UCRL-MA-129067

User's Manual for MC_Wind: A New Mass-Consistent Wind Model for ARAC-3

Stevens T. Chan
Gayle Sugiyama

November 1997

Lawrence Livermore National Laboratory

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.

ABSTRACT 1
1. INTRODUCTION 1
2. NUMERICAL MODEL 2
2.1 Governing Equations 2
2.2 Parameterization for Atmospheric Stability 4
2.3 Finite Element Discretization 6
2.4 Specification of Boundary Conditions 8
2.5 Stabilization of the Discrete Poisson Equation 10
2.6 Solution of Linear System of Equations 12
2.7 Map Projections 13
3. USER'S GUIDE 14
3.1. The Namelist Groups 15
3.1.1 Namelist group 'mc_idfiles' 15
3.1.2 Namelist group 'mc_hills_xyz' 15
3.1.3 Namelist group 'mc_hills_ics' 16
3.1.4 Namelist group 'adapt_mc_adjust' 16
3.1.5 Namelist group 'adapt_mc_bc' 18
3.2. Tables of Namelist Groups 18
Table 3.1. Namelist MC_IDFILES 19
Table 3.2. Namelist MC_HILLS_XYZ 19
Table 3.3. Namelist MC_HILLS_ICS 19
Table 3.4. Namelist ADAPT_MC_ADJUST 20
Table 3.5. Namelist ADAPT_MC_BC 21
3.3 Input and Output Files 22
Table 3.6. MC_WIND input files 22
Table 3.7. MC_WIND output files 22
4. NUMERICAL EXAMPLES 23
4.1 Potential Flow Over a Hemispheric Hill 23
4.2 Stably Stratified Flow Around a Polynomial Hill 26
4.3 Application for the San Francisco Bay Area 29
4.4 Wind Flow in the LANL Area 31
ACKNOWLEDGMENTS 34
REFERENCES 34

USER'S MANUAL FOR MC_WIND: A NEW MASS-CONSISTENT WIND MODEL FOR ARAC-3

S.T. Chan
G. Sugiyama

ABSTRACT

Based on a mixed variational principle and the Galerkin finite element method (GFEM), a new model for efficiently generating mass-consistent wind fields over continuous terrain has been developed. The main features of the model include: treatment of continuous terrain, variable horizontal and vertical grid resolution, grid-point representation of the wind fields, ability to preserve velocities at desired grid points, incorporation of map projections, efficient iterative solvers, and a new parameterization for atmospheric stability and topographical effects. This report presents a description of the model, a user's guide for the computer code, and four numerical examples to illustrate the usage of the model.

1. INTRODUCTION

Three-dimensional wind fields are required in many practical applications. Some examples are the assessments of the atmospheric transport and dispersion of pollutants, selection of potential sites for wind energy generation, and studies of wind effects on structures such as buildings and bridges.

There are basically two kinds of wind flow models: prognostic and diagnostic. Prognostic models are largely based on solving the time-dependent conservation equations and frequently with various degrees of approximation. Diagnostic models are constructed using a limited amount of measurements, empirical relations, and certain important physical constraints. Having more complete physics, prognostic models are generally regarded as more realistic than the simpler, diagnostic models. However, models of this type are complex in nature, requiring delicate input and heavy use of computer resources. While not as sophisticated, diagnostic models generally have some of the important physics and, more importantly, require less delicate input and much less computer resources. Such models are, therefore, better-suited for emergency preparedness and response situations. Validated diagnostic models can also be highly useful in providing the necessary initial and boundary conditions for prognostic models. A recent review of diagnostic models is given by Ratto, et al. [1994].

As part of the modernization efforts of the Atmospheric Release Advisory Capability (ARAC) project (Lee, et al. [1997]), the development of a new diagnostic model for generating mass-consistent wind fields over continuous terrain was initiated recently (Chan and Sugiyama [1997]). This model, MC_WIND, will replace ARAC-2's operational code MATHEW (Sherman [1978], Sugiyama, et al. [1994]), which uses stair-step topography and constant grid spacings. The mass-adjustment algorithm of MC_WIND has also been incorporated in ADAPT (Sugiyama and Chan [1997b]), which will replace MEDIC/MATHEW operationally. The resulting wind fields from either ADAPT or MC_WIND can be used to drive ARAC's new dispersion model LODI

(Leone, et al. [1997]) and to aid the initialization and validation of prognostic models such as NORAPS (Hodur [1987]) and COAMPS (Hodur[1997]).

The present model is based on a mixed variational principle and the GFEM for spatial discretization. The finite element method is employed to effectively treat continuous terrain with variable grid resolution. In addition, the grid-point representation of the wind fields by FEM, in contrast to the flux-based, staggered grid representation often used in finite difference approaches, offers a more rigorous treatment of boundary conditions and allows the preservation of velocities at desired grid-points for cells in which wind observations are located. Other salient features of the model include map projections and a Strouhal-number-dependent parameterization scheme for differential weighting of the horizontal and vertical velocity adjustments to account for atmospheric stability effects. Two preconditioned conjugate gradient solvers are available for efficiently solving the Poisson equation resulting from the numerical formulation. Their performance is further improved by a stabilization procedure applied to the Poisson equation matrix.

In the next section, the theoretical and numerical aspects of the model are described. A user's guide for the model is given in section 3, and four numerical examples are presented in section 4 to illustrate the usage of the model.

2. NUMERICAL MODEL

In this section, the theoretical and numerical aspects of the model are described. These include the governing equations, a parameterization for atmospheric stability, spatial discretization by the finite element method, specification of boundary conditions, stabilization of the discrete Poisson equation for Lagrange multiplier, and the iterative solution for such a linear system of equations. The treatment of map projections is discussed at the end of the section.

2.1 Governing Equations

The underlying theoretical basis of our model is, with the assumption of constant density, the following functional (Sherman [1978]),

$$\begin{aligned} I(u,v,w,\lambda) = & \frac{1}{2}\int_{\Omega}\left[\alpha_H^2\left(u-u^0\right)^2+\alpha_H^2\left(v-v^0\right)^2+\alpha_V^2\left(w-w^0\right)^2\right]d\Omega \\ & +\int_{\Omega}\lambda\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)d\Omega \end{aligned} \tag{2.1}$$

In the above equation, (u^0, v^0, w^0) and (u, v, w) are the components of the initial and adjusted velocity fields, λ is the Lagrange multiplier, and α_H and α_V are the Gauss precision moduli controlling the relative amount of vertical and horizontal velocity adjustments, and Ω is the domain under consideration. Equation (2.1) is a mixed variational principle for which the solution (u, v, w, λ) is a saddle point, rather than an extremum point (Arnold [1990]). The solution corresponds to a minimum with respect to the difference between the adjusted and the initial velocity fields and a maximum with respect to the Lagrange multiplier. With this approach, the

mass-conservation requirement is enforced as a strong constraint (Sasaki [1970]). It should be noted that, for constant values of α_H and α_V, the adjustment procedure conserves vorticity.

It can be shown that the solution of (2.1), via taking $\delta I=0$, leads to the following Euler-Lagrange equations,

$$u = u^0 + \frac{1}{\alpha_H^2}\frac{\partial \lambda}{\partial x} \tag{2.2a}$$

$$v = v^0 + \frac{1}{\alpha_H^2}\frac{\partial \lambda}{\partial y} \tag{2.2b}$$

$$w = w^0 + \frac{1}{\alpha_V^2}\frac{\partial \lambda}{\partial z} \tag{2.2c}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2.3}$$

in Ω , and

$$\lambda \delta u_n = 0 \tag{2.4}$$

on the boundary.

In equation (2.4), δu_n is the first variation of the adjusted velocity in the normal direction. This equation implies that either

$$\delta u_n = 0$$

or

$$\lambda = 0.$$

The former corresponds to a boundary where the normal velocity is specified, such as $u_n = u_n^0$ on an inflow plane or $u_n = 0$ on a solid boundary, while the latter corresponds to a "flow through" boundary over which adjustment of the normal velocity is permitted.

Equations (2.2) may be substituted into the mass-conservation constraint of Equation (2.3) to yield the continuum Poisson equation

$$\frac{1}{\alpha_H^2}\left(\frac{\partial^2 \lambda}{\partial x^2} + \frac{\partial^2 \lambda}{\partial y^2}\right) + \frac{1}{\alpha_V^2}\frac{\partial^2 \lambda}{\partial z^2} = -\frac{\partial u^o}{\partial x} - \frac{\partial v^o}{\partial y} - \frac{\partial w^o}{\partial z} \tag{2.5}$$

for α_H and α_V constant throughout the domain. The ARAC-2 model MATHEW (Sherman [1978]) solves for λ using a finite-difference discretization of Equation (2.5) with appropriate boundary conditions and derives the adjusted wind fields from Equations (2.2). MC_WIND discretizes and solves equation (2.1) or its equivalent equations (2.2) through (2.4) directly (see Section 2.3).

2.2 Parameterization for Atmospheric Stability

The characteristics of atmospheric flows over terrain vary drastically under different stability conditions. Winds tend to pass over the tops of hills in neutral and unstable conditions, but are primarily steered around obstacles in stable flows. One of the ways to incorporate the effects of atmospheric stability in a diagnostic wind field model is to utilize the parameter $\alpha = \alpha_H / \alpha_V$, which is the ratio of the Gauss Precision moduli, to control the degree of adjustment in the vertical wind component relative to the horizontal wind component. In general, small values of α are appropriate for stable flows, resulting in small adjustments to the vertical velocity (which is usually zero initially), while larger values are required for unstable flows to produce larger adjustments to the vertical velocity.

The simplest approach is to use a constant value of α for the entire grid, and set $\alpha_H = 1$ so that $\alpha_V = 1/\alpha$. A value of 1.0 is appropriate for neutral stability, values smaller than 1.0 are for stable conditions and values greater than 1.0 are for unstable conditions. The recommended values for α are 5.0, 2.0, 1.0, 0.5, and 0.2 for very unstable, slightly unstable, neutral, slightly stable, and very stable conditions, respectively.

For situations involving complex topography and atmospheric stability, it is more appropriate to let α vary over the entire domain. Such an option has also been implemented in MC_WIND. Our approach follows the framework proposed by Ross, et al. [1988] and developed further by Moussiopoulos, et al. [1988], with modifications to better correlate predictions with experimental results. Specifically, an improved curve-fitting formula for α^2 as a function of local Strouhal number has been devised along with a generalized formula for determining the characteristic height difference to incorporate topographic effects.

Based on the experimental data for stably stratified flow of Hunt and Snyder [1980], the following exponential relationship between α^2 and the local Strouhal number was adopted,

$$\alpha^2 = e^{-1.5(Str)^{1.5}}, \quad Str \geq 0 \tag{2.6a}$$

$$\alpha^2 = e^{1.5(-Str)^{1.5}}, \quad Str < 0 \tag{2.6b}$$

instead of the formulae proposed by Moussiopoulos, et al.,

$$\alpha^2 = 1 - Str^4\left(\sqrt{1 + 4/Str^4} - 1\right)\Big/2, \quad Str \geq 0 \tag{2.7a}$$

$$\alpha^2 = 2\Big/\left[2 - Str^4\left(\sqrt{1 + 4/Str^4} - 1\right)\right], \quad Str < 0 \tag{2.7b}$$

Results from the above formula for stable cases are compared in figure 2.1. It is seen that the present formula provides a closer fit to the data, especially for higher Strouhal numbers.

