NASA
Technical
Paper
2313

May 1984

A Spatial Model of Wind Shear and Turbulence for Flight Simulation

C. Warren Campbell

NASA

NASA Technical Paper 2313

1984

A Spatial Model of Wind Shear and Turbulence for Flight Simulation

C. Warren Campbell

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

NASA

National Aeronautics and Space Administration

Scientific and Technical Information Branch

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

The author gratefully acknowledges the support of Mr. Dennis Camp, Dr. George H. Fichtl, and Dr. William W. Vaughan of MSFC without whose support and encouragement this work would not have been possible. The JAWS Project Office of the National Center for Atmospheric Research (NCAR) deserves recognition for providing wind shear data and for many fruitful discussions. In particular, John McCarthy, Jim Wilson, Cathy Kessinger, Kim Elmore, Cindy Mueller, Rita Roberts, and Phyllis O'Rourke deserve special mention. The author also recognizes the encouragement and useful discussions of Dr. Walter Frost of the University of Tennessee Space Institute and Dr. Roland Bowles of Langley Research Center. Many other individuals contributed significantly to this effort but space doesn't permit their enumeration. Finally, the author recognizes the patience, support and encouragement of Professor V. A. Sandborn of Colorado State University.

DEDICATION

This document is dedicated to the family and friends of the hundreds of people who have died in wind shear accidents. It is the author's fondest wish that this document will contribute to the prevention of further deaths resulting from this cause.

SYMBOLS

$a = 1{,}339$ = a dimensionless constant

$e_{1,i}$, $e_{2,i}$ = two unit vectors

$E(\nu) = 4\pi\nu^2 \cdot 1/2\ \Phi_{ii}\ (\nu_1, \nu_2, \nu_3)$

f = longitudinal isotropic correlation coefficient

g = transverse isotropic correlation coefficient

$H(\nu)$, $H(\nu_1, \nu_2, \nu_3)$ = one- and three-dimensional filter functions

$K_\nu(x)$ = the modified Bessel function of the second kind of order ν

L = longitudinal integral length scale of turbulence

N_1, N_2 = first and second components of dimensionless spatial frequency

N_θ = dimensionless frequency

r_i = ith component of separation distance

$r = (r_1^2 + r_2^2 + r_3^2)^{1/2}$

R_{ij} = cross correlation tensor

t = time

T = ambient temperature

$u_i(x,y,z,t)$ = three component simulated gusts

$U_i(x,y,z,t)$ = interpolated JAWS winds

V = aircraft airspeed

$w_i(x,y,z)$ = three-dimensional block of Monte Carlo simulated turbulence

x_{n_1,n_2,n_3} = discrete space function of three variables

X_{k_1,k_2,k_3} = Discrete Fourier Transform (DFT) of x_{n_1,n_2,n_3}

$\alpha(\nu)$ = phase of one-dimensional filter function

β = volumetric expansion coefficient

δ_{ij} = Kronecker delta

$\delta(t)$ = Dirac delta function

Δr_2 = lateral gust probe separation in cross-spectra calculations or measurements

ΔT = parcel temperature difference from surroundings

ϵ = turbulent kinetic energy dissipation

θ = Vt/L = dimensionless time

$\Theta_{ij}(\nu_1)$ = one-dimensional spectrum function

μ_{eff} = effective turbulent viscosity

ν_i = spatial frequencies

ν_{s_i} = spatial sampling frequencies

$\nu = (\nu_1^2 + \nu_2^2 + \nu_3^2)^{1/2}$

ϕ_{ii} (N_θ, $\Delta r_2/L$) = cross-spectra

$\Phi_{ij}(\nu_1,\nu_2,\nu_3)$ = three-dimensional spectrum tensor

$\Psi_{ij}(\nu_1,\nu_2)$ = two-dimensional spectrum tensor

$\rho = (\nu_1^2 + \nu_2^2)^{1/2}$ or fluid density. The meaning is obvious from the usage.

