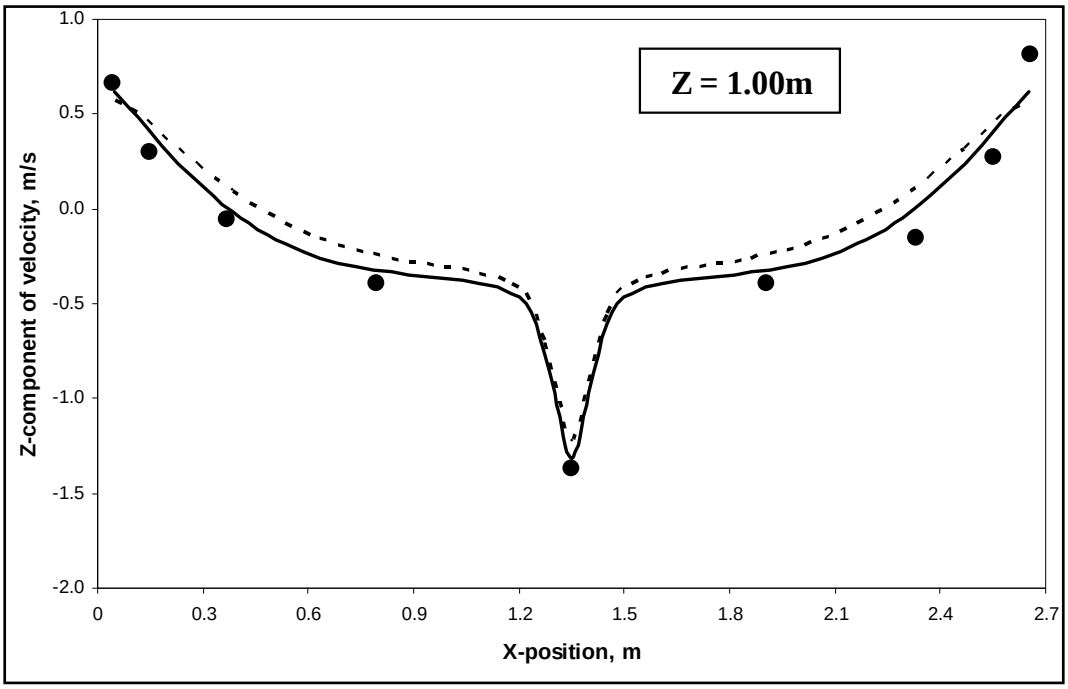


Figure 11. Comparison of experimental data for velocity (•) with PAR3D results computed at z = 100 cm in central *xz*-plane, using slip condition (- - - -) and no-slip condition (—).

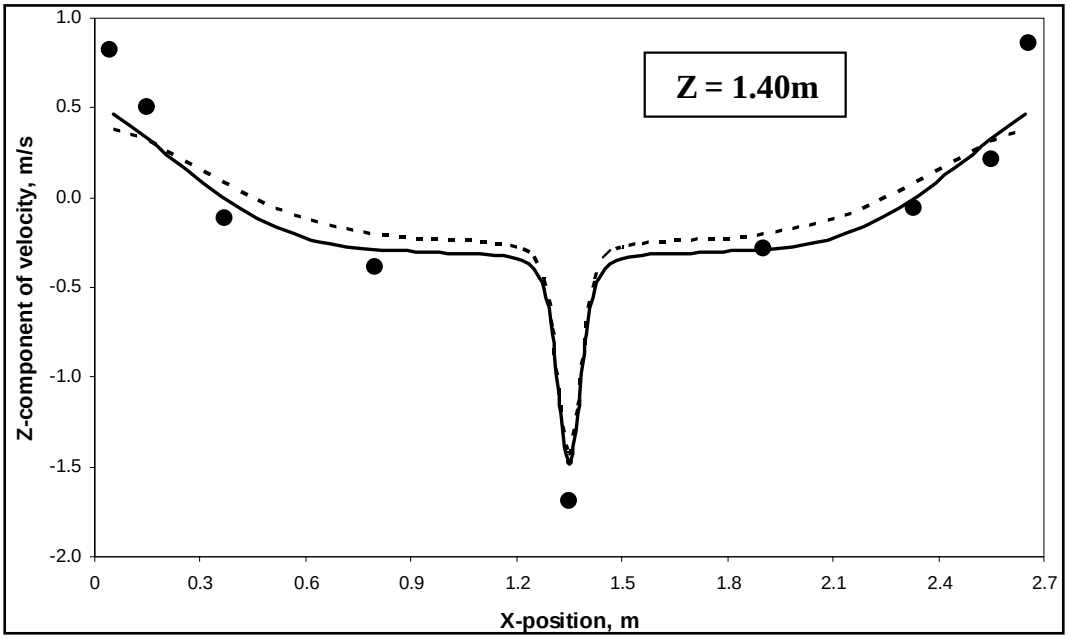


Figure 12. Comparison of experimental data for velocity (•) with PAR3D results computed at z = 140 cm in central *xz*-plane, using slip condition (- - -) and no-slip condition (—).

The PAR3D results shown in Figures 8 to 12 were obtained using small values for the turbulence energy (k = 1.0 cm²/sec²) and eddy viscosity (υ_T = 1.0 cm²/sec) entering the room through the supply vent. Increasing or decreasing these values by a factor of 10 had little effect on the computed flow, which ultimately reached steady state in all cases. The results shown for the no-slip condition were obtained by assuming the walls, floor, and ceiling to be hydraulically smooth. There is little difference between the slip results and the no-slip results, mainly because the influence of friction is unimportant at distances far from the boundaries.

In the particle-dispersion experiments, particle emitters were placed at the intersection of the central *xz*- and *yz*-planes, 50 cm above the floor (emitter A) and directly on the floor (emitter B), as indicated in Figure 13. In each experiment, the air flow in the room was allowed to reach steady state before the particle emitter was activated, and the emitter was operated for 600 sec before being shut off. Particle counters, located 62 cm above the floor in the central *xz*-plane, indicated a sharp rise in particle concentrations during the first 100 sec or so. These concentrations leveled off within the next 100 sec, and they remained (roughly) constant until the emitter was shut off.

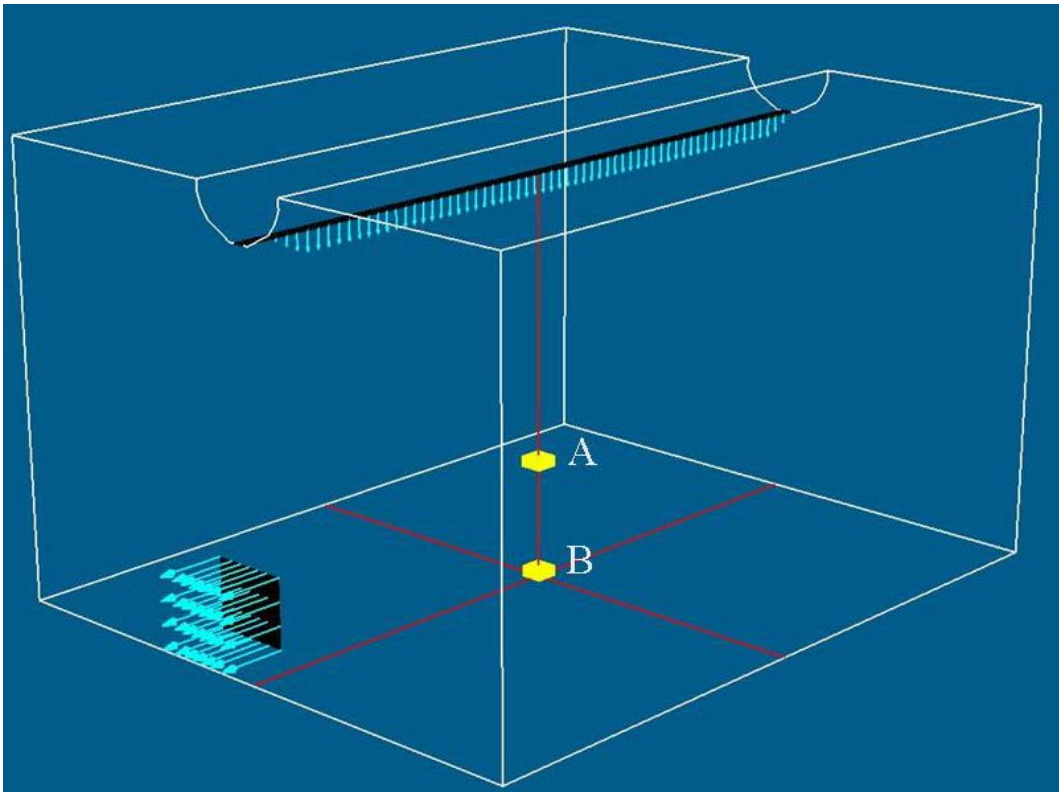


Figure 13. Emitter locations A and B for particle-dispersion experiments.

PAR3D simulations of the particle dispersion followed the same qualitative pattern, with concentrations increasing only slightly after the first 200 sec. Figures 14 to 17 show color maps of predicted steady-state concentrations in the central *xz*- and *yz*-planes; and Figures 18 and 19 show comparisons of predicted and measured concentration profiles for emitters A and B, respectively. The concentrations presented here are normalized values, as reported by Shimada et al. (1996), equivalent to a delivery rate of 0.0167 unit per sec at the emitter. The PAR3D predictions were made using a null settling velocity, and a Prandtl-Schmidt number of unity, in the transport equation (Equation 2.5).

PAR3D overpredicts observed concentrations for emitter A by 20 to 50 percent, and for emitter B by 100 to 300 percent. This is admittedly a gross over-prediction of the data, especially for emitter B; but it might be remedied somewhat by increasing the assumed value of the Prandtl-Schmidt number, and possibly by including a settling velocity in the transport equation for the particles. Shimada et al. assumed their particles to be neutrally buoyant, however, and they made no measurements of settling velocity in still air.

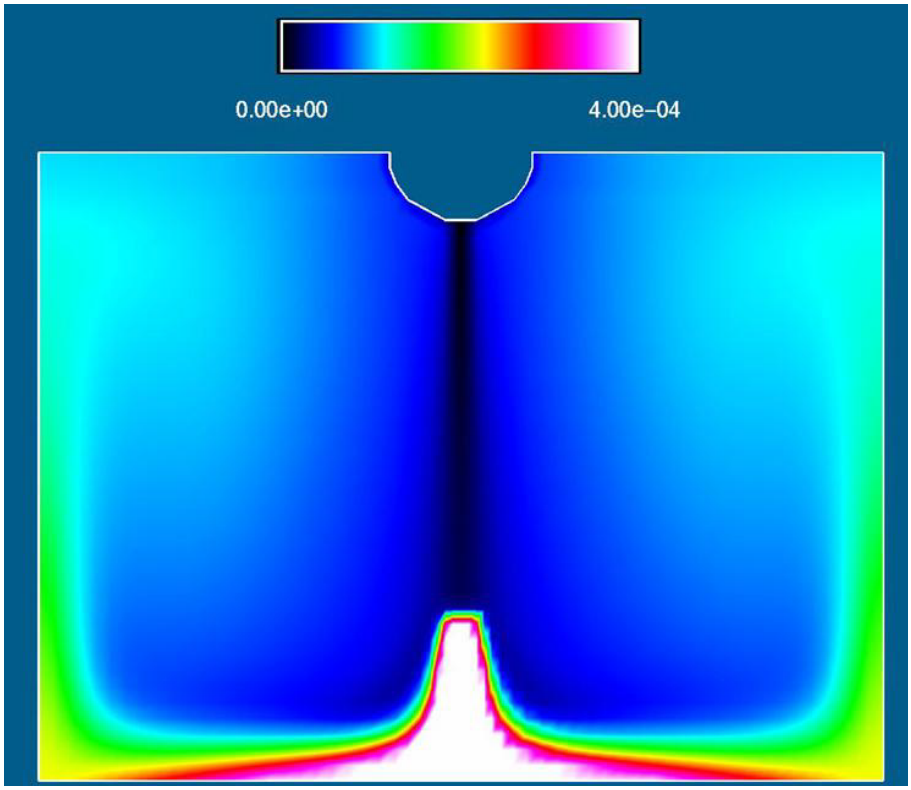


Figure 14. Steady-state particle concentration computed in central *xz*-plane for emitter A.