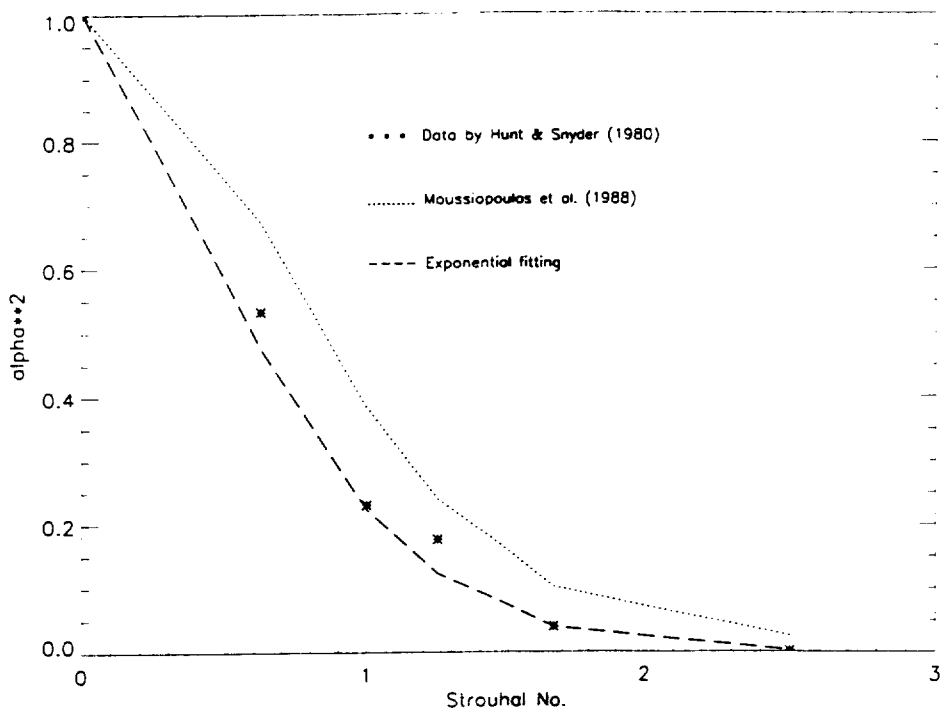


Figure 2.1. Values of α^2 as a function of Strouhal number.

The local Strouhal number is defined as

$$Str = \frac{HN}{U}, \quad N = \sqrt{\frac{g}{\theta}\frac{d\theta}{dz,}\ \frac{d\theta}{dz}} \geq 0 \tag{2.8a}$$

or

$$Str = -\frac{H}{Ut,} \quad t = \sqrt{-\frac{g}{\theta}\frac{d\theta}{dz,}\ \frac{d\theta}{dz}} < 0. \tag{2.8b}$$

In the above, H is the characteristic height difference, N is the Brunt-Väisälä frequency, U is the characteristic velocity, t is the buoyancy time scale, and θ is the potential temperature of the atmosphere.

For the characteristic wind velocity, we retain the formula proposed by Moussiopoulos, et al.

$$U = \max\left(\sqrt{\left(u^o\right)^2 + \left(v^o\right)^2},\ 0.2\,m/s\right) \tag{2.9}$$

For the characteristic height difference, the same authors proposed the following formula

$$H = \sum_{i,j}^{I,J} \left|\Delta h_{ij}\right| / r_{ij}^2 \Big/ \sum_{i,j}^{I,J} 1/r_{ij}^2 \tag{2.10}$$

where Δh_{ij} and r_{ij} are the orographic height difference and the horizontal distance between the considered location and location (i,j) on the $(I*J)$ terrain grid, respectively. An examination of (2.10) reveals that the formula lacks a provision for treating flows characterized by a constant Strouhal number. Further, the inverse distance-squared weighting appears to result in an overly rapid diminishment of topographical influence in regions of flatter terrain.

For the above reasons, a more general formula was devised and implemented, which possesses a constant term involving the difference between the maximum and minimum terrain heights and a horizontally varying term involving the inverse distance weighted height difference,

$$H = c\left(h_{\max} - h_{\min}\right) + (1-c)\sum_{i,j}^{I,J} \left|\Delta h_{ij}\right| / r_{ij} \Big/ \sum_{i,j}^{I,J} 1/r_{ij} \quad . \tag{2.11}$$

In the above formula, h_{max} and h_{min} are respectively the maximum and minimum terrain heights of the ground surface, Δh_{ij} and r_{ij} are as defined before, and c is an input parameter with value varying between 0 and 1.0 for selecting appropriate linear combination of the two terms. For problems involving only a single hill, using only the first term (with c = 1.0) is probably a good choice; for problems with more complex terrain, a linear combination of the two terms is generally more appropriate. The recommended value of c for the general problems is 0.5.

2.3 Finite Element Discretization

Finite-element spatial discretization is performed directly on Equation (2.1), rather than on the Poisson equation (2.5), and a consistently derived discrete equation is then obtained for λ. This approach is preferred because it conserves mass exactly in the *discrete* sense; discretization of the continuum Poisson equation may result in noticeable final divergences (Moussiopoulos et al. [1988]). Specifically, trilinear basis functions ϕ_i defined on a general hexahedron are used for the velocities and piecewise constant functions ψ_i are used for the field of Lagrange multiplier. With repeated indices indicating summation, these fields are approximated as

$$u = \phi_i u_i, \quad v = \phi_i v_i, \quad w = \phi_i w_i, \tag{2.12}$$

$$\lambda = \psi_i \lambda_i \tag{2.13}$$

in which there are n nodes for the velocities and m elements for the field of Lagrange multiplier. Similar approximations are used for the initial winds $\left(u^0, v^0, w^0\right)$.

Upon substituting the above approximations into (2.1) and taking δI=0, a coupled system of algebraic equations is obtained,

$$M\,U + C\lambda = M\,U^0 \tag{2.14}$$

and

$$C^T\,U = 0\ . \tag{2.15}$$

In the above equations, U is a vector containing all the nodal values of (u, v, w), U^0 is a similar vector for (u^0, v^0, w^0), and λ is a vector for the field of Lagrange multiplier, M is the mass matrix of size $3n\ x\ 3n$, C is the gradient matrix of size $3n\ x\ m$, and C^T, the transpose of C, is the divergence matrix. Equations (2.14) and (2.15) are the finite-element equivalent of the continuum equations (2.2) and (2.3). The system matrices M and C are obtained by assembling appropriately the following element matrices:

$$M_{ij}^{(k)} = \int_e \alpha^{(k)} \phi_i \phi_j d\Omega \tag{2.16}$$

$$C_{ij}^{(k)} = \int_e \frac{\partial \phi_i}{\partial x_k} \psi_j d\Omega \tag{2.17}$$

in which (k) refers to the k-th coordinate direction, $\alpha^{(k)}$ denotes the relevant value of the Gauss precision moduli, and the integral is over the element volume e.

In principle, equations (2.14) and (2.15) could be solved as a coupled system of equations. However, the cost-effectiveness of such an approach is questionable, because the system is not strictly positive definite (due to the zero coefficients on the diagonal). This fact makes pivoting necessary when solving the system by direct methods and limits the applicability of almost all iterative solution techniques. Further, for large-scale problems, direct methods are impractical because of the excessive memory requirements and possibly high CPU costs. In order to be able to use iterative methods, we adapted and implemented a cost-effective alternative developed by Gresho, et al. [1984]. By approximating the original consistent mass matrix with a lumped (via row-sum) mass matrix (denoted below as M_L), the following uncoupled system of equations is obtained,

$$C^T M_L^{-1}\, C\lambda = C^T U^0 \tag{2.18}$$

and

$$U = U^0 - M_L^{-1} C\lambda \tag{2.19}$$

in which $C^T M_L^{-1}\, C$ is a symmetric, semi-definite matrix. Eq. (2.18) is called the consistently derived discrete Poisson equation for λ and can be solved with appropriate iterative methods. Eq. (2.19) is used to obtain the nodal values of velocities for the mass-consistent wind field.

Prior to solving equations (2.18) and (2.19), modifications to these equations are necessary in order to accommodate: (1) specification of velocities including no-penetration on the ground

surface, and (2) stabilization of the discrete Poisson equation to enhance the convergence rate of the iterative solution. These topics are discussed in the next two subsections.

2.4 Specification of Boundary Conditions

Boundary conditions may be imposed by specifying velocities u, v and/or w at appropriate grid points. This is easily accomplished by placing the desired value in the initial velocity vector and setting the corresponding value of M_L^{-1} equal to zero in equations (2.18) and (2.19). It is critical that any specification of velocities and the use of boundary specifications to preserve observational input be consistent with mass conservation. A particular element needs to satisfy mass consistency in the (unlikely) event that *all* the velocities for that element are specified. Similarly, if velocities on all the simulation domain boundaries are fixed, global mass-conservation must be satisfied.

In order to impose the no-penetration condition (i.e., zero normal velocity) on the terrain surface, the surface (x, y, z) velocity components are transformed into a local, orthogonal coordinate system $(\underline{t_1}, \underline{t_2}, \underline{n})$ in which two of the directional components are tangential and one is normal to the surface. The above matrix equations can then be reestablished in terms of the transformed velocity components. Once this is done, imposition of a zero normal velocity can be performed using the method described in the preceding paragraph.

In the local coordinate system, the consistent normal direction defined by Engelman, et al. [1981] is used. With its positive direction pointing into the computational domain, the unit normal vector $\underline{n} = (n_x, n_y, n_z)$ is given by

$$n_x = -\int_\Omega \frac{\partial \phi_i}{\partial x} d\Omega \Big/ n \tag{2.20a}$$

$$n_y = -\int_\Omega \frac{\partial \phi_i}{\partial y} d\Omega \Big/ n \tag{2.20b}$$

$$n_z = -\int_\Omega \frac{\partial \phi_i}{\partial z} d\Omega \Big/ n \tag{2.20c}$$

where

$$n = \left[\left(\int_\Omega \frac{\partial \phi_i}{\partial x} d\Omega \right)^2 + \left(\int_\Omega \frac{\partial \phi_i}{\partial y} d\Omega \right)^2 + \left(\int_\Omega \frac{\partial \phi_i}{\partial z} d\Omega \right)^2 \right]^{1/2} \tag{2.21}$$

For a given normal vector, since there exists a unique tangent plane with an infinite number of possibilities for the pair of $(\underline{t_1}, \underline{t_2})$, a tangential direction has to be selected. For convenience, we choose $\underline{t_1}$ to be in the x-z plane. The new, local coordinate system is then uniquely defined by $(\underline{t_1}, \underline{t_2}, \underline{n})$ with

$$\underline{t_1} = \frac{\underline{j} \times \underline{n}}{|\underline{j} \times \underline{n}|} = \left(\frac{n_z}{n_*}, 0, -\frac{n_x}{n_*} \right) \tag{2.22}$$

$$\underline{t_2} = \underline{n} \times \underline{t_1} = \left(-\frac{n_x n_y}{n_*}, n_*, -\frac{n_y n_z}{n_*} \right) \tag{2.23}$$

$$n_* = \left(n_x^2 + n_y^2 \right)^{1/2} \tag{2.24}$$

The relationships between the velocity components in the two coordinate systems at a grid point are given by

$$\begin{Bmatrix} u_{t1} \\ u_{t2} \\ u_n \end{Bmatrix} = \begin{bmatrix} \frac{n_z}{n_*} & 0 & -\frac{n_x}{n_*} \\ -\frac{n_x n_y}{n_*} & n_* & -\frac{n_y n_z}{n_*} \\ n_x & n_y & n_z \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{2.25}$$

and

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{bmatrix} \frac{n_z}{n_*} & -\frac{n_x n_y}{n_*} & n_x \\ 0 & n_* & n_y \\ -\frac{n_x}{n_*} & -\frac{n_y n_z}{n_*} & n_z \end{bmatrix} \begin{Bmatrix} u_{t1} \\ u_{t2} \\ u_n \end{Bmatrix} \tag{2.26}$$

The gradient and divergence matrices associated with the transformed velocities are defined by

$$\hat{C} = R^T \, C \tag{2.27}$$

and

$$\hat{C}^T = C^T \, R \tag{2.28}$$

in which R is a block diagonal matrix consisting of the 3×3 transformation matrices of (2.26) for grid points on the terrain surface and the 3×3 identify matrices for the remaining grid points.