$\sigma_i(x,y,z,t)$ = gust standard deviation

CHAPTER I. INTRODUCTION

Approximately every two years in the United States a major wind shear related airliner accident occurs killing tens of people. The most recent of these (as of this writing) was the July, 1982 crash of Pan Am Flight 759 during takeoff at New Orleans International Airport, 156 died aboard the plane and eight others were killed on the ground as the plane crashed into a subdivision.

This crash and others like it were caused by wind shear associated with a small scale atmospheric phenomenon known as a microburst. In the past few years two field programs were funded to study the microburst. The programs were the Northern Illinois Meteorological Research on Downburst (NIMROD) Project and the Joint Airport Weather Studies (JAWS) Project. The JAWS Project measured some 70 microburst events with Doppler radar during May through August, 1982. Aside from scientific interest, several wind shear data sets were subjected to detailed analysis and put into a form for use in flight simulator research. These data sets constitute the best wind shear measurements ever made.

The JAWS data are presented on a three-dimensional Cartesian grid with grid spacings which vary from one case to another but are approximately 200 meters. Hence, the JAWS data contain no information on turbulence with length scales shorter than 400 meters. For nominal landing speeds, some frequencies of importance to aircraft response are not contained in the JAWS data. These small scales of turbulence must be added for realistic flight simulations.

During September, 1983 a workshop sponsored jointly by the FAA, NCAR, and NASA was held in Boulder, Colorado. The workshop brought together researchers directly involved in the JAWS program and potential users of the JAWS data. The users

were offered a selected JAWS microburst case and asked what additional information they required. From this exchange, several facts emerged. Turbulence was high on their list of priorities. Users from private industry look to public agencies such as NASA to tell them how to add turbulence to the JAWS data. In the present economic climate, the airlines have no money to fund basic turbulence simulation research, nor can they subsidize the simulator manufacturers to do it. Of the agencies involved, NASA is probably best equipped to do the job.

This document presents an approach to wind simulation which is a significant advance in the state-of-the-art. The technique involves the addition of three-dimensional Monte-Carlo simulated turbulence to the JAWS data sets. Using this approach, all aerodynamic loads and moments (including roll and yaw) may be calculated from the winds simulated over the body of the aircraft. This level of information was previously unavailable from wind simulation models. The spatial model concept, in part, provides the answers to the question of how to add turbulence to the JAWS data sets. It also serves to direct future measurement programs and microburst research to obtain required turbulence information.

In addition, an extension of a previous cross-spectral model based on the von Karman turbulence model is presented. Results of the theory are compared with measurements. The cross-spectra are a natural part of a three-dimensional simulation.

A complete explanation of the generation of turbulence, and addition to the JAWS data are presented along with FORTRAN computer codes. Background information on turbulence, microbursts, the JAWS data, and Monte-Carlo turbulence simulation are also presented. The background material is necessary for a thorough understanding of the spatial model. The procedure for generating the turbulence is rather complex and an attempt was made to present the material in an intuitive fashion with illustrations and geometric interpretations while complex derivations were relegated to the Appendices.

CHAPTER II. ATMOSPHERIC WIND SHEAR AND TURBULENCE

Wind shear, or more precisely wind gradients ($\partial u/\partial x$ is a wind dilation) have been recognized as a cause of aircraft crashes for some time. What was not recognized until 1976 was the small areal extent of the crash-causing phenomenon. The small scale wind shear events were called microbursts by Fujita [1]. After the recognition of microbursts, field programs to study them were done, most notably NIMROD and JAWS. One of the achievements of JAWS was the development of microburst data sets for use in flight simulation. These data sets are on a relatively coarse grid and therefore do not contain information on fine scale turbulence.