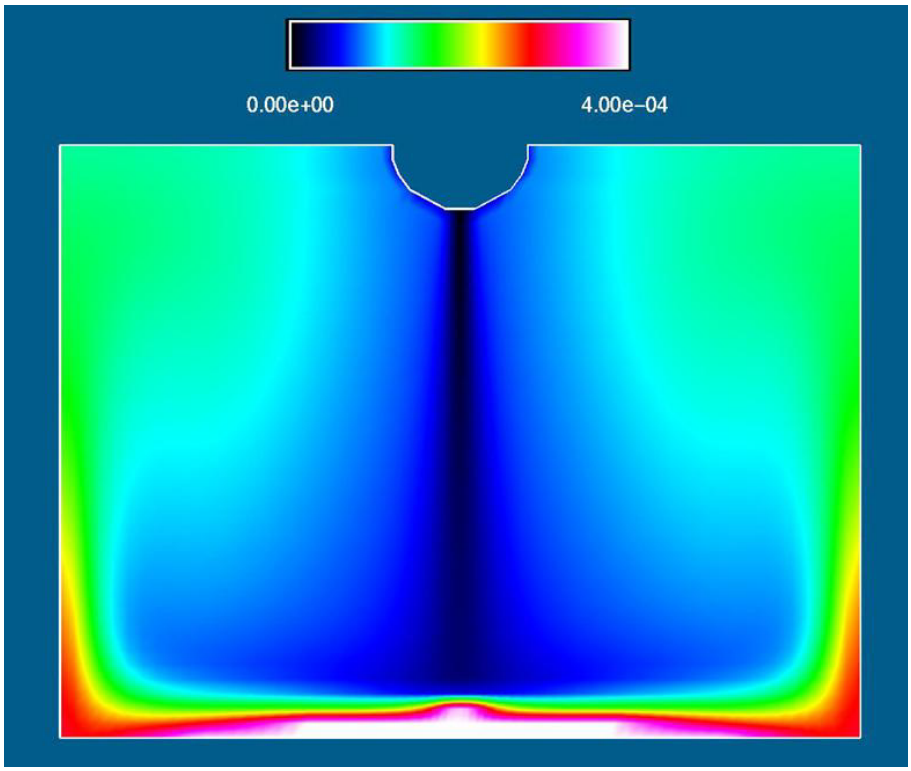


Figure 15. Steady-state particle concentration computed in central *xz*-plane for emitter B.

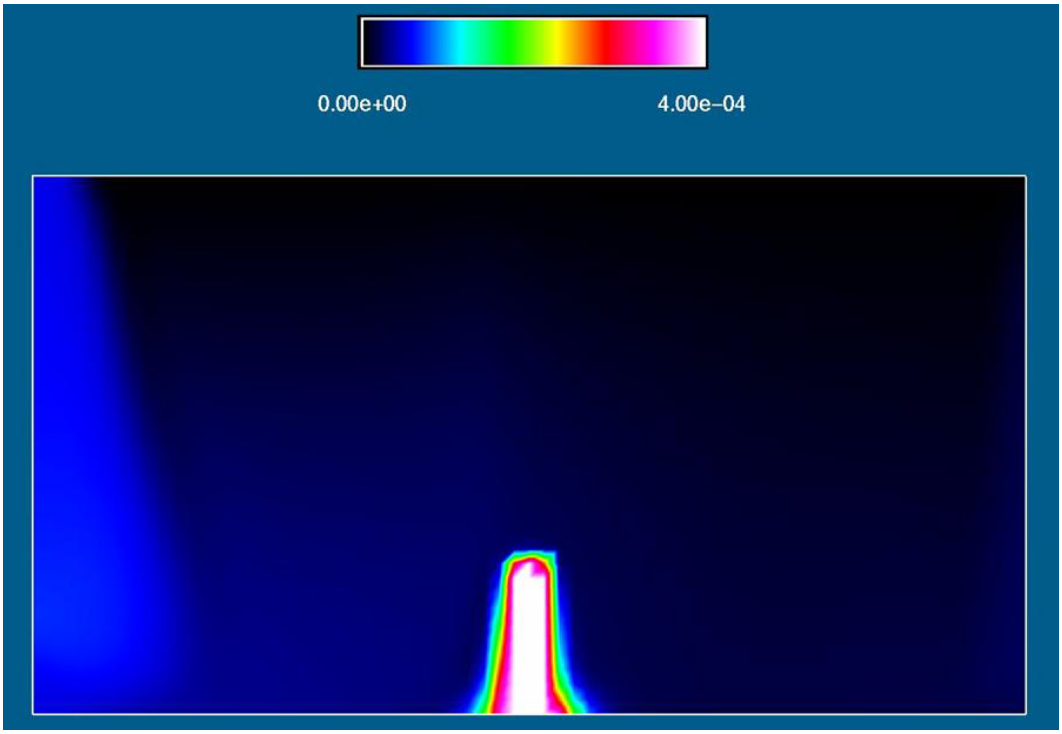


Figure 16. Steady-state particle concentration computed in central *yz*-plane for emitter A.

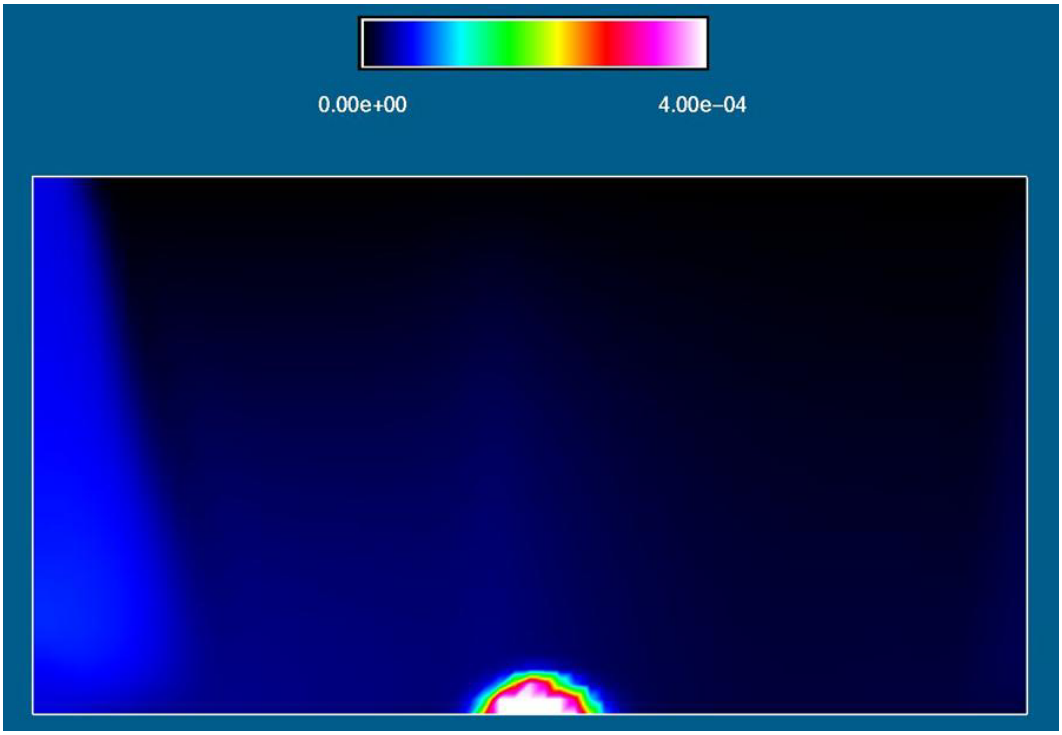


Figure 17. Steady-state particle concentration computed in central *yz*-plane for emitter B.

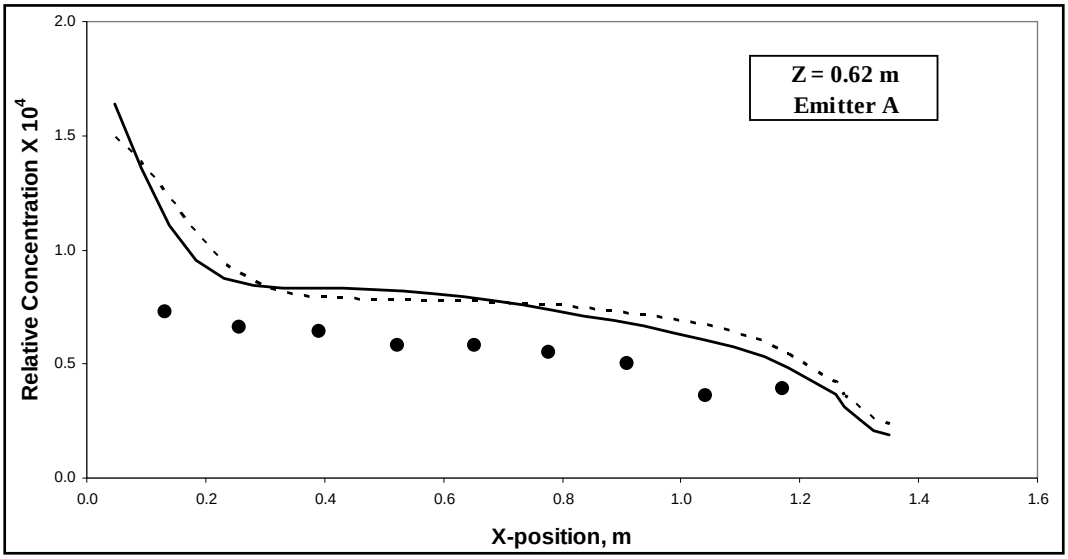


Figure 18. Comparison of experimental particle-concentration data (•) for emitter A with PAR3D results computed at z = 62 cm in the central *xz*-plane, using slip condition (- - - -) and no-slip condition (—).

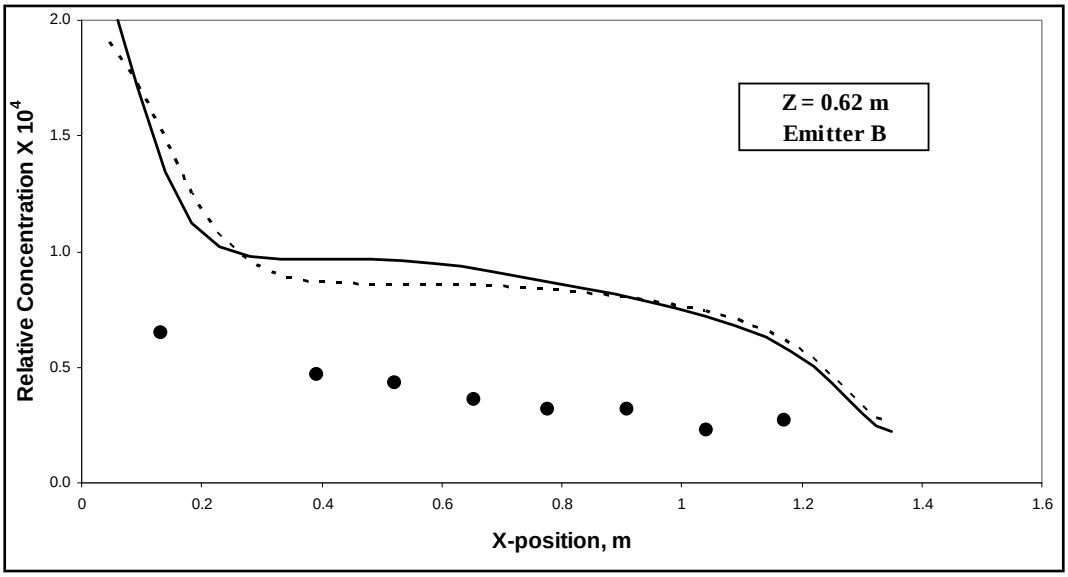


Figure 19. Comparison of experimental particle-concentration data (•) for emitter B with PAR3D results computed at z = 62 cm in the central *xz*-plane, using slip condition (- - - -) and no-slip condition (—).

Air flow and aerosol dispersion in a complex enclosure

Whicker et al. (2000) used CFD to determine effective locations for continuous air monitors (CAMs) in a ventilated workroom of the main plutonium facility at Los Alamos National Laboratory (LANL). The workroom under consideration (Figure 20) contained five rows of glove

boxes and an overhead trolley.[1] Ventilation was provided by four inlet air diffusers in the ceiling and four exhaust vents near the floor, as shown in Figure 21. The dimensions of the trolley and the glove boxes are shown in Figure 22. To validate their CFD code prior to the CAM placement study, Whicker et al. conducted experiments in which aerosol particles were released at one location and monitored at three other locations in the workroom. An overhead view of the release and detector stations is shown in Figure 23. The particle source (station 11) was placed 4.5 ft above the floor, and the particle detectors (counters) for stations 8, 13, and 15 were placed at heights of 7.0, 2.0, and 10.8 ft, respectively.