Finally, in place of equations (2.18) and (2.19), the uncoupled system of matrix equations become

$$\left[\hat{C}^T M_L^{-1}\,\hat{C}\right]\lambda = \hat{C}^T U^0 \tag{2.29}$$

$$U = U^0 - M_L^{-1}\hat{C}\lambda \tag{2.30}$$

where the velocity vector contains the normal and tangential components for all grid points on the terrain surface, thus allowing a zero normal velocity to be specified. Upon obtaining the velocities from (2.30), Eq. (2.26) is then used to transform the normal and tangential velocity components back to the components in the (x,y,z) coordinate system.

For "flow through" boundaries, such as the exit and lateral planes, the condition of $\lambda = 0$ is automatically imposed (more precisely, approximated due to discretization errors) by the present formulation.

2.5 Stabilization of the Discrete Poisson Equation

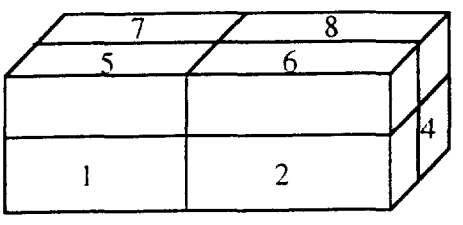
The discrete Poisson equation constructed using the popular, 'low cost' Q1-P0 finite elements for the Stokes problem is known to be unstable and spurious pressure modes could contaminate the solution (see Sani, et al. [1981]). As a result, convergence rate of the solution deteriorates as the mesh is refined (see, for example, results of Vincent and Boyer [1992]). Since the same Q1-P0 finite elements are used here, similar deficiencies are expected. Our problem is somewhat less severe, because the entries of the ill-conditioned matrix are $O(h^3)$, h being the mesh size, for the Stokes problem but are only $O(h)$ for the present problem.

To stabilize the linear system of equations (2.29) and improve the convergence rate of its solution, we adapted the 'local jump formulation' proposed by Silvester and Kechkar [1990]. In this approach, the elements are grouped into disjoint macroelements and a stabilization term is designed to make λ continuous in each macroelement. In essence, a stabilization matrix S with a tunable parameter β is added to the left-hand-side matrix of the original linear system of equations,

$$\left[\hat{C}^T M_L^{-1}\hat{C} + \beta S\right]\lambda = \hat{C}^T U^0 \tag{2.31}$$

With this approach, mass conservation is strictly enforced for each macroelement (actually accurate up to the convergence tolerance used for the iterative solution of λ) but is only approximately enforced for each individual element.

The matrix S is obtained by assembling the submatrices for the macroelements which, in general, are made up of 8 (i.e., $2\times2\times2$) adjacent elements, as shown in the following sketch, except on the boundaries where the number of elements may be less than 8 if there exists an odd number of zones in any of the coordinate directions.



Numbering of hexahedrons in a macroelement. Element No. 3 is behind No. 1 and below No. 7.

For a macroelement of 8 elements as ordered in the above sketch, the submatrix is defined below:

$$S^{(m)} = \begin{bmatrix} h_{1,2}+h_{1,3}+h_{1,5} & -h_{1,2} & -h_{1,3} & 0 & -h_{1,5} & 0 & 0 & 0 \\ -h_{2,1} & h_{2,1}+h_{2,4}+h_{2,6} & 0 & -h_{2,4} & 0 & -h_{2,6} & 0 & 0 \\ -h_{3,1} & 0 & h_{3,1}+h_{3,4}+h_{3,7} & -h_{3,4} & 0 & 0 & -h_{3,7} & 0 \\ 0 & -h_{4,2} & -h_{4,3} & h_{4,2}+h_{4,3}+h_{4,8} & 0 & 0 & 0 & -h_{4,8} \\ -h_{5,1} & 0 & 0 & 0 & h_{5,1}+h_{5,6}+h_{5,7} & -h_{5,6} & -h_{5,7} & 0 \\ 0 & -h_{6,2} & 0 & 0 & -h_{6,5} & h_{6,2}+h_{6,5}+h_{6,8} & 0 & -h_{6,8} \\ 0 & 0 & -h_{7,3} & 0 & -h_{7,5} & 0 & h_{7,3}+h_{7,5}+h_{7,8} & -h_{7,8} \\ 0 & 0 & 0 & -h_{8,4} & 0 & -h_{8,6} & -h_{8,7} & h_{8,4}+h_{8,6}+h_{8,7} \end{bmatrix} \tag{2.32}$$

in which $h_{i,j}$ is the length scale for elements 'i' and 'j' in the macroelement. The matrix $S^{(m)}$ is symmetric, since $h_{i,j} = h_{j,i}$. Two schemes are available for computing the length scale $h_{i,j}$: (1) the square root of the interior inter-element surface area between elements 'i' and 'j', or (2) the quotient of the interior inter-element surface area divided by the cube root of the average element volume of the macroelement.

Numerical experimentation shows that the optimal value for the stabilization parameter is somewhat problem dependent but is not very important. Generally speaking, a larger value of β will make the system more stable. However, too large a value can cause noticeable loss in the accuracy of the solution. A value between 0.01 and 0.1 appears to work well for most applications conducted so far.

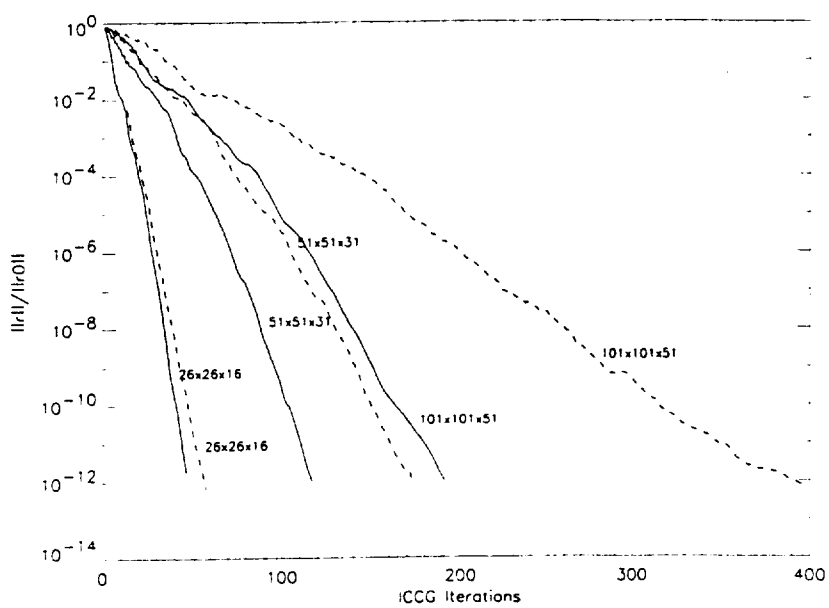


Figure 2.2. Reduction in residuals as a function of ICCG iterations from the Poisson equation solutions for a hemispheric hill using three different grids. - - - - -unstabilized, ————stabilized.

In Fig. 2.2, the reduction in the residuals during the solution of λ as a function of ICCG (defined in 2.6) iterations is shown for a hemispheric hill problem using three different grids. The current stabilization scheme is quite effective, achieving up to 50% reduction in the number of iterations with the finest grid. These results also show that, when the grid is refined by a factor of 2, the number of ICCG iterations increases by a factor of slightly over 2 for the unstabilized cases but increases only by a factor of $\sqrt{2}$ for the stabilized case using the finest grid. The final wind fields obtained with the stabilization scheme differ only by a few percent from those obtained with the unstabilized scheme.

2.6 Solution of Linear System of Equations

Conjugate gradient methods for solving an $N \times N$ linear system $Ax = b$ have several advantageous features. They require no parameter estimations and provide optimal error minimization. The algorithms have a finite termination after $M \leq N$ steps, where M is the number of distinct eigenvalues of A. Since they reference A only through vector multiplication, the codes are also highly vectorizable and allow concise storage for sparse systems.

Conjugate gradient algorithms (Varga [1962]) are based on the idea of minimizing a function $f(x) = \frac{1}{2} x^T \cdot A \cdot x - x^T \cdot b$ by finding the point where $\nabla f = A \cdot x - b = 0$. The minimization is carried out by generating a succession of search directions p_k and a corresponding set of improved solutions x_{k+1} which minimize f in the space spanned by $\{p_1, p_2, \cdots, p_k\}$. Convergence is monotonic, with the generalized error function $(Ax - z)^T \cdot (x - z)$ taking on its smallest value in the subspace spanned by the search directions when $z = x_{k+1}$ and decreasing in magnitude with each iteration.

For improved convergence, preconditioning is used to modify the matrix A to one that is very close to the identity matrix except for a few extreme eigenvalues. Preconditioning is performed by multiplying A by an easily calculable sparse symmetric matrix $\tilde{A}^{-1}$

$$\tilde{A}^{-1} A \approx I$$
$$\tilde{A}^{-1} A x = \tilde{A}^{-1} b.$$

Two preconditioned conjugate gradient solvers are available in MC_WIND for efficiently solving the linear system of equations (2.31). They are the Incomplete Choleski Conjugate Gradient (ICCG) method of Kershaw [1978] and the Diagonally Scaled Conjugate Gradient (DSCG) method.

Described below are the procedures for solving a symmetric system of equations in the form $Ax = b$, using the ICCG method,

1. Given x_0, compute

   $r_0 = b - Ax_0, p_0 = \left(LL^T\right)^{-1} r_0, z_0 = p_0$

   where $LL^T = \tilde{A}$ is the incomplete factorization matrix of A.