In order to understand the nature of the proposed wind shear model, a description of microbursts is necessary. The necessary description and the characteristics of the JAWS data sets are included in this chapter. These descriptions contain a discussion of some microburst models. In anticipation of the need for adding turbulence to the JAWS data for the purpose of flight simulation, relevant aspects of turbulence theory are presented.

A. Microbursts

Fujita defines a downburst as, "a strong downdraft which induces an outburst of damaging winds at the surface." A microburst is defined by Fujita [1] as a downburst of horizontal dimension less than 4 kilometers. A more useful definition is given by Wilson and Roberts [2] who define the microburst as a downburst having a differential surface velocity greater than 10 meters/sec with the distance over which the velocity difference occurs being between 0.4 and 4 km.

Figure 1 shows the life cycle of a microburst as idealized by Fujita [3]. In the figure, descending air meets the ground and spreads out creating strongly diverging

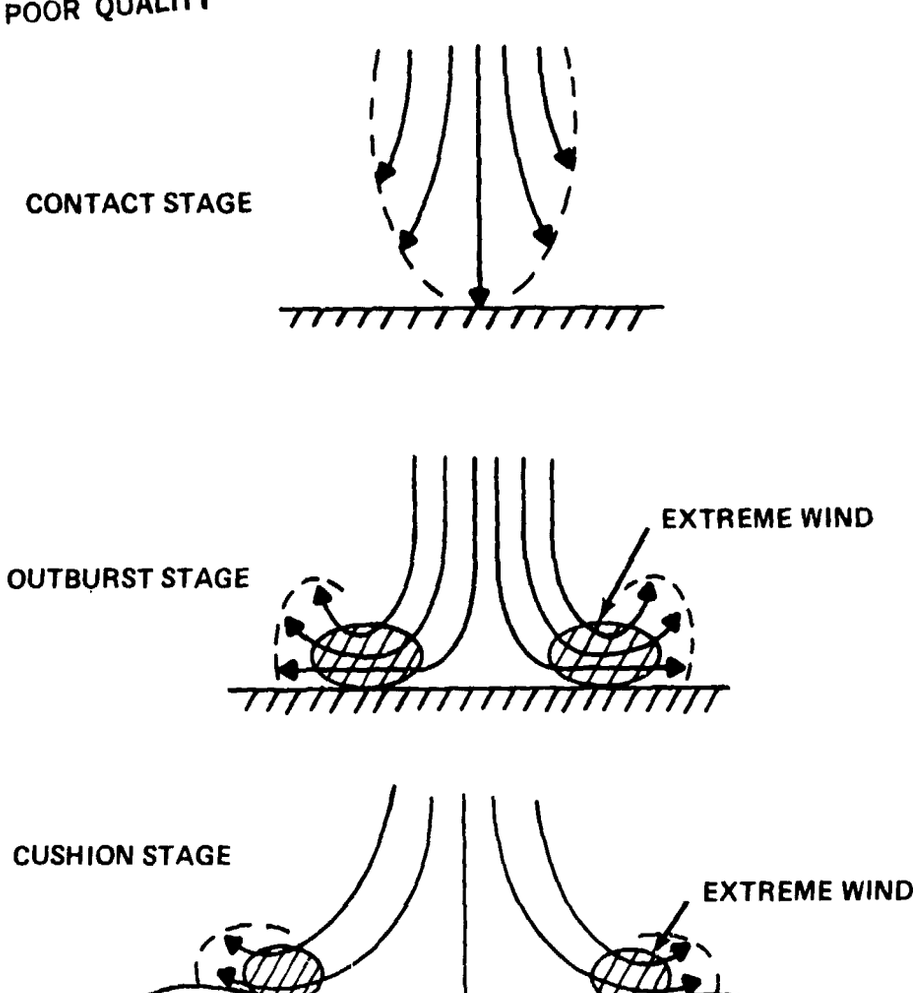


Figure 1. Life cycle of a microburst [3].

surface winds. Microbursts may or may not be associated with rainfall at the surface and are accordingly classified as wet or dry. The microburst cases studied during the JAWS program show no correlation between surface rain rate and microburst intensity [2]. The most intense JAWS microburst had a velocity differential of 48 m/sec (96 kts) and was associated with moderate rainfall at the surface [2]. The non-correlation of microburst intensity with rainfall intensity increases their hazard because they can occur in apparently benign conditions.