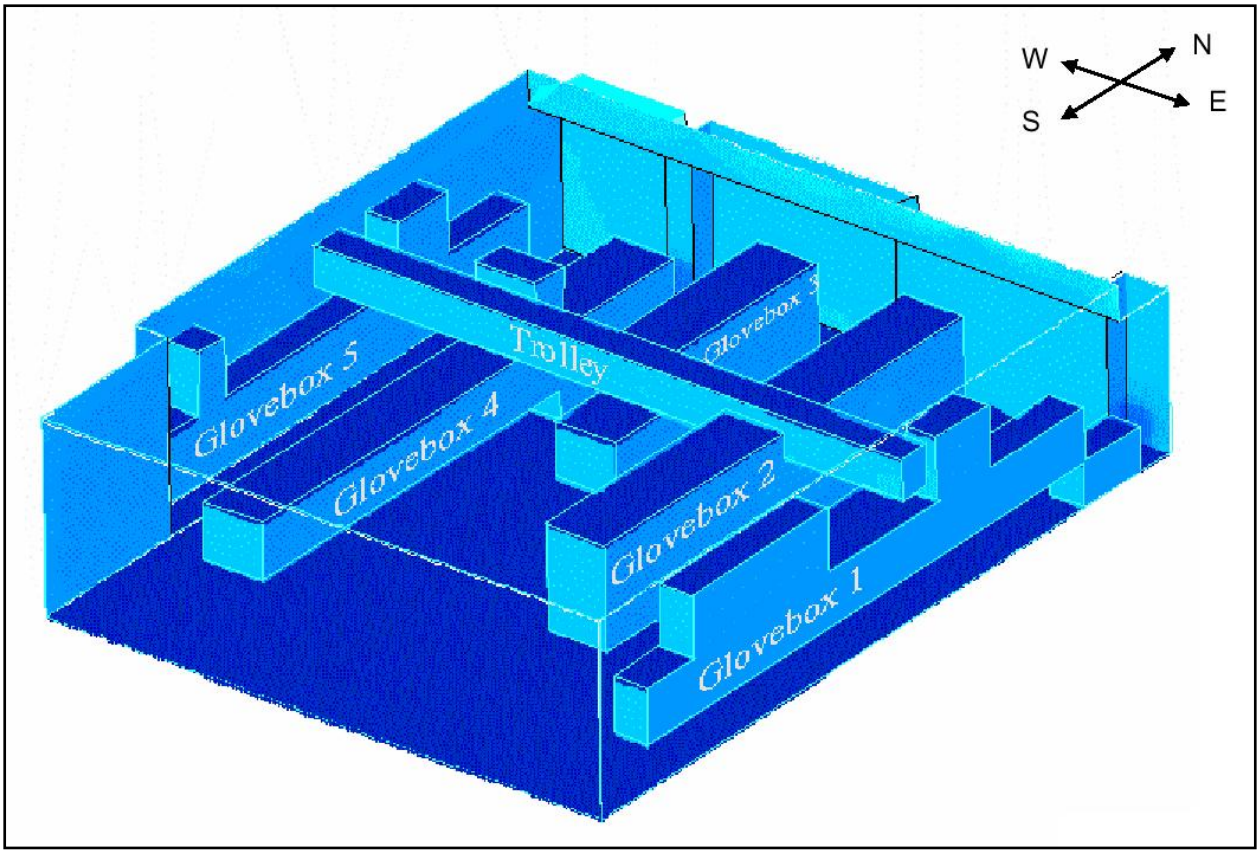


Figure 20. Isometric view of the LANL plutonium-facility workroom used for dispersion experiments by Whicker et al. (2000).

[1] Figures 20 to 23, which show the experimental workroom layout, are taken directly from the LANL report by Whicker et al. (2000).

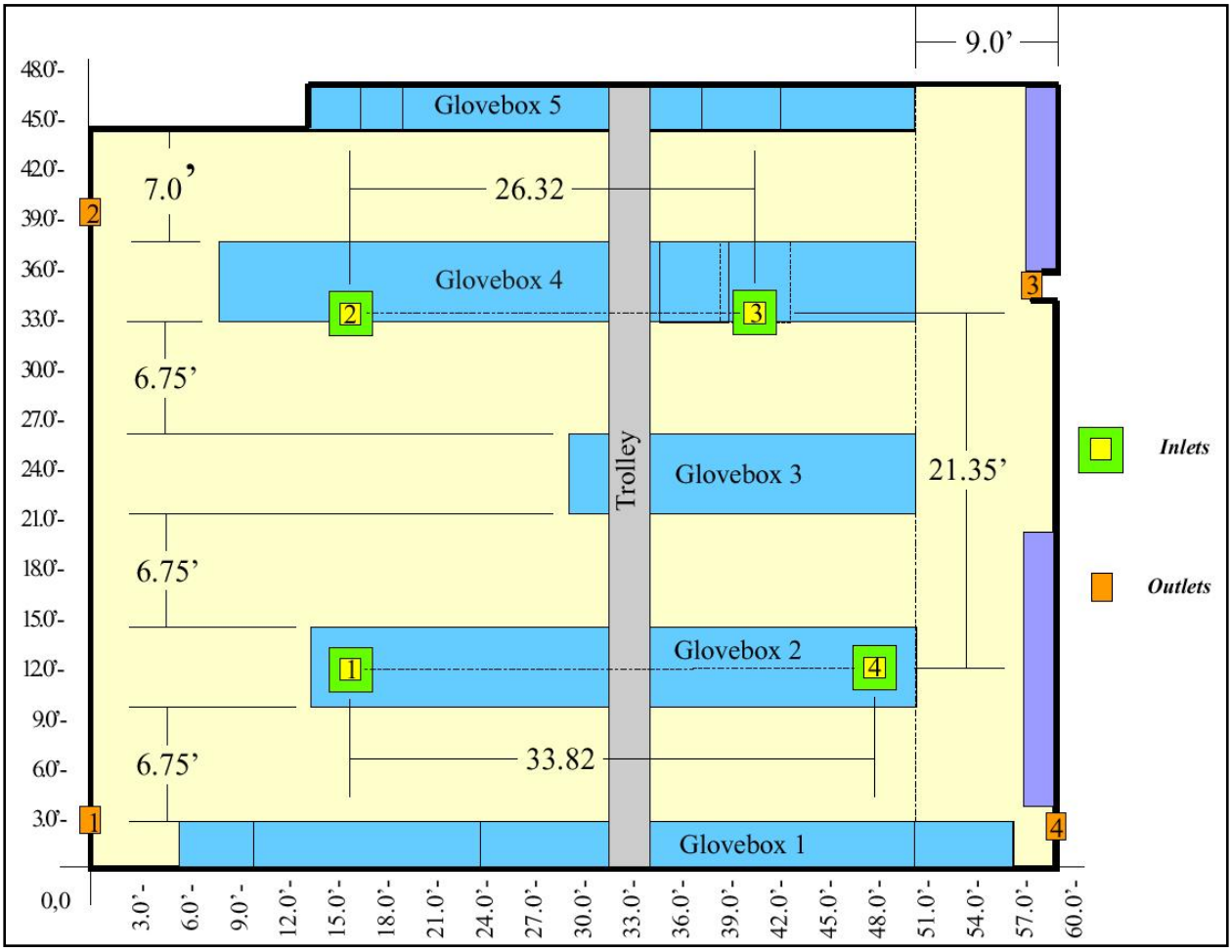


Figure 21. Overhead view of the workroom.

During the LANL dispersion experiments, the room ventilation system was operated with a constant (total) flow rate of approximately 85.3 cfs, which was equivalent to 10 room-air exchanges per hour. The inlet diffusers were circular with a diameter of 1 ft, and these injected air laterally into the room at a 28-degree angle relative to the ceiling. The exhaust vents were 1.33 ft wide and 2.0 ft high, and centered 2.0 ft above the floor. The flow rates through the four inlet diffusers were roughly the same (about 21.33 cfs each), and the average flow rates through exhaust vents 1, 2, 3, and 4 were 25.6, 24.7, 14.5, and 20.5 cfs, respectively.[1]

[1] Flow rates were provided by S. Konecni, co-author of the report by Whicker et al. (2000), who prepared and executed the CFD simulations for the CAM placement study.

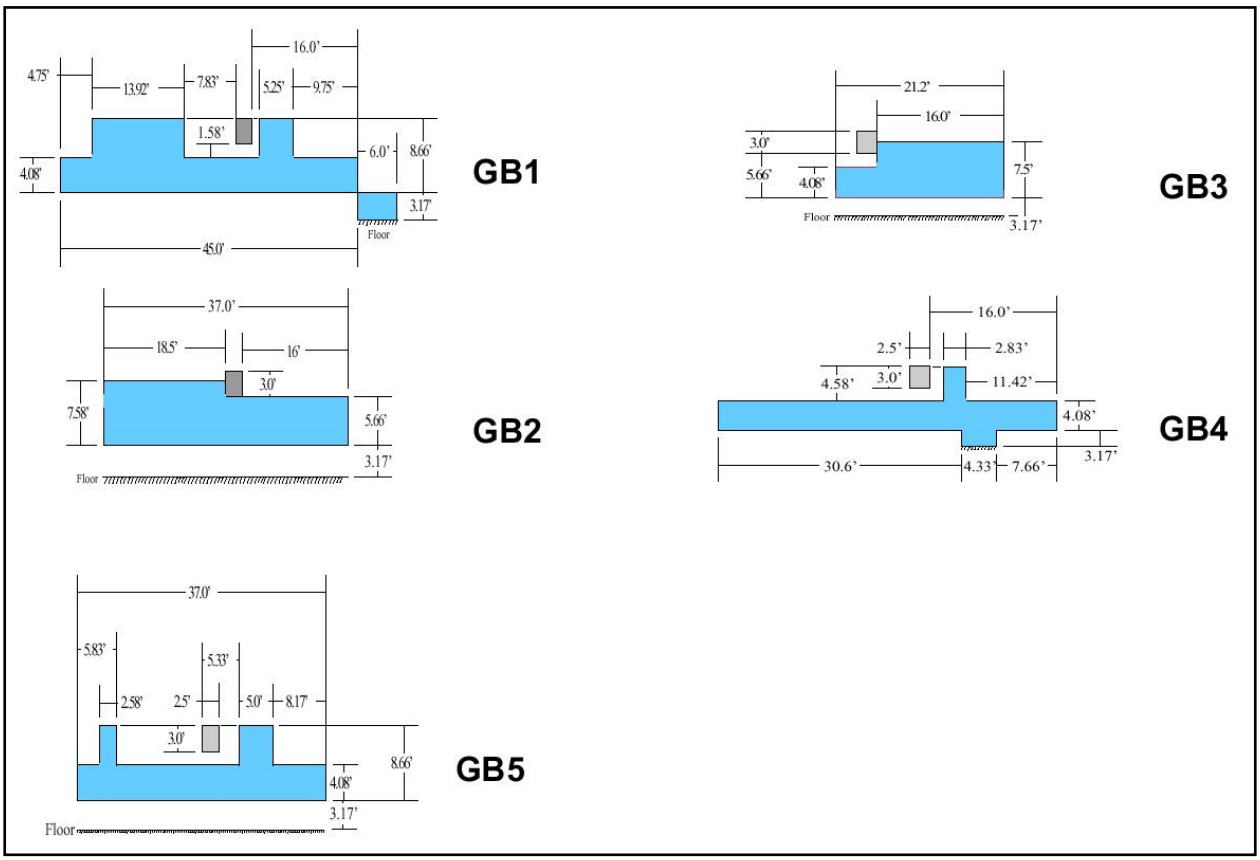


Figure 22. Dimensions for glove boxes (GB) in the workroom.

The computational grid used for the PAR3D simulations had an average spatial increment of 0.5 ft in all three directions, with slightly smaller increments around the inlet diffusers. Figure 24 shows overhead views of the grid at different heights (z) relative to the floor, and Figure 25 shows side views of the grid at the glove boxes. The grid was sufficiently fine to resolve primary eddies in the simulated flow field, but not fine enough to resolve boundary layers along the walls and the glove boxes. Therefore, all solid surfaces were modeled as slip (frictionless) boundaries.

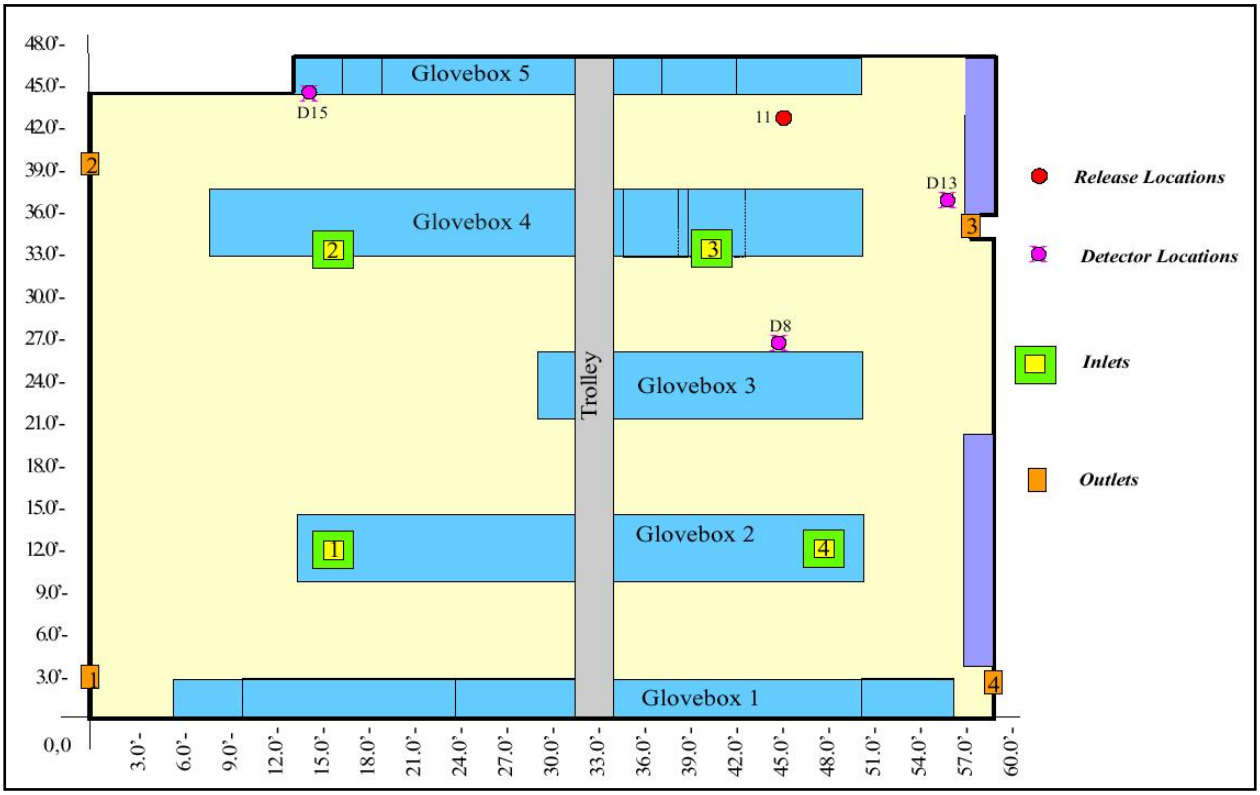


Figure 23. Release and detector stations for the particle-dispersion experiments.