2. For $k = 0,1,2,....$ until convergence, compute

$$\alpha_k = (r_k,\ z_k)/(p_k,\ Ap_k)$$
$$x_{k+1} = x_k + \alpha_k\ p_k$$
$$r_{k+1} = r_k - \alpha_k\ Ap_k$$
$$z_{k+1} = \left(LL^T\right)^{-1} r_{k+1}$$
$$\beta_{k+1} = (r_{r+1},\ z_{k+1})/(r_k,\ z_k)$$
$$p_{k+1} = z_{k+1} + \beta_k p_k$$

Remarks:

(1) The following convergence criteria are used to stop the iterations:

$$\|x_{k+1} - x_k\|/\|x_{k+1}\| < \varepsilon \tag{2.33}$$

and

$$\|r_{k+1}\|/\|b\| < \varepsilon \tag{2.34}$$

where ε is the convergence tolerance.

(2) For DSCG, replace $\left(LL^T\right)^{-1}$ with D^{-1}, of which D is the diagonal vector of A.

2.7 Map Projections

Calculations are generally performed in a conformal map projection coordinate system, which preserves the angles between intersecting curves. The relationships between the map projection coordinates (X,Y,Z) and the Cartesian coordinates (x,y,z) can be written as

$$X = mx, \quad Y = my, \quad Z = z \tag{2.35}$$

where $m(x,y)$ is the map length scaling factor. Similarly, the velocities are related by

$$U = mu, \quad V = mv, \quad W = w \tag{2.36}$$

The continuity equation, written for the mapped coordinate system, takes the form (see Haltiner and Williams [1980]),

$$m^2\left[\frac{\partial(U/m)}{\partial X} + \frac{\partial(V/m)}{\partial Y}\right] + \frac{\partial W}{\partial Z} = 0 \tag{2.37}$$

The mixed variational principle in the map projection coordinates is,

$$I(U,V,W,\lambda)=\frac{1}{2}\int_{\Omega}\left[\alpha_H^2\left(\frac{U-U^0}{m}\right)^2+\alpha_H^2\left(\frac{V-V^0}{m}\right)^2+\alpha_V^2\left(W-W^0\right)^2\right]\frac{1}{m^2}dXdYdZ$$
$$+\int_{\Omega}\lambda\left\{m^2\left[\frac{\partial(U/m)}{\partial X}+\frac{\partial(V/m)}{\partial Y}\right]+\frac{\partial W}{\partial Z}\right\}\frac{1}{m^2}dXdYdZ \tag{2.38}$$

For numerical computations, it is advantageous to directly compute the values of U/m, V/m, and W/m^2 at the grid points. To do so, we recast (2.38) as

$$I\left(\frac{U}{m},\frac{V}{m},\frac{W}{m^2},\lambda\right)=\frac{1}{2}\int_{\Omega}\left[\frac{\alpha_H^2}{m^2}\left(\frac{U-U^0}{m}\right)^2+\frac{\alpha_H^2}{m^2}\left(\frac{V-V^0}{m}\right)^2+\alpha_V^2 m^2\left(\frac{W-W^0}{m^2}\right)^2\right]dXdYdZ$$
$$+\int_{\Omega}\lambda\left[\frac{\partial(U/m)}{\partial X}+\frac{\partial(V/m)}{\partial Y}+\frac{\partial\left(W/m^2\right)}{\partial Z}\right]dXdYdZ \tag{2.39}$$

with the associated Euler-Lagrange equations

$$\frac{U}{m}=\frac{U^0}{m}+\frac{m^2}{\alpha_H^2}\frac{\partial\lambda}{\partial X} \tag{2.40a}$$

$$\frac{V}{m}=\frac{V^0}{m}+\frac{m^2}{\alpha_H^2}\frac{\partial\lambda}{\partial Y} \tag{2.40b}$$

$$\frac{W}{m^2}=\frac{W^0}{m^2}+\frac{1}{\alpha_H^2 m^2}\frac{\partial\lambda}{\partial Z} \tag{2.40c}$$

and

$$\frac{\partial(U/m)}{\partial X}+\frac{\partial(V/m)}{\partial Y}+\frac{1}{m^2}\frac{\partial W}{\partial Z}=0 \tag{2.41}$$

The same numerical procedures described previously can then be applied to these equations by substituting U/m, V/m, W/m^2 for u,v,w and incorporating the map scaling factor m into the lumped mass matrix M_L.

3. USER'S GUIDE

In this section, the namelist groups used by MC_WIND are described. For easy reference, they are also summarized in tables. Finally, a brief description is presented for the input and output files used in the code.

3.1. The Namelist Groups

There are five namelist groups for MC_WIND: mc_idfiles, mc_hills_xyz, mc_hills_ics, adapt_mc_adjust, and adapt_mc_bc. Of the five namelist groups, mc_idfiles, adapt_mc_adjust, and adapt_mc_bc are more commonly used; mc_hills_xyz and mc_hills_ics have been designed specifically for the hemispheric hill and polynomial hill problems and are required only when one chooses to generate the grids and the initial winds internally. The namelist groups adapt_mc_adjust and adapt_mc_bc are also used by ADAPT (Sugiyama and Chan [1997b]), when mass-consistent wind fields are desired in the run. Default values for the variables are given in square brackets.

3.1.1 Namelist group 'mc_idfiles'

This namelist group is used in MC_WIND to define the problem, grid dimensions, and the names of input/output files. The variables in this group are:

title	
imax	
jmax	
kmax	
grid_file	[grid_nc]
wind_file	[wind_nc]
outgtx_file	[mc_wind_nc]

The variable 'title' is a string of 80 characters for problem identification, 'imax', 'jmax', and 'kmax' are the number of grid points in the x, y, and z coordinate directions, respectively, 'grid_file' and 'wind_file' are the names of input files containing the nodal coordinates generated by GRIDGEN and the initial winds generated by ADAPT, and 'outgtx_file' is a netCDF output file containing the mass-consistent wind field, which is used as input to the dispersion code LODI.

3.1.2 Namelist group 'mc_hills_xyz'

MC_WIND has some capabilities for generating the grids internally for problems involving simple or analytic geometric shapes, such as the hemispheric and polynomial hill problems. The following variables are used to generate such grids:

xmin	xmax
ymin	ymax
zmin	zmax
x0	y0 z0
radius	height
ihmin[0]	ihmax[0]
jhmin[0]	jhmax[0]
ggf	

In the above list, the first six variables specify the extent of the computational domain, while the next three variables define the location of the hill center. The variable 'radius' is the radius parameter for a hemispheric hill, while 'height' is the height parameter for the axisymmetric polynomial hill problem studied by Hunt and Snyder [1980]. 'ihmin', 'ihmax', 'jhmin', and 'jhmax' are optional inputs for the hemispheric hill problem used in rearranging the horizontal grid

locations to better represent the circular base of the hill and the steep gradients near the base without requiring extra grid points. Their values define the boundary of a square, whose four corner points are near the circular base of the hill and are used as control points in the remeshing process. The variable 'ggf' is a geometric grading factor to be used in generating a graded mesh in the vertical direction. Its value is, in general, dictated by the desired grid spacings near the ground and the number of grid points used in the vertical direction. However, for solution accuracy, a value between 1.0 and 1.2 is recommended.

3.1.3 Namelist group 'mc_hills_ics'

In MC_WIND, constant initial winds and a temperature field with a linear profile can be generated with the following variables:

u0	[1.0]
v0	[0.0]
w0	[0.0]
t0	[300.0]
dtdz	[0.0]

The variables 'u0', 'v0', and 'w0' are the Cartesian velocity components, and 't0', 'dtdz' are the parameters for defining a linear potential temperature profile. This namelist can only be used for the hemispheric and polynomial hill problems.

3.1.4 Namelist group 'adapt_mc_adjust'

The variables of this namelist group are used by both MC_WIND and ADAPT to define the various parameters required by the mass-adjustment algorithm. Based on their functionality, the variables can be divided into three subgroups:

flag_vgpm	['off']
alpha_h	[1.0]
alpha_v	[1.0]
cfhmax	[0.5]
grav	[9.806]
itdpts	[0]
file_tprof	
nbc_specified	[0]
cg_solver	['iccg']
epsilon	[0.001]
itmax	[400]
istab	[2]
cstab	[0.01]
prt_level	[-1]

Variables in the first subgroup pertain to atmospheric stability and boundary conditions, which are used in constructing the Poisson equation matrix for the field of Lagrange multiplier. Specifically, the first seven variables are related to the evaluation of the Gauss precision moduli

(GPM), the weighting factors for controlling the adjustment of vertical velocity relative to horizontal velocity, and the last variable is the number of boundary condition namelists 'adapt_mc_bc' present in the input file.

In the present codes, the field of GPM can be either constant or variable. To use constant GPM, 'flag_vgpm' is set to 'off' and the values 'alpha_h' and 'alpha_v' are used. The value for 'alpha_h' is usually set equal to 1.0, and the recommended values for 'alpha_v' are 0.2, 0.5, 1.0, 2.0, and 5.0 for very unstable, slightly unstable, neutral, slightly stable, and very stable atmospheric conditions, respectively. If variable GPM are desired, 'flag_vgpm' is set to 'on' and the values for 'cfhmax', 'grav', 'itdpts', and 'file_tprof' must be given. The variable 'cfhmax' is a floating-point number between [0,1], which specifies the fraction of the maximum and minimum terrain difference to be used in the characteristic height difference formula (2.11), 'grav' is the constant of gravitational acceleration, 'itdpts' is the number of temperature data points to be extracted from the upper air sounding file, and 'file_tprof' specifies the name of that file.

The next five variables are input parameters to be used in solving the linear system of equations for the field of Lagrange multiplier. The variables 'cg_solver', 'epsilon', and 'itmax' are associated with the iterative solver and 'istab' and 'cstab' are related to the stabilization schemes. In the present codes, either the 'iccg' (incomplete Choleski conjugate gradient) or 'dscg' (diagonally scaled conjugate gradient) solver can be used. 'epsilon' is a convergence tolerance for the iterative solution, and 'itmax' is the maximum number of iterations allowed. In order to enhance the convergence rate of the iterative solution, two slightly different stabilization schemes, each involving the addition of a small, block diagonal stabilization matrix to filter out the unstable modes in the solution, have been implemented. The desired scheme is selected by the value of 'istab' and the weight of the stabilization matrix (β in Eq. (2.3)) is specified by 'cstab'. The stabilization scheme is activated with either istab=1 (for which the stabilization matrix is built with length scales based on the inter-element areas and element volumes) or istab=2 (for which the stabilization matrix is built with length scales based on the inter-element areas only). The recommended value for 'cstab' is 0.01, but larger values (on the order of 0.1 or 0.2) may sometimes be needed for problems having very ill-conditioned matrices, such as those involving rough terrain and/or very stable atmospheric conditions.