The final stage of the microburst is called the cushion stage by Fujita. During this period surface winds decay and the outflow no longer penetrates to the surface. The entire life cycle of the microburst from contact to decay is typically 10 minutes. During the mature stage, the maximum wind will occur 50 to 100 meters above the surface while the depth of the outflow is typically 600 meters.

The short duration, high energy, and small length scale of the phenomenon create problems for those who would predict, detect, or fly through microbursts. For prediction, the short duration and random nature of the microburst make forecasting of specific events impossible. The best that can be done is to forecast conditions conducive for the occurrence of microbursts.

For detection, the small size creates problems. For years at major U.S. airports the FAA has operated Low Level Wind Shear Alert Systems (LLWSAS). LLWSAS consists of several wind monitoring stations located around the periphery of the airport and one center field station. If vector differences of wind velocity between the center field detector and any of the other stations exceed a certain level, a wind shear alert is issued. Unfortunately, fatal wind shear related crashes have occurred at airports with operating LLWSAS, the latest being the crash of Pan Am 759 at New Orleans. One problem is that LLWSAS station spacing is larger than the average microburst. The only system offering hope of reliable detection in the near future is Doppler radar, but the cost of the required system may be prohibitive. Wilson and Roberts [2] have studied the requirements of such a system.

People encountering microbursts at low levels during takeoff or approach frequently find it a once-in-a-lifetime experience. Airliner approach airspeeds are typically 75 to 80 m/sec while stall speeds in landing configuration are roughly 60 m/sec. With measured airspeed differences of as much as 48 m/sec, microbursts obviously have a drastic effect on aircraft performance during takeoff and landing. Figure 2 depicts the situation. On approach the aircraft encounters an increasing headwind and begins to rise above the glide slope. Pilots have a tendency to throt. · back in an attempt to return to the glide slope. Passing through the center of the microburst, the plane encounters a strong downdraft and an increasing tail wind. A subsequent loss in lift occurs and the engines cannot respond quickly enough to save the aircraft. Engines of large air transport planes generally require about seven seconds to spool up because of their large inertia.

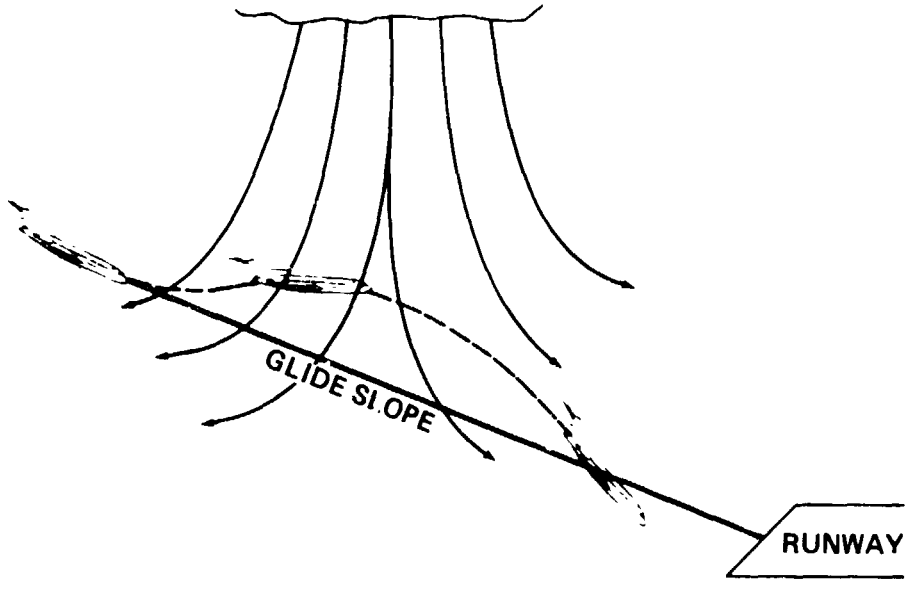


Figure 2. Landing aircraft microburst encounter.