The circular (ceiling-mounted) inlet diffusers were represented as rectangular insets in the PAR3D grid.[1] The inflow distributions were adjusted to achieve a roughly circular (axially symmetric) flow pattern in the radial direction, with a 28-degree angle relative to the ceiling. The resulting velocity vectors are shown around the base of Inlet Diffuser 1 in Figure 26.

Assuming that the inlet diffusers were supplied by smooth-walled, 1-ft-diameter, circular ducts, and that the supply flows (21.33 cfs) were fully turbulent, then Equations 4.7 and 4.10 imply that the supply-flow turbulence energy would have been k = 3.42 ft²/sec². Further assuming that Equation 4.11 approximates the local dissipation rate in the ducts (with δ_1 representing distance from the walls), then the average dissipation

[1] The floor is located at z = 0.0 ft, and the ceiling at z = 13.6 ft. The xy-locations of the inlet diffusers are indicated by the four small openings in the ceiling grid in Figure 24, and the downward protrusion of the modeled diffusers is shown in Figures 25 and 26. The (solid) base of each diffuser was three grid spaces square, with a total base area of 1.0 ft². Each diffuser protruded two spaces (0.67 ft) downward into the grid, and air was injected laterally from the lower cell faces adjacent to the base. The upper lateral faces of the diffuser were represented as solid faces, which allowed return flow along the ceiling to turn downward *at* the diffuser.

rate would have been ε = 15.6 ft²/sec³, which is equivalent to an eddy viscosity υ_T = 0.0675 ft²/sec.

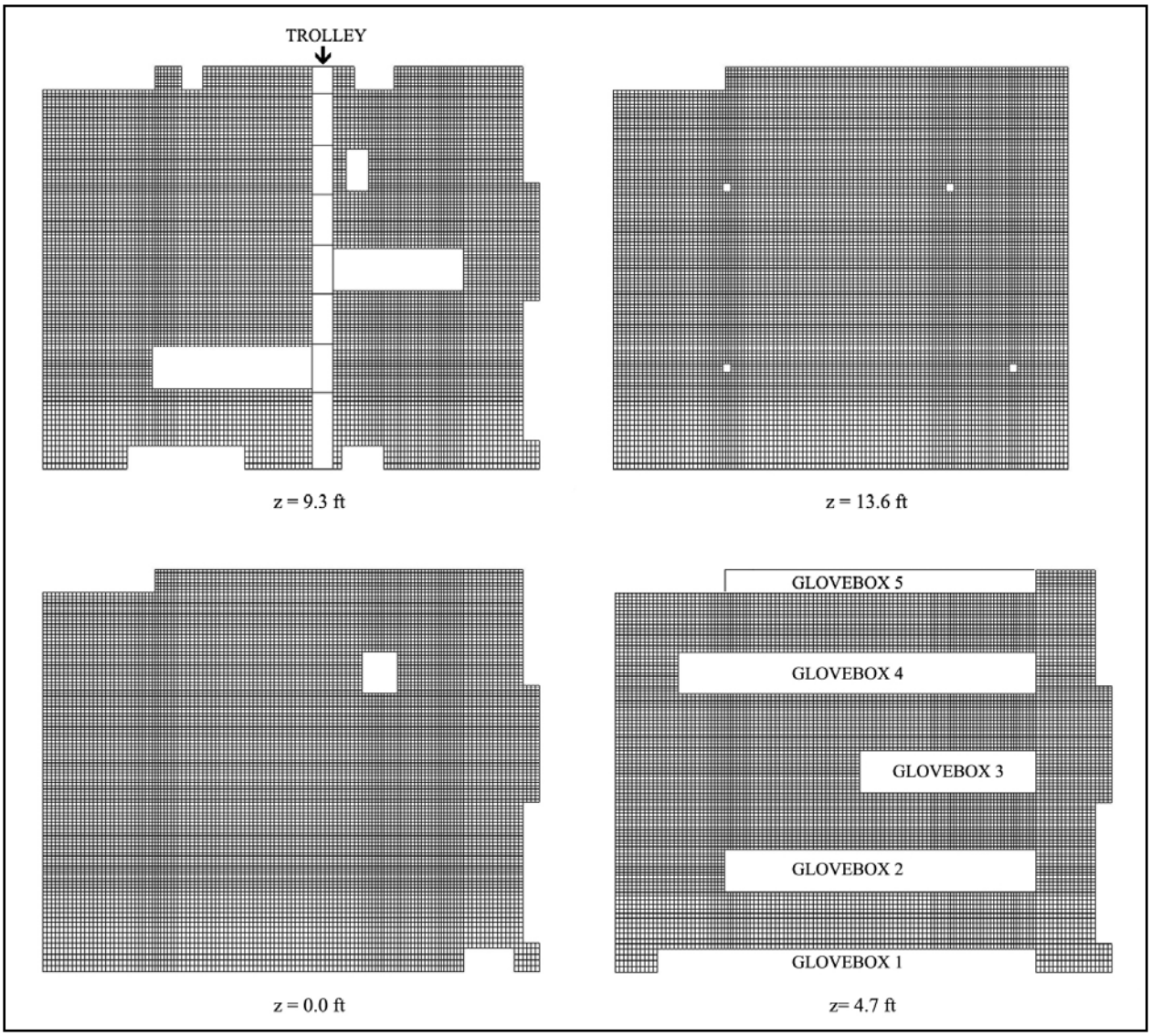


Figure 24. Overhead views of computational grid at different heights (*z*) above the floor of the workroom.

Figure 25. Side views of computational grid (looking west) at lateral positions (*y*) relative to the east wall of the workroom.

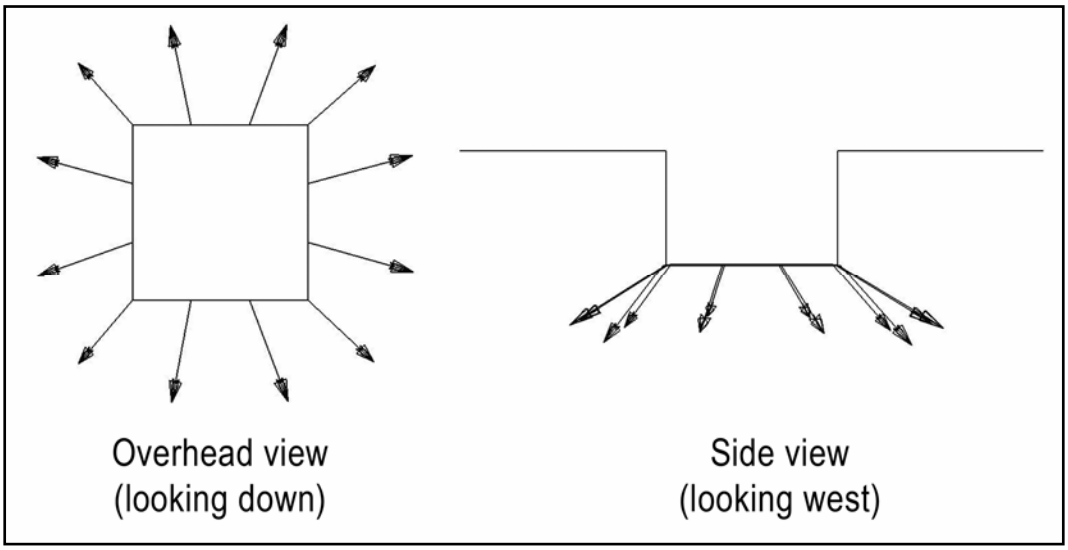


Figure 26. Overhead and side views of computed velocity vectors around the base of Inlet Diffuser 1 for the workroom.

These values for supply-flow turbulence energy and eddy viscosity represent approximate lower bounds for the workroom inflow values at the diffusers. When they were used in a PAR3D simulation of the room air flow, however, no steady state was achieved. After about 10 min of

simulated time, the flow reached a quasi-periodic condition that was well-behaved but persistently time-varying. Moreover, reducing the time-step Δt (from its nominal value of 0.02 sec) did not eliminate the unsteadiness; it merely provided better time-resolution of the unsteady behavior.

When the inflow value for eddy viscosity was increased to υ_T = 0.10 ft²/sec, the flow remained unsteady; but when it was increased to υ_T = 0.15 ft²/sec, the flow approached steady state after about 20 min of simulated time. When these two values were used as fixed (uniform) viscosities throughout the workroom, the lower value produced unsteady flow, and the higher value produced steady flow.

In retrospect, perhaps, it should not be surprising that there existed a viscosity threshold for steady-state flow in the room under consideration. The glove boxes and the overhead trolley all represent obstacles that could individually trigger vortex shedding in a simpler (unidirectional) flow situation; but the configuration of the workroom is so complex that one cannot guess the spatial distribution or the temporal behavior of the flow in advance. In any case, increasing the value of υ_T beyond 0.15 ft²/sec ensured that PAR3D would yield a steady state for the workroom discretized as shown in Figures 24 and 25.

In the absence of experimentally determined values for the turbulence energy and the eddy viscosity (or alternatively the dissipation rate) at the diffusers, results are presented that were obtained using k = 3.42 ft²/sec² and υ_T = 0.15 ft²/sec as the inflow boundary values. Using the flow rates outlined above for the inlet diffusers and the exhaust vents, PAR3D was executed for 30 min of simulated time to ensure that the computed flow reached steady state. Figures 27 to 30 present overhead views of the computed velocity vectors at four different heights above the floor, and Figure 31 offers side views at four lateral stations between the glove boxes.

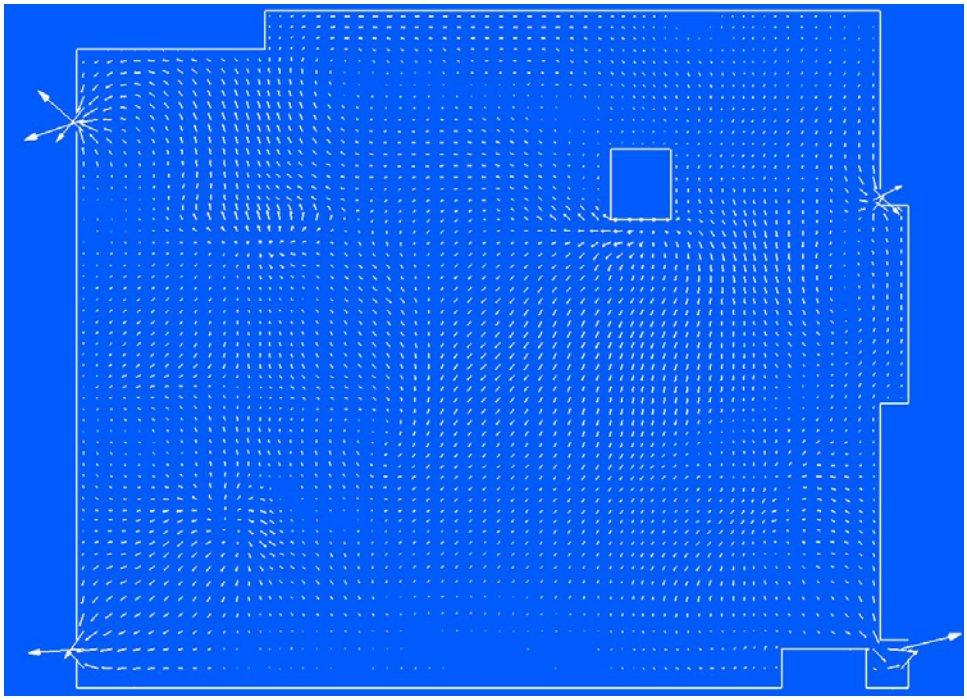


Figure 27. Overhead view of computed velocity vectors 2.0 ft above floor of the workroom.