The variable 'prt_level' is a parameter for controlling the level of output to a file called mc_wind.sol. For most problems, a value of '-1' (default) is adequate and the output will include an echo of the input parameters and certain global results sampled before and after mass-adjustment. A value of '0' will put out, in addition, the nodal values of the coordinates, map factors, the field of the Lagrange multiplier, potential temperatures, and the initial and adjusted wind fields. The fields from this file can be processed and postprocessed by a graphics code named GRIZ. Values higher than '0' are for debugging purposes only. Specifically, a value of '1' will produce additional output, including grid points with specified velocities, the values of direction cosines for grid points on the ground surface, and the right-hand-side (RHS) vector of the linear system of equations. A value of '2' will also print out entries of the linear system of equations, the stabilization matrix, and the incomplete Choleski decomposed matrix.

3.1.5 Namelist group 'adapt_mc_bc'

The variables in this namelist group have been designed to specify or hold velocity components at selected grid points, which are usually on the domain boundaries, from being altered by the mass adjustment process. Some examples are the specification of velocities on the inlet plane, no-penetration on the ground surface, and the preservation of interpolated data at desired grid points within the grid. For convenience, the imposition of no-penetration on the ground surface has been hard-wired in the code so that no input is required.

Other constraints can be imposed by using various combinations of the following variables:

```
kind_bc
value_bc
i1  j1  k1
i2  j2  k2
```

The character variable 'kind_bc' denotes the action to be taken, 'value_bc' is the specified value for the selected velocity component, and the six integer variables, 'i1', 'i2', 'j1', 'j2', 'k1', and 'k2', are the (i,j,k) indices of two control points defining the group of grid points for which the desired specification is to take place. Depending on the values of the indices, the two control points can define a group of grid points forming a 3-D block of cells (with the control points placed at the opposite corners of the block), a plane (with the control points placed at the opposite corners of the plane), or a line (with the control points placed at the ends of the line).

The entry of 'kind_bc' can be one of the following:

```
hold_i1uvw
hold_u   hold_v   hold_w
set_u    set_v    set_w
```

The key word 'hold_i1uvw' directs the code to hold all the velocity components on the i=1 (inlet) plane to the initial values. The key words 'hold_u', 'hold_v', and 'hold_w' hold the designated velocity components to their initial values for the locations defined by the control points. The key words 'set_u', 'set_v', and 'set_w' are used similarly except that the variable 'value_bc' is used to specify a value different from that in the initial wind field.

This namelist group can be used as many times as necessary in both ADAPT and MC_WIND.

3.2. Tables of Namelist Groups

For easy reference, the namelist groups are summarized in the following tables.

Table 3.1. Namelist MC_IDFILES

Variable	Type	Description	Default
grid_file	char*160	name of grid input file generated by GRIDGEN	'grid_nc'
imax	integer	number of grid points in the x-direction	
jmax	integer	number of grid points in the y-direction	
kmax	integer	number of grid points in the z-direction	
title	char*80	character string for problem identification	
wind_file	char*160	name of wind input file generated by ADAPT	'wind_nc'
outgtx_file	char*160	name of netCDF output file containing grid point coordinates and the wind fields	'mc_wind_nc'

Table 3.2. Namelist MC_HILLS_XYZ

Variable	Type	Description	Default
ggf	real	geometric grading factor for grid spacings in the vertical direction	
height	real	height parameter for the polynomial hill problem	
ihmax	integer	i-index of the right side of the square used in remeshing the hemispheric hill problem	0
ihmin	integer	i-index of the left side of the square used in remeshing the hemispheric hill problem	0
jhmax	integer	j-index of the upper side of the square used in remeshing the hemispheric hill problem	0
jhmin	integer	j-index of the lower side of the square used in remeshing the hemispheric hill problem	0
radius	real	radius parameter for a hemispherical hill	
xmin, xmax	real	the minimum and maximum x-coordinates of the computational domain	
ymin, ymax	real	the minimum and maximum y-coordinates of the computational domain	
zmin, zmax	real	the minimum and maximum z-coordinates of the computational domain	
x0, y0, z0	real	location of hill center	

Table 3.3. Namelist MC_HILLS_ICS

Variable	Type	Description	Default
dtdz	real	slope of potential temperature profile	0.0
t0	real	value of potential temperature at z=0	300
u0	real	constant initial wind component in the x-direction	1.0
v0	real	constant initial wind component in the y-direction	0.0
w0	real	constant initial wind component in the z-direction	0.0

Table 3.4. Namelist ADAPT_MC_ADJUST

Variable	Type	Description	Default
alpha_h	real	weighting factor for horizontal velocity adjustment	1.0
alpha_v	real	weighting factor for vertical velocity adjustment, with the following recommended values: 0.2, very unstable 0.5, slightly unstable 1.0, neutral 2.0, slightly stable 5.0, very stable	1.0
cfhmax	real	fraction of maximum terrain height difference to be used in the formula for the characteristic height difference	0.5
cg_solver	char*24	conjugate gradient solver 'iccg'=incomplete Choleski conjugate gradient 'dscg'=diagonally scaled conjugate gradient	'iccg'
cstab	real	weighting factor for the stabilization matrix	0.01
epsilon	real	convergence tolerance for the iterative solution	0.001
file_tprof	char*160	name of upper air sounding file	
flag_vgpm	char*24	flag for variable Gauss precision moduli 'off'=constant Gauss precision moduli 'on' =variable Gauss precision moduli	'off'
grav	real	constant of gravitational acceleration	9.806
istab	integer	stabilization matrix selector 0: no stabilization matrix 1: stabilization matrix using length scales based on inter-element surface areas and element volumes 2: stabilization matrix using length scales based on inter-element surface areas only	2
itdpts	integer	number of temperature data points to be extracted from upper air sounding file	0
itmax	integer	maximum number of iterations allowed for the conjugate gradient solver	400
nbc_specified	integer	number of boundary condition namelist groups	0
prt_level	integer	level of ASCII output to mc_wind.sol -1: input parameters and sample global results 0: plus coordinates, map factors, solution vectors, initial and adjusted winds 1: plus specified velocities and their locations, values of direction cosines on the ground surface, and the RHS vector of Eq. (2.31) 2: plus entries of the original matrix, the stabilization matrix, and the incomplete Choleski decomposed matrix	-1

Table 3.5. Namelist ADAPT_MC_BC

Variable	Type	Description
i1	integer	i-index of first control point of a line, a plane, or a 3-D block for which the wind component is to be held or set
i2	integer	i-index of second control point of a line, a plane, or a 3-D block for which the wind component is to be held or set
j1	integer	j-index of first control point of a line, a plane, or a 3-D block for which the wind component is to be held or set
j2	integer	j-index of second control point of a line, a plane, or a 3-D block for which the wind component is to be held or set
k1	integer	k-index of first control point of a line, a plane, or a 3-D block for which the wind component is to be held or set
k2	integer	k-index of second control point of a line, a plane, or a 3-D block for which the wind component is to be held or set
kind_bc	char*24	type of boundary conditions 'hold_i1uvw': hold all velocity components on the i=1 plane to their initial values 'hold_u': hold u-velocity component to the initial values for grid points defined by (i1,j1,k1) and (i2,j2,k2) 'hold_v': hold v-velocity component to the initial values for grid points defined by (i1,j1,k1) and (i2,j2,k2) 'hold_w': hold w-velocity component to the initial values for grid points defined by (i1,j1,k1) and (i2,j2,k2) 'set_u': set u-velocity component to input value, value_bc, for grid points defined by (i1,j1,k1,) and (i2,j2,k2) 'set_v': set v-velocity component to input value, value_bc, for grid points defined by (i1,j1,k1,) and (i2,j2,k2) 'set_w': set w-velocity component to input value, value_bc, for grid points defined by (i1,j1,k1) and (i2,j2,k2)
value_bc	real	specified wind component value, if kind_bc is 'set_u', 'set_v', or 'set_w'

3.3 Input and Output Files

The primary input to MC_WIND is a namelist file called mc_wind.nml, which is usually supplemented by file(s) containing the grid-point coordinates and the initial winds. In addition, when space-dependent weightings for horizontal and vertical velocity adjustment are desired, an upper air sounding file for the temperature profile is also required. However, for the hemispheric hill and polynomial hill problems, an option exists to obtain the required temperature field from the initial conditions generated by 'mc_hills_ics', thus no upper air sounding file is required. The input files are described in the following table.

Table 3.6. MC_WIND input files

File name	Unit No.	Description	Default
mc_wind.nml	5	A namelist file used to define the problem, grid dimensions, names of files containing grid-point locations and the initial winds, parameters for the mass-adjustment algorithm, and specified velocity components, etc. Details of the namelists are given in sections 3.1 and 3.2	
user supplied	?	A netCDF file generated by GRIDGEN defining the grid-point locations.	'grid_nc'
user supplied	?	A netCDF file generated by ADAPT defining the interpolated/extrapolated initial winds.	'wind_nc'
user supplied	11	An ASCII formatted file defining the upper air sounding of the temperature profile, which is required when flag_vgpm='on' and itdpts>0. Data are read beginning at line #11 with format (a10,f5.0,a34,f5.1).	

? Unit number assigned by the system at run time.

In addition to screen output, output files are generated by MC_WIND for storing error messages, a log of the run, the initial and adjusted wind fields and temperatures for post-processing. These files are described in the following table.

Table 3.7. MC_WIND output files

File Name	Unit No.	Description	Default
adapt.err	2	An ASCII file for error messages.	
mc_wind.log	3	A log file containing input parameters, certain monitored data such as minimum and maximum velocities, minimum and maximum divergence, convergence information of the iterative solution, and timing information for various tasks.	
mc_wind.sol	10	An ASCII file containing data similar to the log file except the convergence information. In addition, depending on the input value of prt_level (see table 3.4), various level of output for post-processing by GRIZ and for debugging purposes may also be included.	
user supplied	?	A netCDF file containing grid-point locations, the initial and adjusted winds for post- processing by GTX.	'mc_wind_nc'

? Unit number assigned by the system at run time.

4. NUMERICAL EXAMPLES

In this section, four examples are presented to demonstrate the usage and some of the features available in the MC_WIND model. The examples are: potential flow over a hemispheric hill, stably stratified flow around an axisymmetric polynomial hill, an application for the San Francisco Bay Area, and an adjusted wind flow for the Los Alamos National Laboratory area.