Microburst Models

The preceding paragraphs contained a brief summary of microburst characteristics and effects on aircraft. In the Wind Shear Simulation Workshop in Boulder (Sept. 7-8, 1983), simulator users and manufacturers expressed a need for generic microburst models.

Because of storage limitations in current simulators, a microburst described by algebraic equations is needed. We now proceed to a more detailed consideration of microbursts with discussions of both conceptual models and simple algebraic models.

Examination of Figure 1 suggests a pair of microburst models. The first is that the microburst is a turbulent jet impinging on a surface. The general vertical velocity profile through the outflow has much the same appearance as the circular wall jet. Because of the transient nature of the microburst, the steady state circular wall jet is not a totally satisfactory model. In addition, certain turbulence characteristics of a wall jet do not seem to agree with measurements [4].

A second model requires the occurrence of a source of cooling at cloud base. The region of cooling is in effect a momentum source. One mechanism for cooling is the evaporation of falling rain. As water droplets evaporate and absorb latent heat, the surrounding air is cooled. Figure 1 depicts Fujita's microburst life cycle. The cool parcel interpretation is clear for the contact and outburst stages of the microburst. The cushion stage requires some explanation. During the decaying stage of the microburst the momentum source at cloud base would be dying. The result is that the falling air would be warmer than the air below it so that a stable layer forms near the ground. Because the descending air is warmer than air already at the surface, deceleration and spreading of the downdraft begin at a higher altitude.

A third model was devised by Caracena [5] to explain some narrow damage swaths observed by Fujita and Wakimoto [6]. Caracena feels that if the jet model is accepted a more diffuse damage swath would result. He looked for a mechanism capable of maintaining its integrity while transferring momentum over long distances. An obvious candidate is the vortex ring. Supporting the vortex ring model are some microburst photographs which have a vortex ring appearance. A vortex ring approaching a friction-less surface expands and decays similar to observed behavior of microbursts. To illustrate his hypothesis Caracena constructed a simple vortex ring generator. If a ring is generated

obliquely to a grassy surface, the motion of the grass outlines a narrow "damage swath." Caracena's arguments concerning jet diffusion are not altogether compelling, however. A similar "damage swath" can be created with a hair dryer.

A question concerning Caracena's model is the source of the vortex ring which requires an impulsive force for generation. One possible source identified by Caracena is the collapse of overshooting tops above the anvil of large convective storms. Indeed, Fujita and Wakimoto [6] have shown some apparent correlation in time between the collapse of overshooting tops and the occurrence of microbursts.

Jet, cool parcel (momentum source), and vortex ring models of microbursts are conceptual models and full quantitative ramifications of these ideas are not yet developed. Two models are developed to the point of algebraic equations, however. These are not scientific models, but are intended for use in flight simulators. The equations were devised without reference to the equations of flow in one case and with reference to continuity only in the other case.

Bray proposed an extremely simple model for use on Ames Research Center simulators. This model is described and critiqued by Elmore [7]. The Bray model has several shortcomings criticized by Elmore including uniform downdraft source and failure to satisfy continuity. Elmore also states that the downdraft diameter is too large for the amount of outflow produced. Elmore offers suggestions to make the Bray model more realistic. These suggestions include modification of the source terms and alteration to satisfy continuity

A more sophisticated model was proposed by Zhu and Etkin, 1983 [8] which is an ideal fluid model. It involves the flow induced by a doublet disc of variable strength together with its image. With this model, Zhu and Etkin were able to demonstrate the excitation of aircraft phugoid frequencies by microbursts.