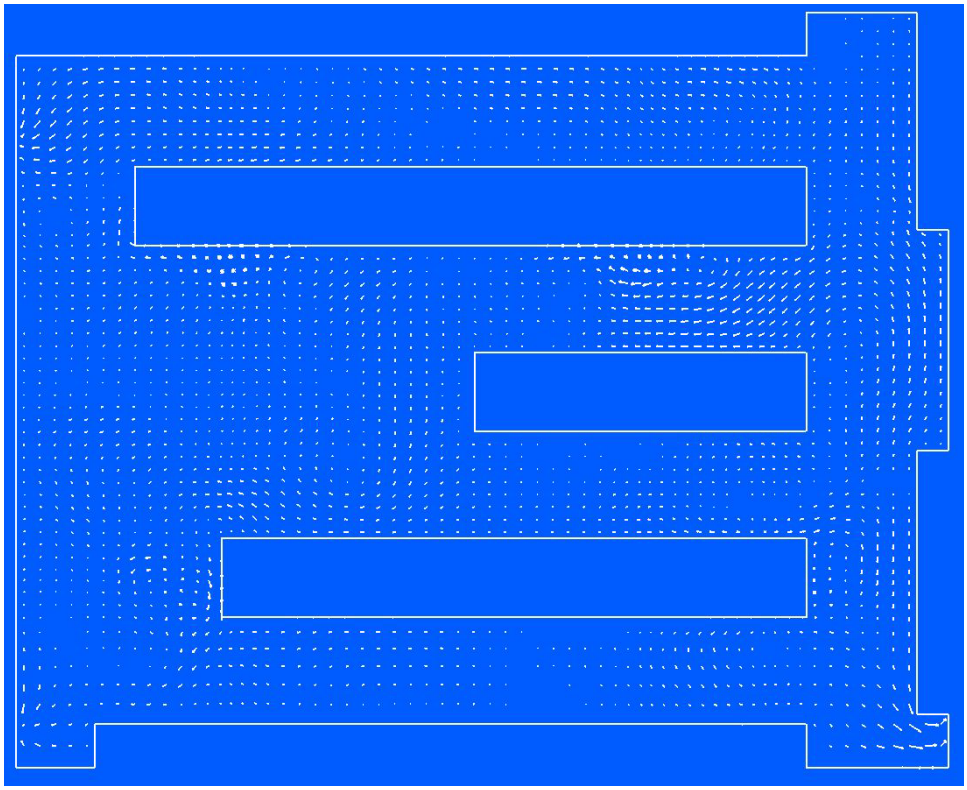


Figure 28. Overhead view of computed velocity vectors 4.2 ft above floor of the workroom.

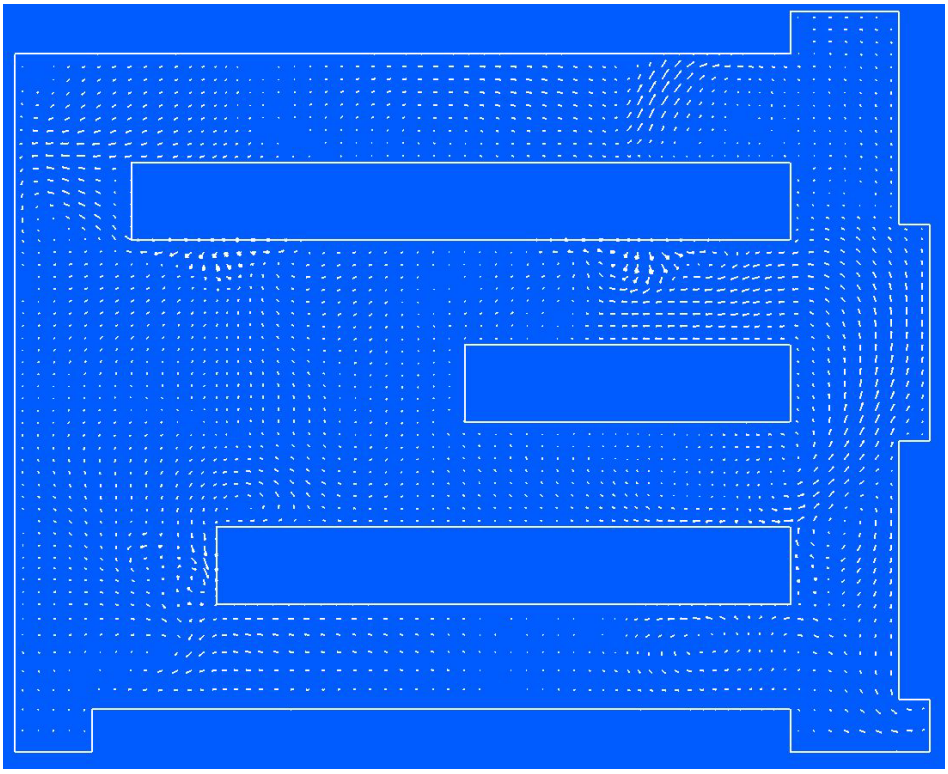


Figure 29. Overhead view of computed velocity vectors 6.2 ft above floor of the workroom.

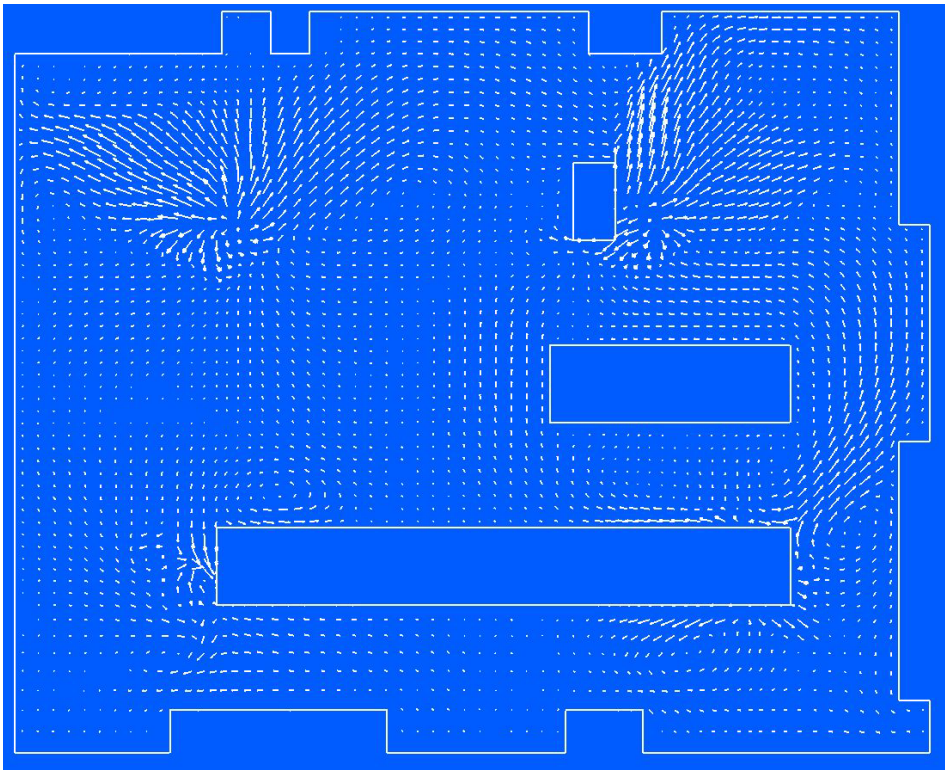


Figure 30. Overhead view of computed velocity vectors 8.3 ft above floor of the workroom.

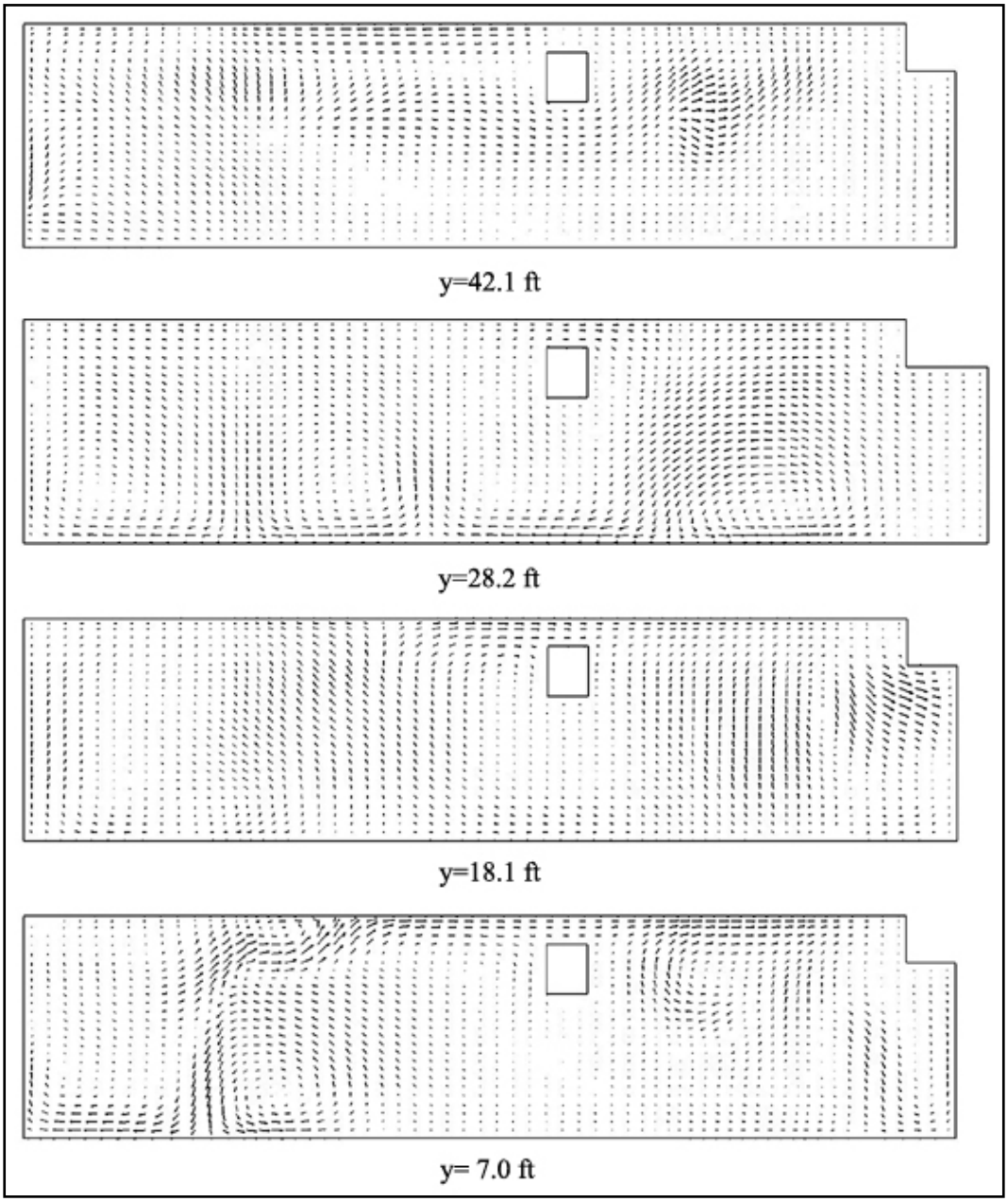


Figure 31. Side views (looking west) of computed velocity vectors at intermediate lateral positions (*y*) between glove boxes.

In the LANL experiments, neutrally buoyant aerosol particles were released from station 11 at a constant rate of 0.1 kg/sec for 30 sec. Following the same procedure with PAR3D, a point source was activated at station 11 after 30 min of flow simulation, which released contaminant at 0.1 kg/sec. After 30 sec of simulated release, the source was shut off, but the dispersion simulation (with an assumed Prandtl-Schmidt number of unity) was continued for an additional 360 sec.

Figures 32, 33, and 34 present color maps of particle concentrations at heights of 2.0, 7.0, and 10.8 ft, predicted 120 sec after source activation (and 90 sec after shutoff). These are the heights of detector stations 13, 8, and 15, respectively, for which predicted and observed time-histories are compared in Figures 35, 36, and 37. Note that the instant represented by the color maps (120 sec after starting release) falls between the times at which the peak concentrations were predicted for stations 8 and 13.