4.1 Potential Flow Over a Hemispheric Hill

To test the accuracy of the present model, the potential flow around a hemispheric hill is simulated. The solution for this problem, in terms of the velocity potential, is

$$\Phi = u_o x\left(1 + r_o^3/2r^3\right), \quad r \geq r_o$$

in which u_o is the free stream velocity and r_0 is the radius of the hill. The corresponding velocity components are

$$u = u_o\left[1 + \left(y^2 + z^2 - 2x^2\right) r_o^3/2r^5\right]$$
$$v = -3u_o xy \;\; r_o^3/2r^5$$
$$w = -3u_o xz \;\; r_o^3/2r^5$$

Listed below is the namelist file for the problem:

```
&mc_idfiles
    title = 'Potential flow over a hemispheric hill'
    imax   =  81
    jmax   =  81
    kmax   =  51
    grid_file    = 'hemispheric_hill'
    outgtx_file  = 'mc_wind_nc'
    /
&mc_hills_xyz
    xmin   =  -2500.0      xmax   =  2500.0
    ymin   =  -2500.0      ymax   =  2500.0
    zmin   =  0.0000       zmax   =  2500.0
    x0     =  0.0          y0     =  0.0      z0  =  0.0
    ihmin  =  31           ihmax  =  51
    jhmin  =  31           jhmax  =  51
    radius =  1000.0
    ggf    =  1.02
    /
&mc_hills_ics
    u0     =  1.00
    v0     =  0.00
    w0     =  0.00
    /
&adapt_mc_adjust
    istab          =  2
    cstab          =  0.1
```

```
    prt_level        =  0
    nbc_specified    =  1
    /
&adapt_mc_bc
    kind_bc = 'hold_iluvw'
    /
```

In this numerical simulation, a computational domain of $81 \times 81 \times 51$ grid points is used. The entry of grid_file directs the code to generate the grid and initial wind field internally, with the parameters supplied by namelists mc_hills_xyz and mc_hills_ics. The computational domain is 5000m x 5000m x 2500m, with a hemispheric hill of radius 1000m placed at the center of the grid. A geometric grading factor of 1.02 is used to generate the vertical grid, with a minimum grid spacing of 17.74m on the hill top. The flexibility of the finite element method is utilized in the horizontal directions, through the input of ihmin, ihmax, jhmin, and jhmax, to obtain a closer approximation of the circular base of the hill and the steep gradients nearby. The default choice of constant Gauss precision moduli and the ICCG solver are used in this example. As indicated by the input value of prt_level, nodal coordinates and the solution, including the initial and adjusted flow field, etc., will be written into mc_wind.sol for post-processing. The velocity components on the inlet plane are held to their initial values, as requested by nbc_specified=1 and the namelist group adapt_mc_bc at the end of the file.

Some sample results are shown in figures 4.1 and 4.2. In Fig. 4.1, the predicted velocity vectors on the center plane and the 100-m horizontal plane are displayed. The wind field, as expected, exhibits the stagnation points at the front and the rear of the hill and shows the potential flow following the surface of the hill. In Fig. 4.2, the predicted velocity profile above the hill crest is compared with the exact potential flow solution. The agreement is excellent, with a predicted maximum speed of 1.54 versus the theoretical value of 1.50.

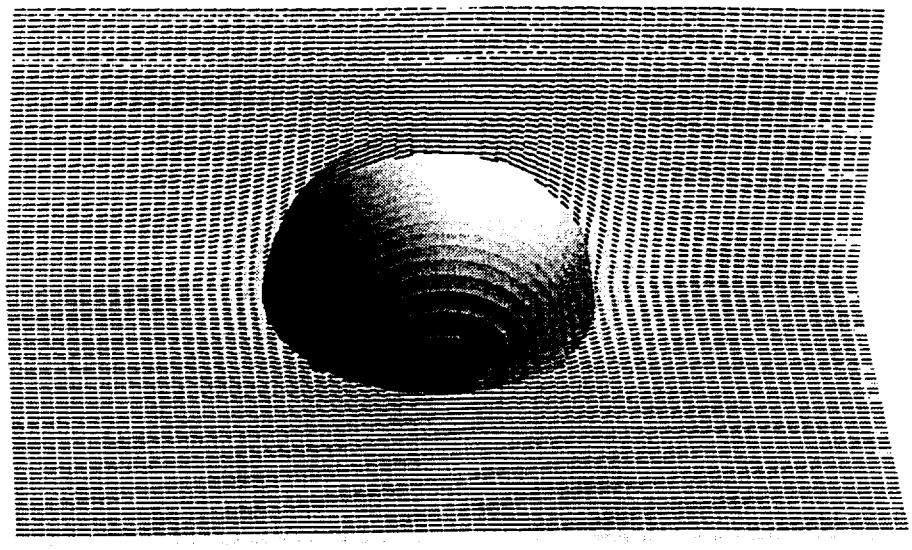


Figure 4.1. Predicted velocity vectors on z=100m and the center plane for a hemispheric hill using a 81x81x51 grid.

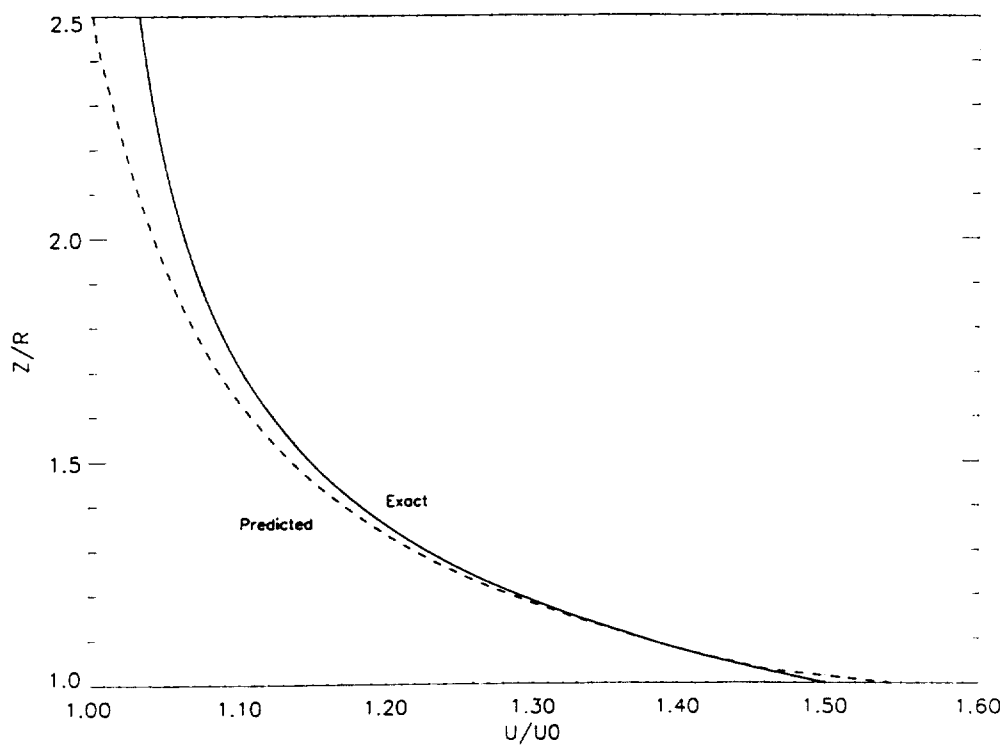


Figure 4.2. Comparison of predicted versus exact u-velocity profile above the crest of a hemispheric hill using a 81x81x51 grid.

4.2 Stably Stratified Flow Around a Polynomial Hill

In this example, a mass-consistent wind over an axisymmetric polynomial hill under stable atmospheric conditions (Strouhal number=1) is generated. This is a simulation for one of the laboratory experiments conducted by Hunt and Snyder [1980], whose results have been used for parameterizing the stability effects in the diagnostic wind models by Moussiopoulos, et al. [1988] and Ross, et al. [1988] . The following is the input file necessary for the simulation.

```
&mc_idfiles
   title='Stably stratified flow around a polynomial hill, Str=1'
   imax   =  51
   jmax   =  51
   kmax   =  31
   grid_file    = 'polynomial_hill'
   outgtx_file  = 'mc_wind_nc'
   /
&mc_hills_xyz
   xmin   =  -100.0     xmax  =  100.0
   ymin   =  -100.0     ymax  =  100.0
   zmin   =  0.00       zmax  =  50.0
   x0     =  0.0        y0    =  0.0       z0  =  0.0
   height =  22.9       ggf   =  1.05
   /
&mc_hills_ics
   u0     =  1.0        v0    =  0.0       w0  =  0.0
   t0     =  300.0      dtdz  =  0.05535
   /
&adapt_mc_adjust
   flag_vgpm    =  'on'
   cfhmax       =  1.0
   grav         =  9.806
   nbc_specified  =  1
   prt_level      =  -1
   /
&adapt_mc_bc
   kind_bc      =  'hold_i1uvw'
   /
```

This computational mesh has 51x51x31 grid points. The grid_file entry directs the code to generate the grid and initial wind field internally. The size of the computational domain is 200m x 200m x 50m. The polynomial hill, with a height of 22.9m, is placed at the center of the grid. A geometric grading factor of 1.05 is used to yield a minimum vertical grid spacing of 0.408m above the hill top. To account for the desired level of stable stratification, the variable GPM flag is turned 'on' and a linear temperature profile is defined by the values of t0 and dtdz based on equation (2.8a) (thus no upper air sounding file is required). On the inlet plane, the velocities are held to their initial values of u=1 and v=w=0.

In Fig. 4.3, the predicted stream function on the center plane is shown, together with three streamlines beginning upstream at the heights z/h = 0.5, 1.0, 1.5, where h is the height of the hill.

In Fig. 4.4, a close-up view of the velocity projection and the (shaded) w-component on the plane at one-half of the hill height are displayed. These figures clearly show how the flow 'climbs' up and 'channels' itself around the hill. Due to the fact that the velocity adjustments are based on potential flow theory, the observed wake effects of separations and horseshoe vortices behind the hill are absent in the numerical predictions. The overall flow patterns are, however, quite consistent with those observed by Hunt and Snyder [1980]. The streamline beginning at z=h is predicted to be displaced 0.3h above the hill crest (see Fig. 4.3) versus 0.21h as observed in the laboratory experiment.

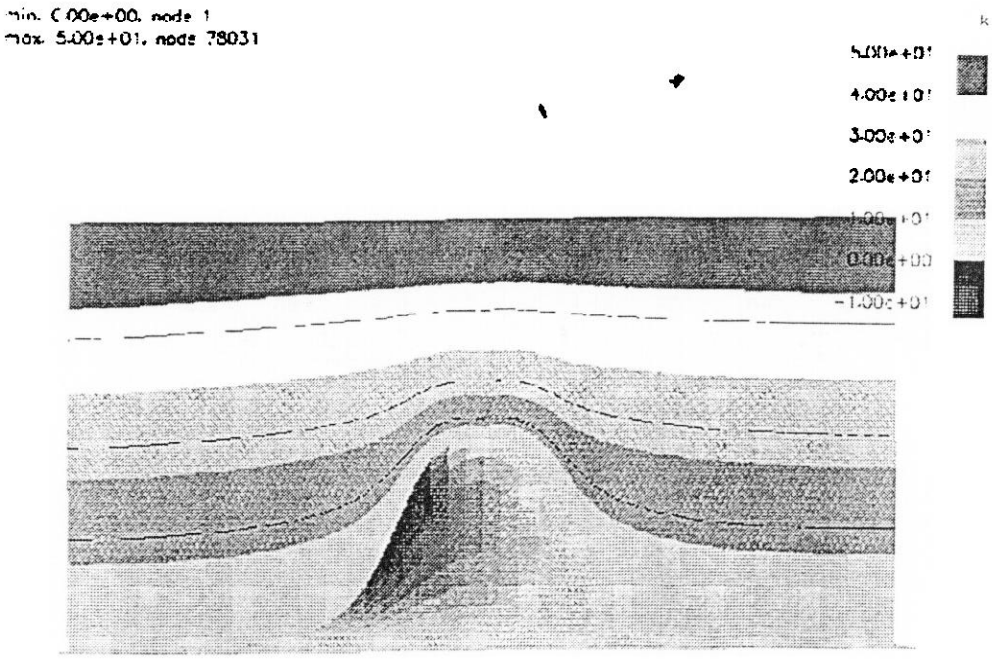


Figure 4.3. Predicted stream function and three streamlines from upstream at z/h = 0.5, 1.0, and 1.5 on the center plane of an axisymmetric polynomial hill under stable conditions with Strouhal number = 1.0.