The Bray model, though admittedly crude, does provide a reasonable simulation. With the addition of Elmore's suggestions, simulations should be improved. Of the three

models described, Zhu's and Etkin's model is probably the best. Neither the Bray nor the Elmore model were intended as scientifically accurate pictures of microbursts and as a result some of the physics not considered essential for simulation are neglected. Neither of the three show a microburst front and consequent abrupt shear at the leading edge. The equations which can be made functions of time as well as space are assumed separable, i.e., $f(x,t) = X(x)\ T(t)$. This is not a realistic picture since it implies that the microburst intensifies and decays uniformly at every point.

The preceding discussions have dealt with conceptual and algebraic pictures of microbursts. In the next subsection, characteristics of real JAWS measurements are described.

JAWS Data Sets

At the present time three of the JAWS data cases have undergone multiple Doppler analysis to determine three component winds. These three cases are summarized in Table 1.

Table 1. JAWS Dual Doppler Cases

Date	Δx^1 (m)	Δy^2 (m)	Δz^3 (m)	ΔV_{max} (m/sec)	Number of x x y x z grid points
June 29, 1982	300	300	250	25	60 x 60 x 9
July 14, 1982	200	200	150	30	60 x 60 x 11
August 5, 1982	150	150	250	30	81 x 81 x 9

1. Δx is the east-west grid spacing.
2. Δy is the north-south grid spacing.
3. Δz is the vertical grid spacing.

The August 5 case was selected for release to selected companies because of its high resolution and intensity. The data were summarized in a document [9] which contains a plot of low level winds showing a sinuous jet emitted from the northeast quadrant of the microburst. This feature also appears in the winds in the next level up but gradually disappears at higher levels. The jet seems to be a real phenomenon and if so it raises some questions. The main question is, "Why is it there?" Is this a reflection of a convoluted source up at cloud base? This idea seems highly unlikely since diffusion tends to reduce source convolution in nature. The alternative is that something is happening at the surface. In other words the jet may be a result of topography or of some surface roughness anomaly. Reference to a Denver map shows that the jet occurs in the near vicinity of the South Platte River basin and the jet flow is in the same general direction as the river. The implication is that the cold stable air is being channeled along the river basin.

The JAWS data is generally displaced freely relative to airports in simulations. If the jet is a surface effect, freely moving the data to flat areas surrounding airports may not be a reasonable procedure.

Figure 3 is a vertical section through the July 14 case which is a typical asymmetric microburst embedded in a wind field. The center of the section shows a strong downdraft. On the right side of the downdraft is a slight updraft. The width of the downdraft is roughly 1 km. This downdraft is just off center of a moderate to heavy rainshaft and this microburst would be classified as wet.

Microbursts weren't identified until 1976 and many questions concerning their nature remain unanswered. What is the best model of a microburst (jet, momentum source, vortex ring, etc.). What is the effect of source size and strength on outflow intensity? What effect does source height have on the microburst? What effect does surface topography and/or roughness have on the outflow pattern and damage swath? What degree of kinematic organization exists in a microburst (vortex ring)? How do