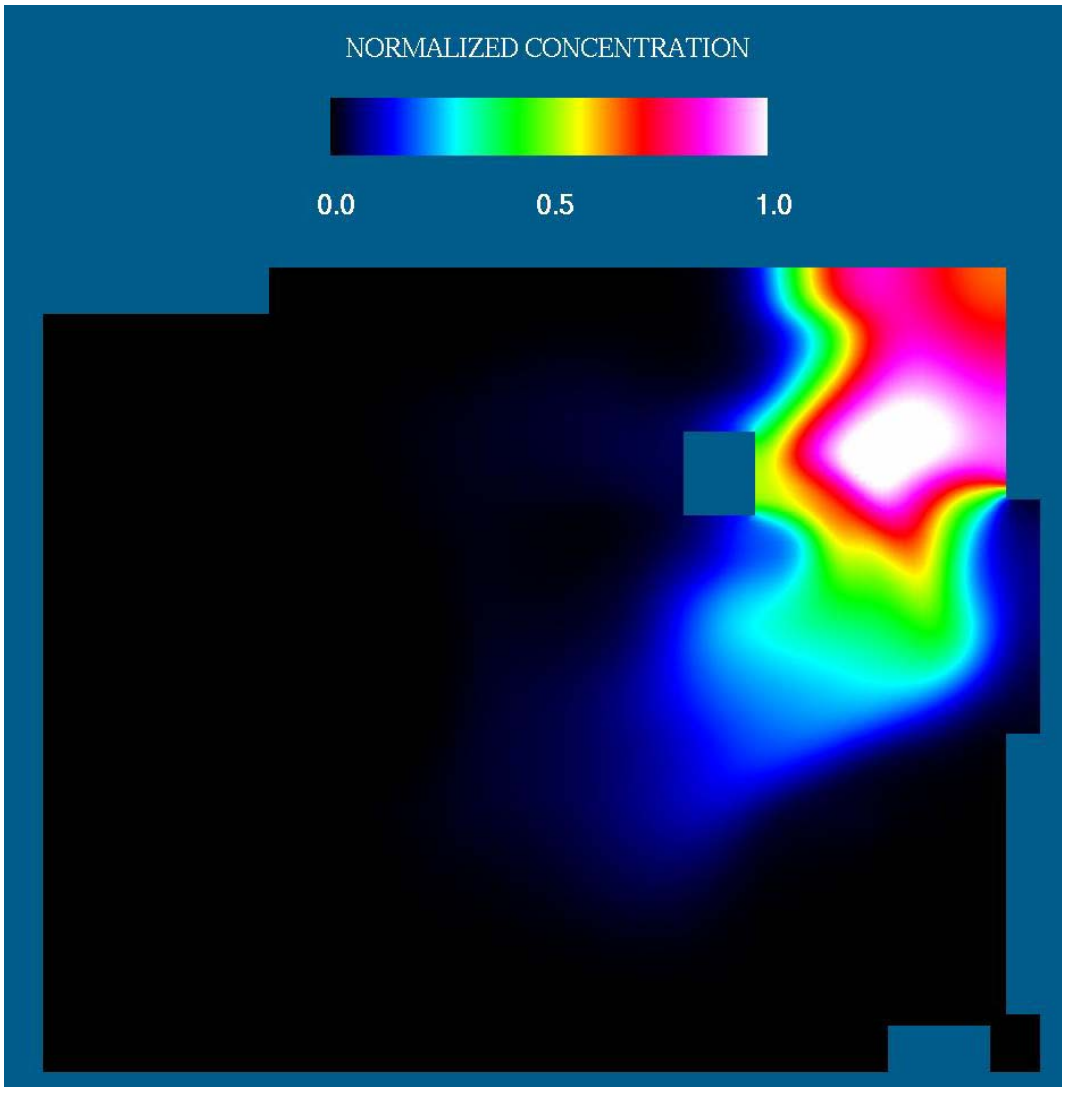


Figure 32. Overhead view of normalized particle concentration, 2 ft above floor of workroom, 120 sec after beginning of release, predicted by PAR3D.

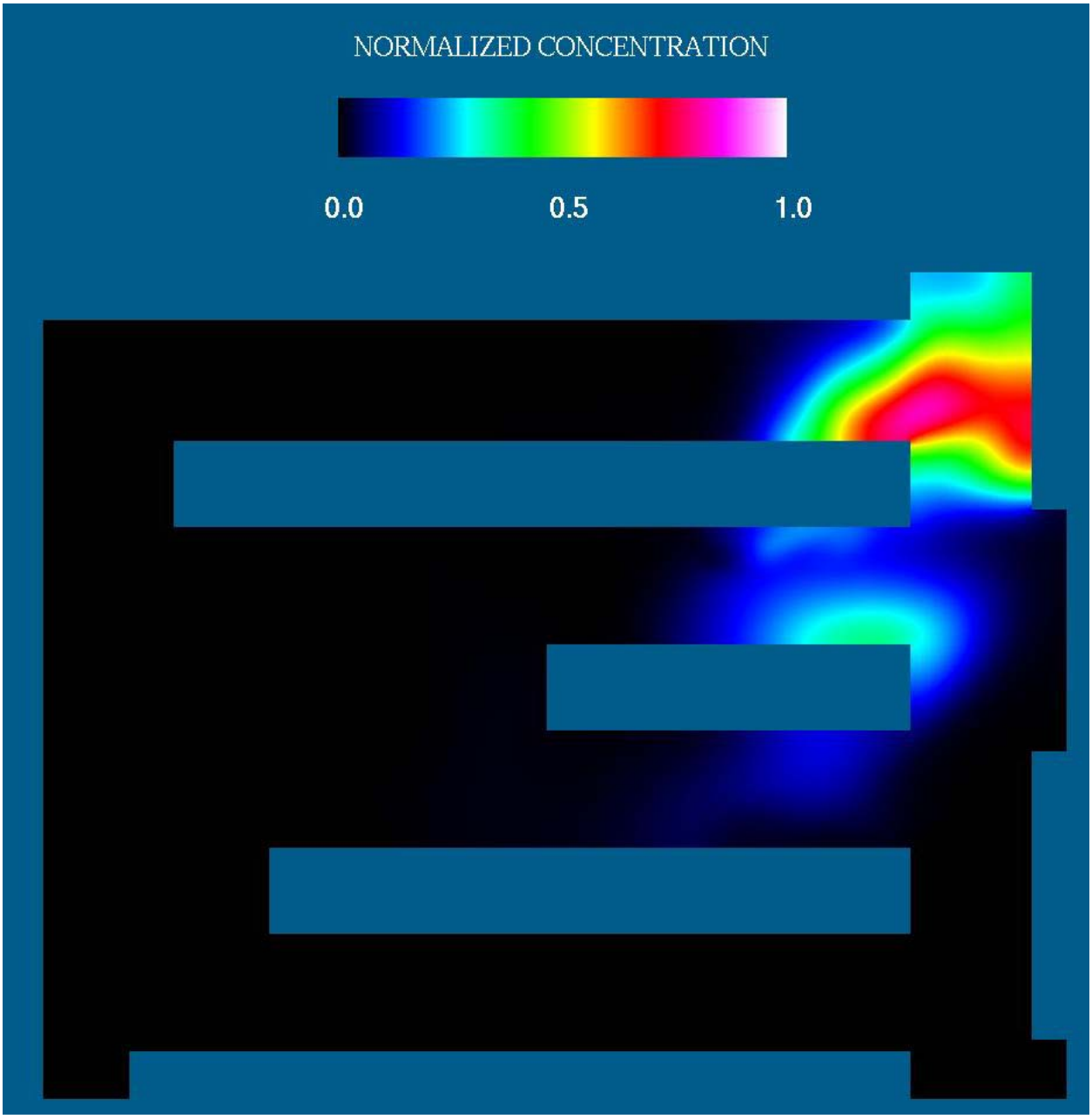


Figure 33. Overhead view of normalized particle concentration, 7 ft above floor of workroom, 120 sec after beginning of release, predicted by PAR3D.

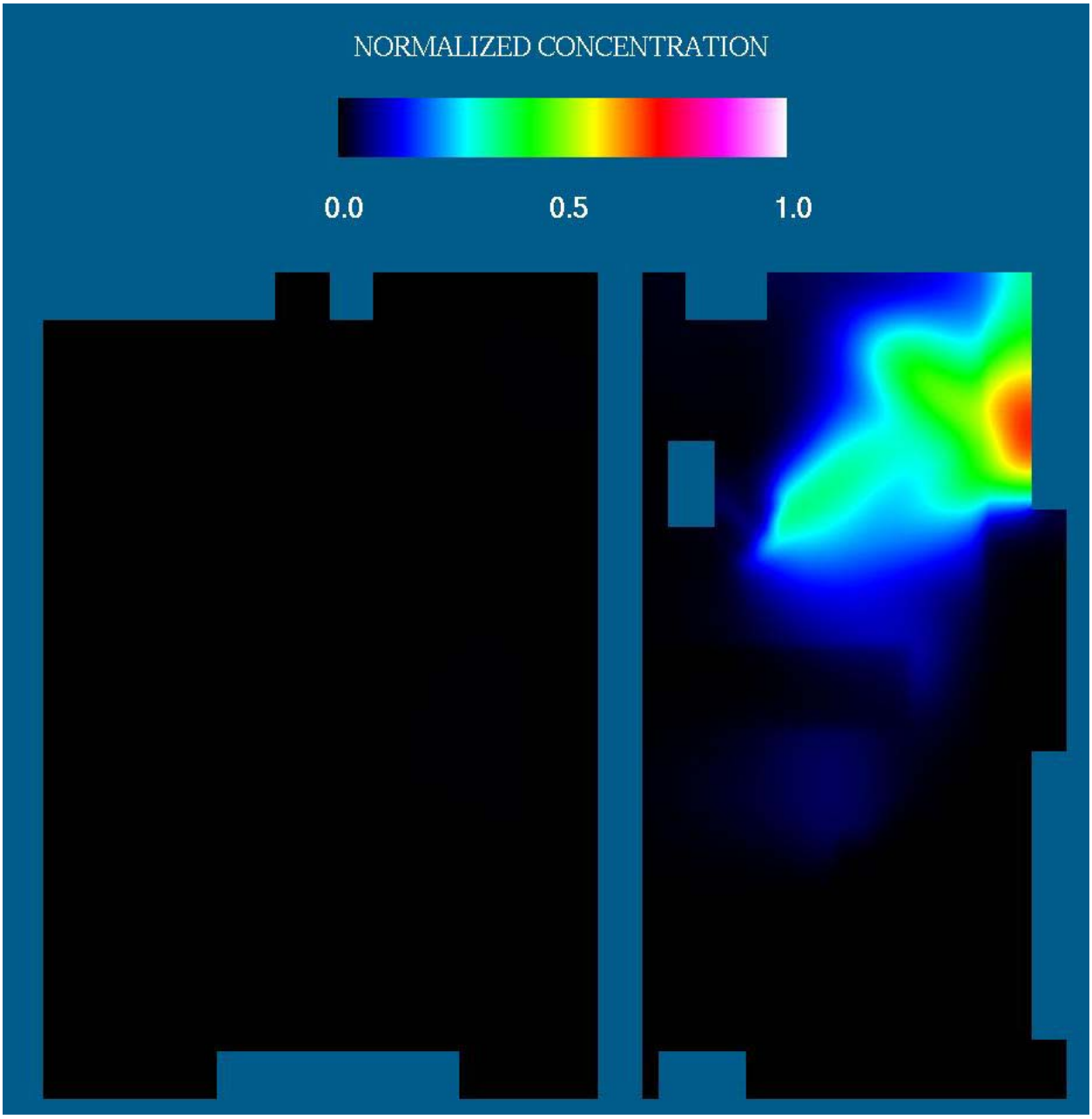


Figure 34. Overhead view of normalized particle concentration, 10.8 ft above floor of workroom, 120 sec after beginning of release, predicted by PAR3D.

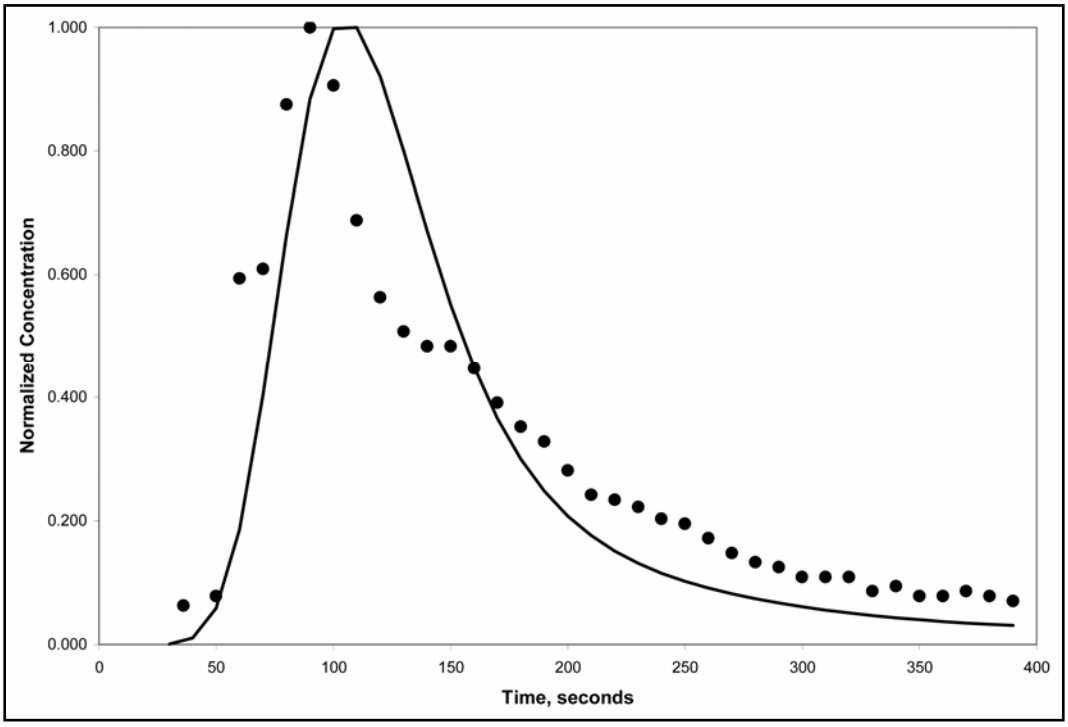


Figure 35. Comparison of normalized particle-concentration data (•) with PAR3D predictions (—) for station 13, located 2 ft above workroom floor.