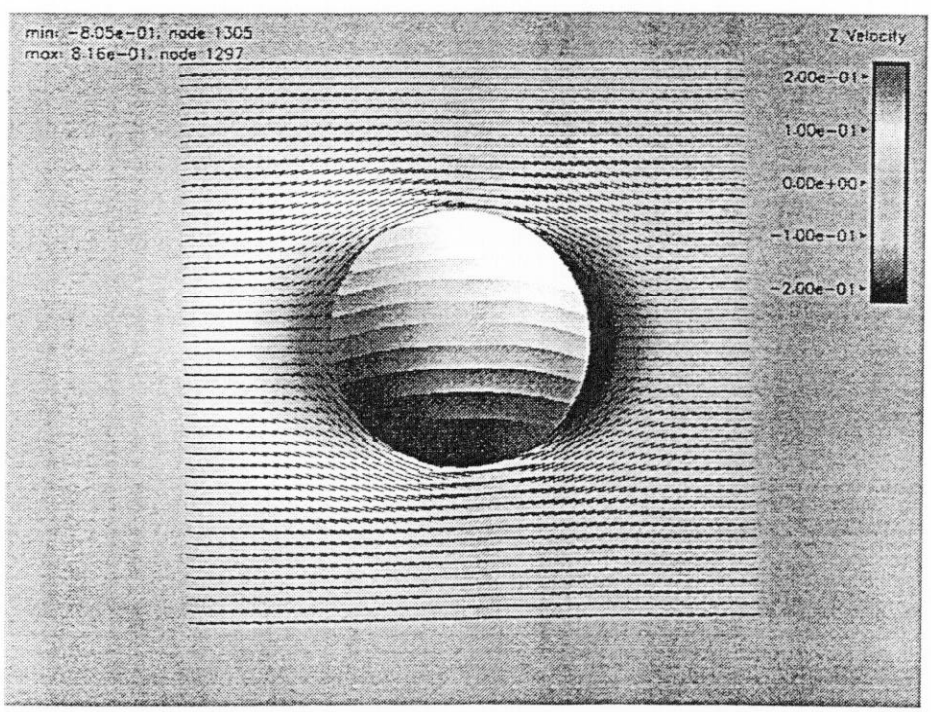


Figure 4.4 Predicted wind vectors and the w-component (in shade) on the plane of one-half hill height for an axisymmetric polynomial hill under stable conditions with Strouhal number = 1.0.

4.3 Application for the San Francisco Bay Area

In this example, MC_WIND is applied to generate the windfields for an emergency response scenario in the San Francisco Bay Area, using interpolated wind data from the new ARAC meteorological database (Sugiyama and Chan [1997a]). The grid is 100km x 100km x 3km, with 81 grid points in each horizontal direction and 31 grid points in the vertical direction. The grid has 10m vertical resolution near the ground surface. The namelist file used in the simulation is listed below, with default values used for the Gauss precision moduli and the ICCG iteative solver.

```
&mc_idfiles
   title     =  'San Francisco Bay Area, 81x81x31 grid'
   imax      =  81
   jmax      =  81
   kmax      =  31
   grid_file    = 'sigmaz___1____100___569000__4176000_001_grd.nc'
   wind_file    = 'magic_oak_19960709_220000.nc'
   outgtx_file  = 'mc_wind_19960709_220000g.nc'
   /
&adapt_mc_adjust
   istab     =  2
   cstab     =  0.1
   prt_level    = -1
   /
```

Figs. 4.5 shows the mass-consistent wind vectors on the 10 m AGL plane, together with the surface observations (plotted within the circles). In general, only small changes were made to the initial winds by the mass-adjustment model. In Fig. 4.6, wind vectors on the y-z plane in the middle of the grid are shown, with the vertical wind components exaggerated by a factor of 10. The variable grid employed and the terrain following adjusted velocity vectors can be seen in this figure. The magnitude of the vertical wind induced by the topography is not very large, which is consistent with the small changes observed for the horizontal winds.

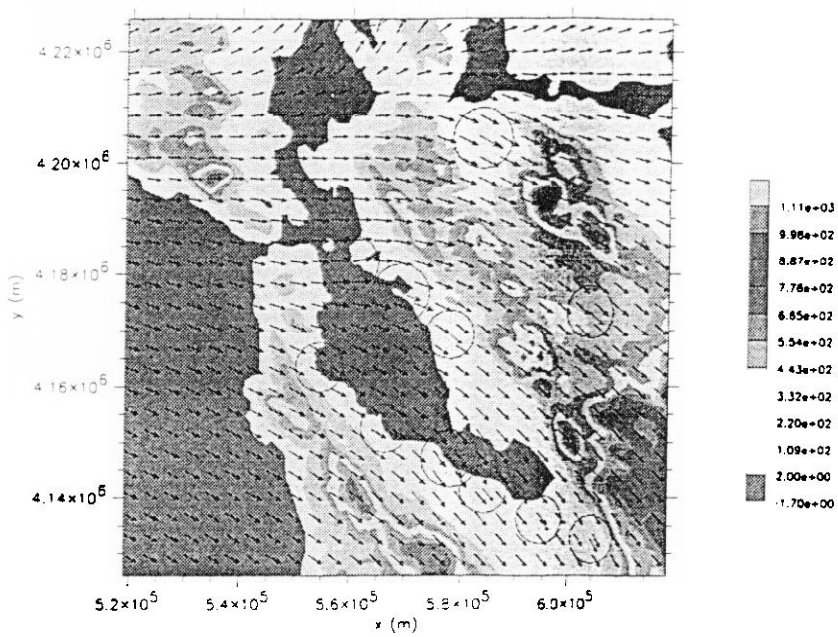
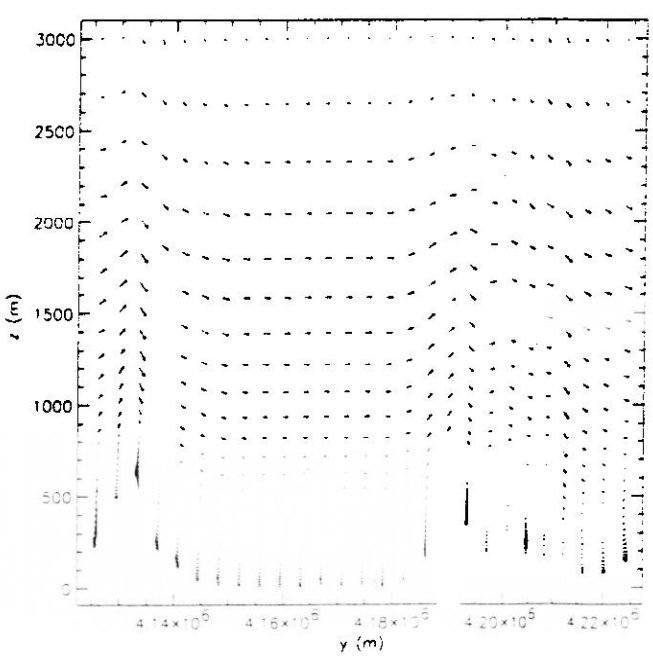


Figure 4.5. Mass-consistent wind field at 10m AGL, with surface wind observations plotted in circles.



(VW) Largest Vector = 8.484e+00

Figure 4.6. Adjusted wind vectors on the crosswind plane at x = 570 km.

4.4 Wind Flow in the LANL Area

In this example, MC_WIND is applied to generate a mass-consistent wind field in the Los Alamos National Laboratory area, where terrain effects are very important. The grid is 20km x 20km x 4km, with 51 grid points in each horizontal direction and 31 grid points in the vertical direction. The grid has 10 m vertical resolution near the ground surface. The namelist file used in the simulation is listed below. The default values of constant Gauss precision moduli for a neutral atmosphere and the ICCG iteative solver have been used in the present simulation. Except for the default of no-panetration on the ground surface, no other boundary condition is specified.

```
&mc_idfiles
   title     =  'LANL_BB 1994/03/02 02:15:00 51x51x31'
   imax      =  51
   jmax      =  51
   kmax      =  31
   grid_file    = 'sigmaz___1_____20___381380__3969170_001_grd.nc'
   wind_file    = 'magic_lanl_nc'
   outgtx_file  = 'mc_wind_nc'
   /
&adapt_mc_adjust
   nbc_specified  =  0
   /
```

Figs. 4.7 and 4.8 show the interpolated and adjusted wind vectors on the $\sigma_z = 0.0124$ plane (11m to 30m AGL). There is a good agreement between the interpolated and adjusted winds except in regions of extreme terrain where, as expected, much larger adjustments are observed. In Figs. 4.9 and 4.10, the interpolated wind vectors on the 23m AGL plane from MEDIC and the adjusted wind vectors on the same plane from the MATHEW code of ARAC-2, which uses a constant vertical grid of 23 m and has the top boundary at 700 m, are displayed. As a result of its stair-step terrain and the inflexible uniform grid representation, the MATHEW model has produced much slower wind speeds and has greatly distorted the wind directions across almost the entire region near the surface.

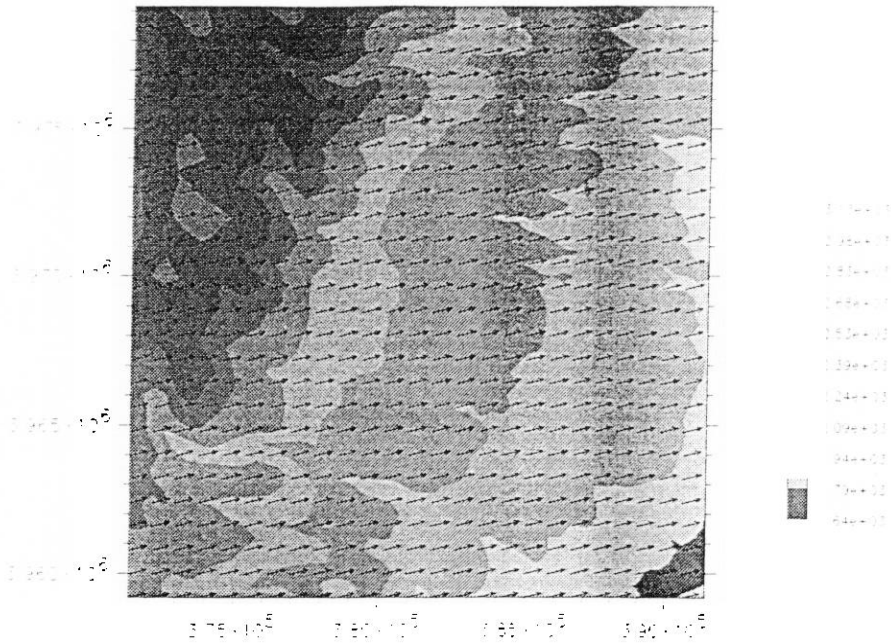


Figure 4.7 Interpolated wind vectors on the $\sigma_z = 0.0124$ plane (11m to 30m AGL)for the LANL area.