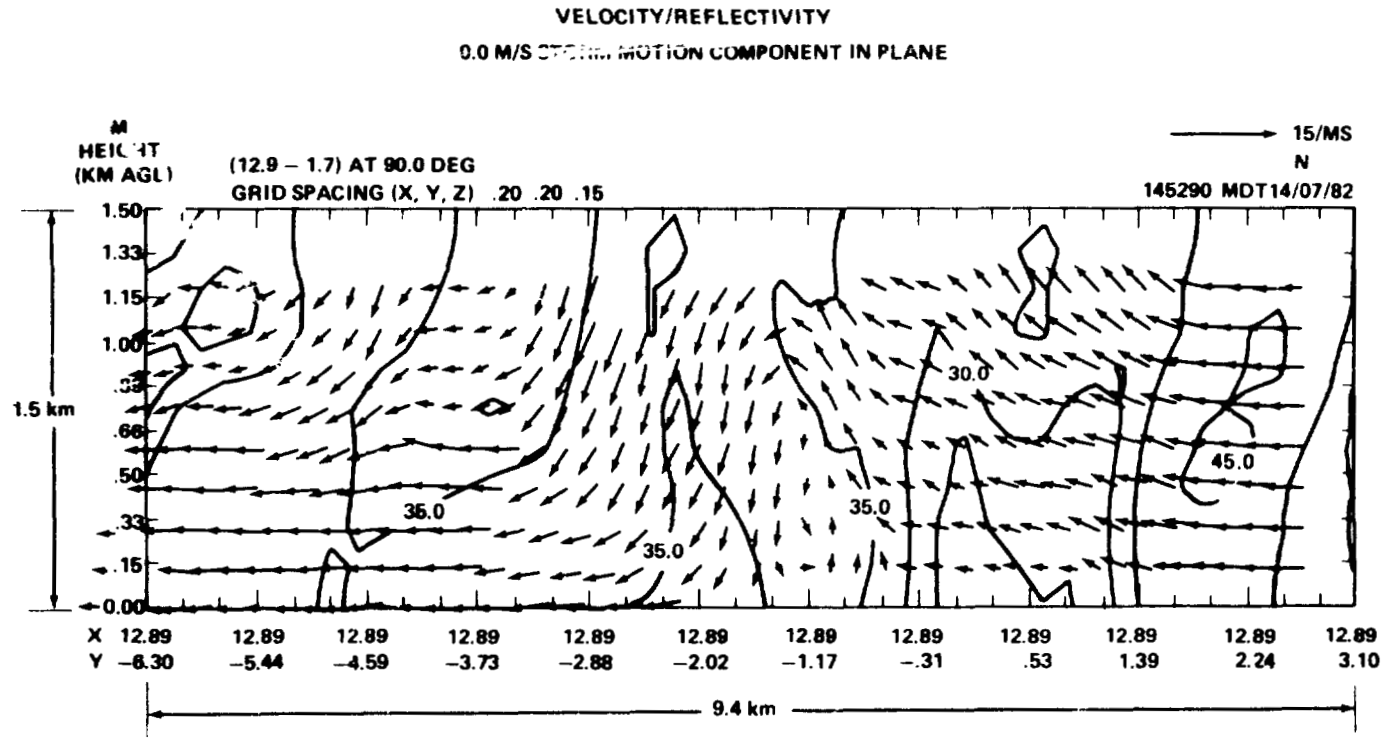


Figure 3. Vertical section through the July 14 JAWS microburst.

turbulence characteristics (length scale and intensity) vary through a microburst? This latter question is crucial for turbulence simulation. JAWS has provided partial answers to some of the above questions. Most require additional measurements preferably in the controlled environment of a laboratory. Appendix A is devoted to laboratory approaches for obtaining hese answers.

B. Isotropic Turbulence

Atmospheric turbulence is neither isotropic, homogeneous, nor Gaussian. Despite this fact at the higher frequencies and shorter length scales, turbulence tends toward isotropy. Isotropic turbulence is the simplest form of turbulence. In the discussion of the spatial wind model, a means for separating the longer wavelengths and non-homogeneities will be discussed. The need for the separation arises from the difficulty of generating nonisotropic turbulence with Monte-Carlo techniques. In anticipation of a later need, some useful properties of isotropic turbulence are presented in this subsection.