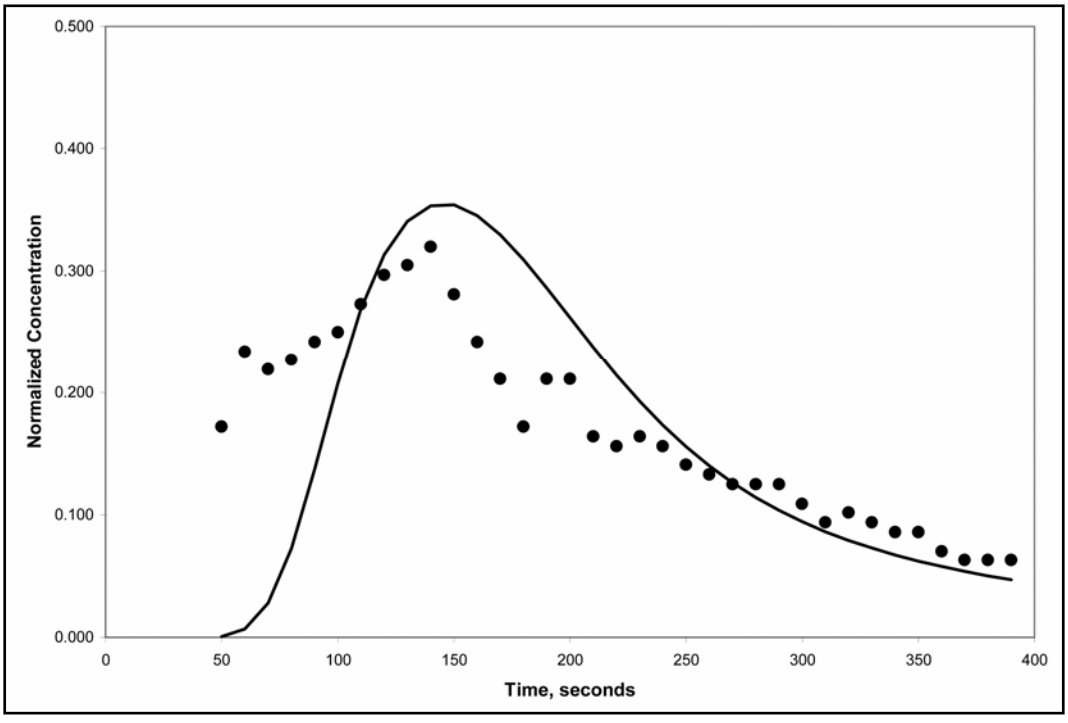


Figure 36. Comparison of normalized particle-concentration data (•) with PAR3D predictions (—) for station 8, located 7 ft above workroom floor.

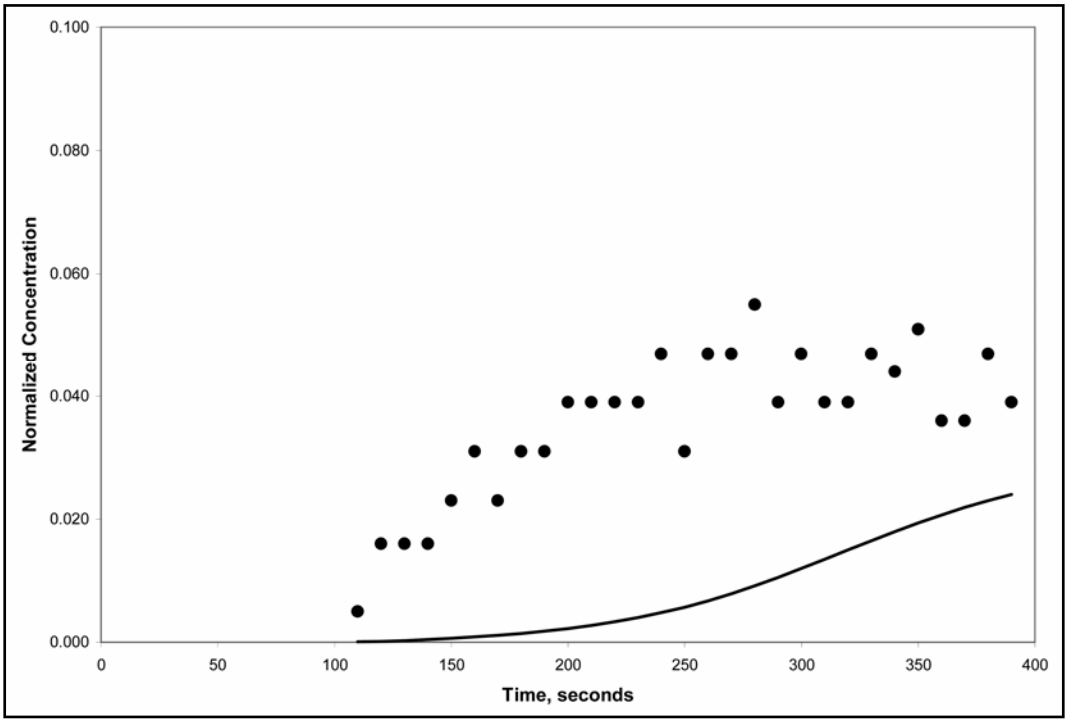


Figure 37. Comparison of normalized particle-concentration data (•) with PAR3D predictions (—) for station 15, located 10.8 ft above workroom floor.

PAR3D computes contaminant concentration in units of mass per unit volume, but the experimental concentrations in the report by Whicker et al. (2000) were given as particles per unit volume. No masses were given for the aerosol particles, so it is not possible here to compare the computed and measured concentrations in the same *dimensional* units. The normalized PAR3D predictions in Figures 32 to 37 were obtained by dividing all the (dimensional) predicted concentrations by the *maximum* (dimensional) concentration *predicted* for detector station 13. Likewise, the normalized experimental data were obtained by dividing all the (dimensional) measured concentrations by the *maximum* (dimensional) concentration *measured* at station 13.

The most favorable comparison with the experimental data was achieved for station 13, as indicated by Figure 35. The arrival time for the peak concentration is late by about 20 sec, but the shape of the curve compares well with the data. Note that the predicted peak matches the observed peak for *this* station because of the normalization.

Similar comments apply for station 8, as demonstrated by Figure 36. The initial rise in concentration is late by about 40 sec, but the peak concentration is late by no more than 20 sec. The shape of the predicted

curve also compares well with the data, and the predicted peak matches the observed peak within 10 percent. Note that the agreement between predicted and observed peaks for *this* station indicates that the relative peaks are correctly predicted for detectors 8 and 13, with or without normalization.

Predicted results for station 15 are poor in comparison with those for stations 8 and 13, as is obvious from Figure 37. The initial rise in concentration is late by at least 100 sec, and the highest predicted concentration, which occurs at the end of the simulated time-interval, is lower than the observed peak by a factor of two or more.

The predictions in Figures 35 to 37 represent the best that PAR3D was able to achieve (with steady flow) for the workroom under consideration. In subsequent simulations, using k = 3.42 ft²/sec² with v_T > 0.15 ft²/sec at the diffusers, it was found that the agreement between prediction and experiment worsened with increasing v_T. In this case, the provisional rule of thumb (for predicting dispersion) seems to be that one should use an inflow value for v_T roughly equal to the smallest value that produces steady flow.

7 Conclusions

The PAR3D code, initially developed as a numerical model for simulating flow and transport in deep water, has been extended for applications involving air flow through complex buildings and other enclosures. Comparisons with experimental results here and elsewhere (Bernard et al.1999; Bernard 2003) indicate that velocity predictions may be acceptable, but that dispersion predictions may be somewhat imprecise. Nevertheless, experience thus far suggests that the model is qualitatively correct in its predictions (most importantly those presented herein for a complex enclosure) and that it may be useful for gaining insight into the dispersion of contaminants under diverse flow conditions.

When the code is executed *with* the turbulence model, inflow values for the turbulence energy and its dissipation rate (or associated eddy viscosity) may serve as free parameters for quantitatively tuning predicted results. These quantities can be determined in principle from experiments or from empirical considerations, but their values in practice may be uncertain by an order of magnitude or more. In any case, the main reasons for including the turbulence model are to capture the qualitative effects of turbulence on the mean flow and to avoid having the model user specify a viscosity for the entire flow field.

When the code is executed *without* the turbulence model, then the model user must specify a single (uniform) value for viscosity that will be used for the duration of the simulated time-interval. Unless the aim is to reproduce truly laminar flow at a given Reynolds number, this viscosity serves as a free parameter that can be used to adjust predicted results for agreement with experimental data and/or sponsor expectations.

Although free parameters can be used (within reasonable bounds) to tune quantitative predictions made by PAR3D, qualitative verisimilitude is mainly a matter of adequate grid resolution in three dimensions. At one extreme, if the grid is too coarse to capture the primary eddies in a given setting, then the resulting flow simulation is qualitatively worthless. At the other extreme, grid independence is achieved when further refinement of the grid produces no further change in quantitative predictions. The degree of grid refinement employed for the applications in Chapter 6 was

more than adequate for the primary flow (and related transport) but not quite sufficient for true grid independence.

The time required for PAR3D execution depends on the number of processors used, the (floating-point) computational speed of the individual processors, and the time required for communication between processors. Computational speed and communication time vary considerably from computer to computer, but they continue to improve dramatically with each new generation of computers. For the applications presented in Chapter 6, the simple enclosure required several hours (with four processors) and the complex enclosure required an overnight run (with eight processors) on an SGI Origin 3900 computer.

References

Anderson, D. A., J. C. Tannehill, and R. H. Pletcher. 1984. *Computational fluid mechanics and heat transfer*. New York: Hemisphere Publishing Corporation, McGraw-Hill.

Azbel, D. 1981. *Two-phase flows in chemical engineering*. Cambridge University Press.

Bernard, R. S. 1995. *Preliminary development of a three-dimensional numerical model for reservoir hydrodynamics*. Technical Report HL-95-9. Vicksburg, MS: U.S. Army Engineer Waterways Experiment Station.

Bernard, R. S. 1998. *MAC3D: numerical model for reservoir hydrodynamics with application to bubble diffusers*. Technical Report CHL-98-23. Vicksburg, MS: U.S. Army Engineer Waterways Experiment Station.

Bernard, R. S. 2003. Gas-transfer simulations with the PAR3D numerical model. *Proceedings of the Watershed System 2003 Conference*. Portland, OR: U.S. Army Corps of Engineers.

Bernard, R. S., and H. Kapitza. 1992. How to discretize the pressure gradient for curvilinear MAC grids. *Journal of Computational Physics* 99(2): 288-298.

Bernard, R. S., M. L. Schneider, A. K. Stagg, and R. S. Maier. 1999. PAR3D: Numerical model for three-dimensional incompressible flow using structured grids and parallel processors. In *Proceedings of the ASCE International Water Resources Engineering Conference*. Seattle, WA: American Society of Civil Engineers.

Fofonoff, N. P. 1985. Physical properties of seawater: A new salinity scale and equation of state of seawater. *Journal of Geophysical Research* 90(C2): 3332-3342.

Fortesque, G. E., and J. R. A. Pearson. 1967. On gas absorption into a turbulent liquid. *Chemical Engineering Science* 22: 1163.

Gropp, W., E. Lusk, and A. Skjellum. 1996. *Using MPI: Portable parallel programming with the message-passing interface*. Cambridge, MA: MIT Press.

Kapitza, H., and D. Eppel. 1987. A 3-D Poisson solver based on conjugate gradients compared to standard iterative methods and its performance on vector computers. *Journal of Computational Physics* 68: 474-484.

Kuethe, A. M., and J. D. Schetzer. 1967. *Foundations of aerodynamics*. New York: John Wiley and Sons.

LaBonte, P., and W. Spani. 2005. Preventing saltwater intrusion. *The Military Engineer* May-June: 71.

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

Orlanski, I. 1976. A simple boundary condition for unbounded hyperbolic flows. *Journal of Computational Physics* 21: 251-269.

Rodi, W. 1980. *Turbulence models and their application in hydraulics*. Delft, Netherlands: State-of-the-Art Paper, International Association for Hydraulic Research.

Shimada, M., K. Okuyama, S. Okazaki, T. Asai, M. Matsukura, and Y. Ishizu. 1996. Numerical simulation and experiment on the transport of fine particles in a ventilated room. *Aerosol Science and Technology* 25: 242-255.

Whicker, J., S. Konecni, L. Parietti, et al. 2000. *Placement of continuous air monitors in PF-4 plutonium laboratories: Consensus findings and recommendations*. Los Alamos National Laboratory Report LA-UR-00-2311.

Appendix: Boundary Conditions for Turbulence

The k–ε turbulence model (Chapter 2) entails only the governing equations for the turbulence energy k and the turbulence-energy dissipation rate ε. Thus, the boundary conditions for these two quantities have to be determined from other considerations.

Inflow boundaries are straightforward in principle because the inflow values for k and ε should be the same as those in the flow upstream of the inflow. In practice, however, the flow upstream may be poorly characterized; this can render the inflow conditions somewhat problematic, as exemplified by the model applications discussed in Chapter 6. Outflow boundaries present less of a problem because they usually lend themselves to the extrapolation procedure (Equations 4.15 to 4.20) outlined in Chapter 4. Slip boundaries are even less troublesome because they can be idealized as symmetry boundaries through which there is no gradient flux of k or ε, as stipulated by Equation 4.1.