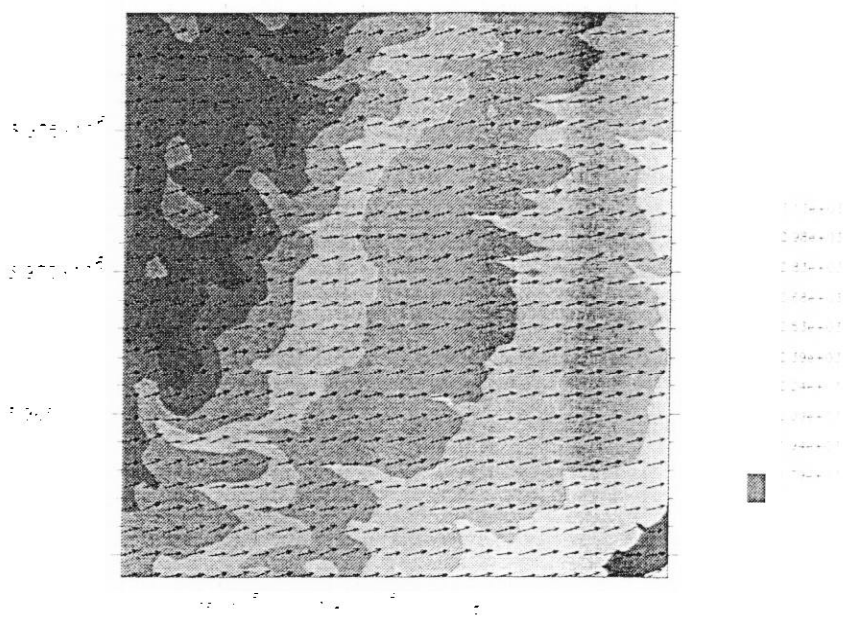


Figure 4.8. Adjusted wind vectors on the $\sigma_z = 0.0124$ plane (11m to 30m AGL) for the LANL area from MC_WIND.

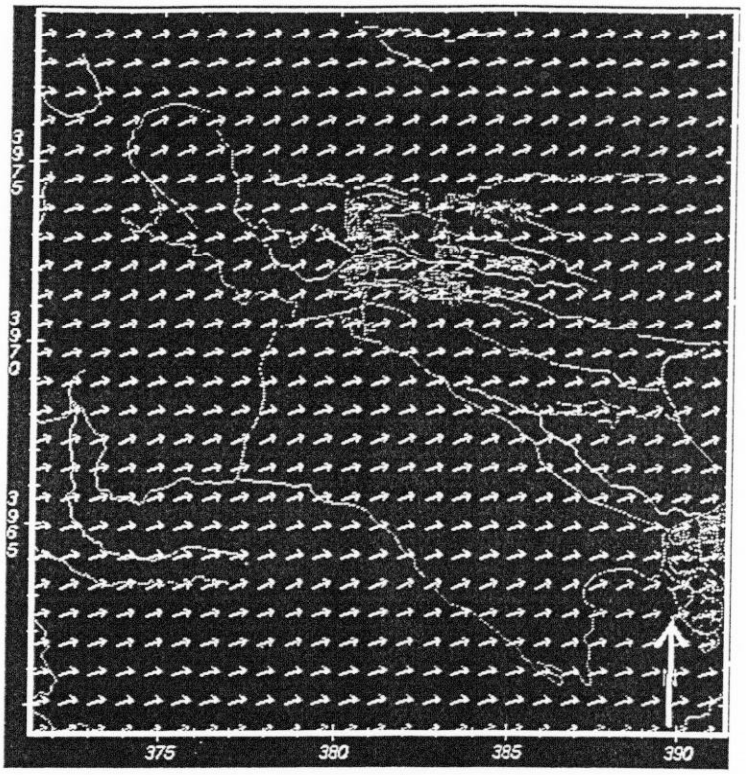


Figure 4.9. Interpolated wind vectors on the 23m AGL plane for the LANL area from MEDIC.

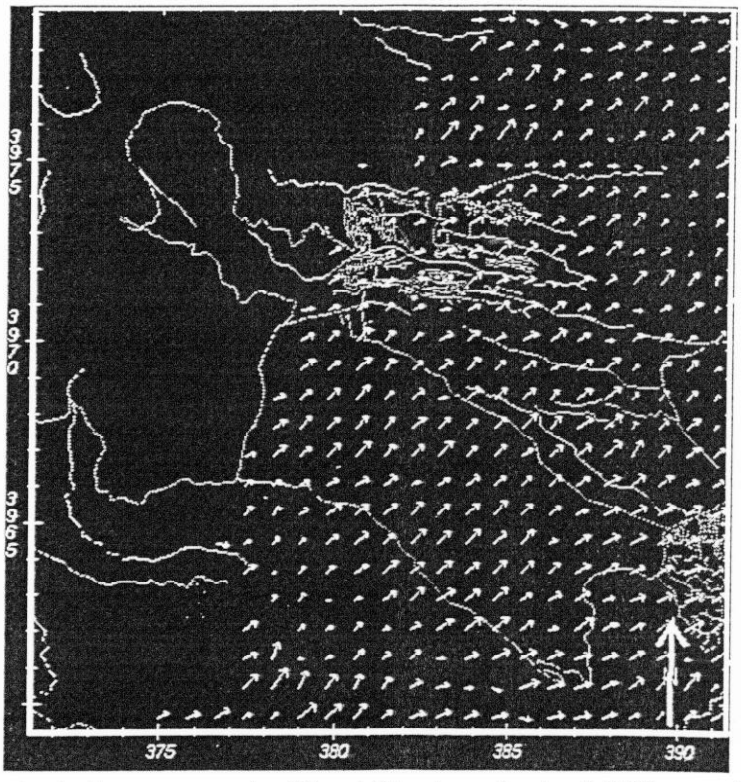


Figure 4.10. Adjusted wind vectors on the 23m AGL plane for the LANL area from MATHEW

ACKNOWLEDGMENTS

The authors would like to acknowledge H. Walker for providing the grid generation model, D. Speck for the GRIZ graphics code, and T. Kuczmarski for the GTX graphics code. Cyndi Brandt patiently and skillfully typed the manuscript. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract number W-7405-Eng-48.

REFERENCES

Arnold, D.N., 1990, "Mixed Finite Element Methods for Elliptic Problems," Comput. Methods Appl. Mech. Engrg. 82, 281-300.

Chan, S.T. and G. Sugiyama, 1997, "A New Model for Generating Mass-Consistent Wind Fields over Continuous Terrain," ANS 6th Topical Meeting on Emergency Preparedness and Response, San Francisco, April 22-25, 379-382.

Engelman, M.S., R.L. Sani, and P.M. Gresho, 1982, "The Implementation of Normal and/or Tangential Boundary Conditions in Finite Element Codes for Incompressible Fluid Flow," Int. J. Num. Meth. Fluids, 2, 225-238.

Gresho, P.M., S.T. Chan, C.D. Upson, R.L. Lee, 1984, "A Modified Finite Element Method for Solving the Time-Dependent, Incompressible Navier-Stokes Equations," Int. J. Num. Meth. Fluids, 4, 557-598 & 619-640.

Haltiner, G.J. and R.T. Williams, 1980, Numerical Prediction and Dynamic Meteorology, Second edition, John Wiley & Sons, 13.

Hodur, R.M., 1987, "Evaluation of a Regional Model with an Update Cycle," Mon. Wea. Rev., 117, 2707.

Hodur, R.M., 1997, "The Naval Research Laboratory's Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS)," Mon. Wea. Rev., 125, 1414-1430.

Hunt, J.C.R. and W.H. Snyder, 1980, "Experiments on Stably and Neutrally Stratified Flow over a Model Three-Dimensional Hill," J. Fluid Mech., 96, part 4, 671-704.

Kershaw, D., 1978, "The Incomplete Cholesky-Conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations," J. Comp. Phys., 26, 43-65.

Lee, R.L. J.R. Albritton, S.T. Chan, J.M. Leone, Jr., J.S. Nasstrom, and G. Sugiyama, 1997, "ARAC-3, A New Generation Emergency Response Modeling System," ANS 6th Topical Meeting on Emergency Preparedness and Response, San Francisco, CA, April 22-25, 575-578.

Leone, Jr, J.M, and J.S. Nasstrom, 1997, "A First Look at the New ARAC Dispersion Model," ANS 6th Topical Meeting on Emergency Preparedness and Response, San Francisco, CA, April 22-25, 383-386.

Moussiopoulos, N., Th. Flassak, and G. Knittel, 1988, "A Refined Diagnostic Wind Model," Envir. Software, 3, No.2, 85-94.

Ratto, C., R. Festa, C. Romeo, O. A. Frumeno, and M. Galluzzi, 1994, "Mass-Consistent Models for Wind Fields Over Complex Terrain: The State of the Art," Envir. Software, 9, 247-268.

Ross, D.G., I.N. Smith, P.C. Manins, and D.G. Fox, 1988, "Diagnostic Wind Field Modeling for Complex Terrain: Model Development and Testing," J. Appl. Meteor., 27, 785-796.

Sani, R.L., P.M. Gresho, R.L. Lee, and D.F. Griffiths, 1981, "The Cause and Cure (?) of the Spurious Pressures Generated by Certain Finite Element Method Solutions of the Incompressible Navier-Stokes Equations," Parts 1 & 2, Int. J. Num. Meth. Fluids, 1, 17–43, 171–204.

Sasaki, Y., 1970, "Some Basic Formlism in Numerical Variational Analysis," Monthly Weather Review, 98, 875-883.

Sherman, C.A., 1978, "A Mass-Consistent Model for Wind Fields Over Complex Terrain," J. Appl. Meteoro., 17, 312-319.

Silvester, D.J. and N. Kechkar, 1990, "Stabilized Bilinear-Constant Velocity-Pressure Finite Elements for the Conjugate Gradient Solution of the Stokes Problem," Comp. Meth. in Appl. Mech. and Engng., 79, 71–86.

Sugiyama, G. and S.T. Chan, 1997a, "Meterological Data Assimilation for Real-Time Emergency Response," ANS 6th Topical Meeting on Emergency Preparedness and Response, San Francisco, CA, 375-378.

Sugiyama, G. and S.T. Chan, 1997b, "ADAPT (Atmospheric Data Assimilation and Parameterization Techniques) User's Manual," Lawrence Livermore National Laboratory, Livermore, CA .

Sugiyama, G., R.L. Lee, and H. Walker, 1994, Conjugate Gradient MATHEW," UCRL-ID-118629, Lawrence Livermore National Laboratory, Livermore, CA.

Varga, R., 1962, *Matrix Iterative Analysis*, Prentice Hall, Englewood Cliffs, New Jersey.

Vincent, C. and R. Boyer, 1992, "A Preconditioned Conjugate Uzawa-Type Method for the Solution of the Stokes Problem by Mixed Q1-P0 Stabilized Finite Elements," Int. J. Num. Meth. Fluids, 14, 289–298.