No-slip boundaries (friction walls) present a different problem altogether because they require in principle a detailed characterization of the turbulent flow near the walls. This often demands much finer grid resolution than is feasible in practice, and one is forced to seek an easier way out. The most common remedy seems to be the use of wall functions for the values of k and ε *in* the flow *near* the walls, as discussed by Rodi (1980). For unidirectional shear flow without a pressure gradient, these are given by

$$k = \frac{\upsilon_*^2}{C_\nu^{1/2}} \quad \text{for} \quad z \le \delta_1 \tag{A.1}$$

$$\varepsilon = \frac{\upsilon_*^3}{\kappa z} \quad \text{for} \quad z \le \delta_1 \tag{A.2}$$

where

υ_* = friction velocity
C_υ = 0.09

z = normal distance from wall
δ_1 = thickness of wall-adjacent grid cell normal to wall
κ = 0.40

Since these wall functions are imposed *in* the flow rather than *on* the walls, they do not represent true boundary conditions. In fact, they are truly valid only for turbulent shear flow in equilibrium, such that

$$P = \varepsilon \tag{A.3}$$

where P is the turbulence-energy production rate, given by Equation 2.11 in Chapter 2.

As an alternative to wall functions, one seeks boundary conditions that yield the near-wall values for k and ε (Equations A.1 and A.2) only at equilibrium, and not otherwise. For a shear flow in equilibrium, the eddy viscosity very near the wall is given by

$$\nu_T = \kappa \upsilon_* z \tag{A.4}$$

and the gradient flux (per unit area) of k normal to the wall reduces to

$$\frac{\nu_T}{\sigma_k}\frac{\partial k}{\partial z} = \frac{\kappa \upsilon_* z}{\sigma_k}\frac{\partial k}{\partial z} \tag{A.5}$$

Assuming that

$$\frac{\partial k}{\partial z} \approx \frac{k - k_*}{z} \tag{A.6}$$

with

$$k_* \equiv \frac{\upsilon_*^2}{C_\nu^{1/2}} \tag{A.7}$$

then Equation A.5 gives way to

$$\frac{\nu_T}{\sigma_k}\frac{\partial k}{\partial z} \approx \frac{\kappa \upsilon_*}{\sigma_k}\left(k - k_*\right) \tag{A.8}$$

When imposed as the wall-normal gradient flux *at* the wall, Equation A.8 ensures that $k \to k_*$ as the flow approaches equilibrium and the governing equation (Equation 2.7) reduces to the limiting condition,

$$\frac{\partial}{\partial z}\left(\frac{\nu_T}{\sigma_k}\frac{\partial k}{\partial z}\right) \to 0 \tag{A.9}$$

The value of k is uniform when the flow is in equilibrium, as dictated by Equation A.1, and the appropriate boundary value is unambiguous. In contrast, however, Equation A.2 indicates that ε increases sharply as $z \to 0$, and it is not clear what boundary value should be used to elicit the z–dependence evident in the wall function (A.2). Therefore, one must begin by examining the equilibrium form of the governing equation (Equation 2.8) for ε:

$$\frac{\varepsilon^2}{k}\left(c_1 - c_2\right) + \frac{1}{\sigma_\varepsilon}\frac{\partial}{\partial z}\left(\nu_T \frac{\partial \varepsilon}{\partial z}\right) = 0 \tag{A.10}$$

First, consider the *average* dissipation rate in a wall-adjacent grid cell, approximated with the expression,

$$\overline{\varepsilon} \approx \frac{\beta \upsilon_*^3}{\kappa \delta_1} \tag{A.11}$$

in which $\beta \geq 1$. Substituting Equation A.1 for k and Equation A.11 for ε, the *first* term in Equation A.10 can be replaced with

$$\frac{\overline{\varepsilon}^2}{k}\left(c_1 - c_2\right) \approx \frac{\beta^2 C_\nu^{1/2} \upsilon_*^4}{\kappa^2 \delta_1^2}\left(c_1 - c_2\right) \tag{A.12}$$

Next, using central differencing to evaluate the *second* term in Equation A.10, one obtains

$$\frac{1}{\sigma_\varepsilon}\frac{\partial}{\partial z}\left(\nu_T \frac{\partial \varepsilon}{\partial z}\right) \approx \frac{1}{\sigma_\varepsilon \delta_1}\left[\left(\nu_T \frac{\partial \varepsilon}{\partial z}\right)_{z=\delta_1} - \left(\nu_T \frac{\partial \varepsilon}{\partial z}\right)_{z=0}\right] \tag{A.13}$$

Differentiating Equation A.2 with respect to z gives

$$\frac{\partial \varepsilon}{\partial z} = -\frac{\upsilon_*^3}{\kappa z^2} \tag{A.14}$$

Multiplying Equation A.14 by Equation A.4, and evaluating the result at $z = \delta_1$ yields

$$\left(\nu_T \frac{\partial \varepsilon}{\partial z}\right)_{z=\delta_1} = -\frac{\upsilon_*^4}{\delta_1} \tag{A.15}$$

with which Equation A.13 reduces to

$$\frac{1}{\sigma_\varepsilon}\frac{\partial}{\partial z}\left(\nu_T \frac{\partial \varepsilon}{\partial z}\right) \approx -\frac{1}{\sigma_\varepsilon \delta_1}\left[\frac{\upsilon_*^4}{\delta_1} + \left(\nu_T \frac{\partial \varepsilon}{\partial z}\right)_{z=0}\right] \tag{A.16}$$

After substituting Equations A.12 and A.16 for the first and second terms, respectively, in Equation A.10, and then rearranging, one finds that

$$\frac{1}{\sigma_\varepsilon \delta_1}\left(\nu_T \frac{\partial \varepsilon}{\partial z}\right)_{z=0} \approx \frac{\beta^2 C_\nu^{1/2} \upsilon_*^4}{\kappa^2 \delta_1^2}\left(c_1 - c_2\right) - \frac{\upsilon_*^4}{\sigma_\varepsilon \delta_1^2} \tag{A.17}$$

Next, assuming that

$$\frac{\partial \varepsilon}{\partial z} \approx \frac{\varepsilon - \varepsilon_*}{z} \tag{A.18}$$

and substituting Equation A.4 for υ_T, the left side of Equation A.17 takes the form

$$\frac{1}{\sigma_\varepsilon \delta_1}\left(\nu_T \frac{\partial \varepsilon}{\partial z}\right)_{z=0} \approx \frac{\kappa \upsilon_*}{\sigma_\varepsilon \delta_1}\left(\frac{\beta \upsilon_*^3}{\kappa \delta_1} - \varepsilon_*\right) \tag{A.19}$$

Combining Equations A.17 and A.19, it follows that

$$\varepsilon_* \approx \frac{\beta \upsilon_*^3}{\kappa \delta_1} + \frac{\upsilon_*^3}{\kappa \delta_1}\left[\frac{\beta^2 \sigma_\varepsilon C_\nu^{1/2}}{\kappa^2}\left(c_2 - c_1\right) + 1\right] \tag{A.20}$$

Lastly, the equilibrium relation, Equation 2.9, dictates that

$$c_2 - c_1 = \frac{\kappa^2}{\sigma_\varepsilon C_\nu^{1/2}} \tag{A.21}$$

so that Equation A.20 gives way to

$$\varepsilon_* \approx \frac{\upsilon_*^3}{\kappa \delta_1}\left[\beta^2 + \beta + 1\right] \tag{A.22}$$

Equation A.22 defines the boundary value ε_* that should be used for computing the gradient flux of ε normal to the wall in Equations A.19 and 4.4.

What now remains to be determined is the influence of the wall on the computed rate of turbulence-energy production P. Invoking Equations A.3 and A.11, the *average* value in a wall-adjacent grid cell is given approximately by

$$\bar{P} \approx \frac{\beta \upsilon_*^3}{\kappa \delta_1} \tag{A.23}$$

and Equations 2.6 and A.11 imply that the average eddy viscosity is

$$\bar{\nu}_T = \frac{\kappa \upsilon_* \delta_1}{\beta} \tag{A.24}$$

For the shear flow under consideration, however, Equation 2.11 reduces to

$$P_0 = \bar{\nu}_T \left(\frac{\partial u}{\partial z}\right)^2 \tag{A.25}$$

Due to the proximity of the wall, and the placement of computed velocities *in* the flow, PAR3D would be forced to evaluate the z-derivative in Equation A.25 at the *top* of the wall-adjacent grid cell (i.e., at $z = \delta_1$). Assuming a logarithmic velocity profile (similar to that in Equation 4.7c) with the form

$$u = \frac{\upsilon_*}{\kappa} \ln z + const \tag{A.26}$$

then the exact value of the z-derivative at $z = \delta_1$ is

$$\left(\frac{\partial u}{\partial z}\right)_{z=\delta_1} = \frac{\upsilon_*}{\kappa\delta_1} \tag{A.27}$$

Combining Equations A.24, A.25, and A.27 now yields

$$P_0 = \frac{\upsilon_*^3}{\beta\kappa\delta_1} \tag{A.28}$$

which is the production rate that PAR3D would predict for the wall-adjacent grid cell (using computed velocities in the flow field) *if* it could do so with infinite precision. Note, however, that the production rate given by Equation A.28 does not match the cell-averaged production rate given by Equation A.23. To make up the deficit, Equation A.28 (and likewise Equation 2.11) has to be supplemented with an additional wall-induced production term,

$$P_1 = \frac{\upsilon_*^3}{\kappa\delta_1}\left(\beta - \frac{1}{\beta}\right) \tag{A.29}$$

and this is the basis for the additional term in the wall-adjusted production rate given by Equation 4.12.

The free parameter β is dependent on the numerical discretization of the governing equations, and its value has to be determined (by trial and error) from computational experiments with unidirectional, wall-bound, turbulent shear flow, for which Equations A.1 and A.2 define the equilibrium values of k and ε, respectively. For this type of flow, in the absence of a pressure gradient, the Reynolds-averaged momentum equation (Equation 2.1) reduces to

$$\frac{\partial}{\partial z}\left(\nu_T \frac{\partial u}{\partial z}\right) = 0 \tag{A.30}$$

and the steady-state governing equations for the turbulence quantities become

$$P - \varepsilon + \frac{1}{\sigma_k}\frac{\partial}{\partial z}\left(\nu_T \frac{\partial k}{\partial z}\right) = 0 \tag{A.31}$$

$$\frac{\varepsilon}{k}\left(c_1 P - c_2 \varepsilon\right) + \frac{1}{\sigma_\varepsilon}\frac{\partial}{\partial z}\left(\nu_T \frac{\partial \varepsilon}{\partial z}\right) = 0 \tag{A.32}$$

Unpublished work by one of the authors (Bernard), in which Equations A.30 to A.32 were solved numerically with a 1-D MAC discretization (Chapter 3), indicates that the correct profiles for velocity, energy, and dissipation rate are consistently reproduced within a few percent when the value used for the free parameter is $\beta \approx 2$. Inserting this value for β in Equations A.22 and A.29, respectively, gives

$$\varepsilon_* \approx \frac{7\upsilon_*^3}{\kappa \delta_1} \tag{A.33}$$

$$P_1 = \frac{3}{2}\frac{\upsilon_*^3}{\kappa \delta_1} \tag{A.34}$$

The developments presented above are intended to remove any mystery associated with the wall-adjusted production rate, given by Equation 4.12, and the wall-normal gradient fluxes for k and ε, given by Equations 4.13 and 4.14, respectively, in Chapter 4. Note that the questionable assumptions in this process, even for turbulent shear flow in equilibrium, are those embodied by Equations A.6 and A.18. The end results, however, are boundary conditions that adequately reproduce wall-bound shear flow (when used with the $k - \varepsilon$ turbulence model) while accommodating other types of flow, *without* imposing wall-function values in the flow field *away* from the walls. The latter attribute is important when the grid is too coarse to resolve near-wall flow because it prevents the computed flow from being (erroneously) over-influenced by the walls.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
September 2007	Final report	

4. TITLE AND SUBTITLE

PAR3D: Numerical Model for Incompressible Flow with Application to Aerosol Dispersion in Complex Enclosures

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Robert S. Bernard, Phu V. Luong, and Mario J. Sanchez

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Coastal and Hydraulics Laboratory
U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

8. PERFORMING ORGANIZATION REPORT NUMBER

ERDC/CHL TR-07-9

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Corps of Engineers
Washington, DC 20314-1000

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report documents the development of the PAR3D numerical flow model, with emphasis on modifications incorporated to facilitate simulations of contaminant dispersion in complex buildings and other enclosures. PAR3D is a general-purpose computational fluid dynamics (CFD) code for predicting three-dimensional flow and transport in air, water, and other incompressible fluids. It includes a two-equation turbulence model with adjustments for buoyancy, as well as transport equations for suspended materials (contaminants), dissolved gases, and gas bubbles (in water). The code employs parallel processing with structured curvilinear grids, which may deform to accommodate quasi-static free-surface displacement in water.

The CFD applications included here concern air flow and aerosol dispersion inside two different man-made enclosures. These consist of single chambers with complex internal geometry, but PAR3D is applicable to multi-chamber enclosures as well. The reported developments are part of a larger computational and experimental effort to characterize the dispersion of contaminants in multi-room, multi-story buildings. Comparison of PAR3D predictions with experimental data from a multi-room test facility will be presented in a separate report.

15. SUBJECT TERMS

Computational fluid dynamics
Incompressible flow
Turbulence modeling
Aerosol dispersion
Contaminant transport

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED		82	