The description of the kinematics of homogeneous isotropic turbulence begins with the definition of two functions, the longitudinal and transverse correlation functions. These two functions, defined in Figure 4, are functions of separation r, and time. A similarity form which is a function of r/L only is assumed. Time dependence is contained in L = L(t). The best known models of f and g are the von Karman and Dryden models, both of which can be written as functions of r/L only. By virtue of continuity a relation exists between f and g.

$$g = f + \frac{r}{2}\frac{df}{dr} \tag{1}$$

Different investigators have used different functions to fit the measured correlations. The two most famous correlation models are by Dryden and von Karman. The Dryden correlations are given by

$$f = \frac{\overline{u_1 u_2}}{\sigma^2} \qquad g = \frac{\overline{v_1 v_2}}{\sigma^2}$$

Figure 4. Longitudinal and transverse correlation functions.

$$f(r) = e^{-r/L} \tag{2}$$

$$g(r) = (1 - r/2L)\, e^{-r/L} \tag{3}$$

where L is the longitudinal length scale of turbulence ($\int_0^\infty f(r)\, dr$). These equations have the advantages of simplicity and rational spectra. Simplicity is an obvious advantage, but the advantages of rational spectra are related to mechanisms for simulating turbulence by Monte-Carlo methods. The main disadvantage of the Dryden model is that it does not fit measured data quite as well as the von Karman model.

The correlation functions of the von Karman model are given by the following equations:

$$f = \frac{2^{2/3}}{\Gamma(1/3)} \left(\frac{r}{aL}\right)^{1/3} K_{1/3}\left(\frac{r}{aL}\right) \tag{4}$$

$$g = \frac{2^{2/3}}{\Gamma(1/3)} \left(\frac{r}{aL}\right)^{1/3} \left[K_{1/3}\left(\frac{r}{aL}\right) - \frac{r}{2aL} K_{2/3}\left(\frac{r}{aL}\right) \right] \tag{5}$$

where

$\Gamma(x)$ = gamma function

a = a constant = 1.339

$K_\nu(x)$ = the modified Bessel function of the second kind of order ν .

The correlation functions are important because the complete three-dimensional correlation tensor can be expressed in terms of these two functions.

$$R_{ij} = \sigma^2 \left[(f-g) \frac{r_i r_j}{r^2} + g \, \delta_{ij} \right] \tag{6}$$

where

$$\delta_{ij} = 1 \quad i = j \quad \text{Kronecker Delta}$$
$$= 0 \quad i \neq j$$

The above result was obtained by direct calculation by von Karman and Howarth [10]. The tensor function can be used to calculate correlations between velocities oriented along any two unit vectors, say $e_{1,i}$ and $e_{2,j}$. The scalar correlation between these two velocities is given by

$$R_{12} = R_{ij} \, e_{1,i} \, e_{2,j} \tag{7}$$

In the above equation the Einstein summation convention is in effect.

Figure 5 gives a geometric interpretation of the preceding discussion. Since R_{ij} is an isotropic tensor, R_{12} is invariant with respect to rotation, i.e., if the three vectors v_1, v_2, and r are rotated in any manner through the isotropic, homogeneous turbulence so that they maintain the same orientation with respect to each other (rigid body rotation) then R_{12} is unchanged.

The correlation tensor has an equivalent representation in frequency space.

$$\Phi_{ij}(\nu_1,\nu_2,\nu_3) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_{ij}(r_1,r_2,r_3) \, e^{-j2\pi(r_1\nu_1+r_2\nu_2+r_3\nu_3)} \, dr_1 \, dr_2 \, dr_3$$

$$= \frac{E(\nu)}{4\pi\nu^4} (\nu^2 \, \delta_{ij} - \nu_i \, \nu_j) \tag{8}$$

where ν_1, ν_2, and ν_3 are the three spatial frequency components

$$\nu = (\nu_1^2 + \nu_2^2 + \nu_3^2)^{1/2}$$

and $E(\nu) = 4\pi\nu^2 \cdot 1/2\ \Phi_{ii}(\nu_1, \nu_2, \nu_3)$ and the Einstein simulation convention applies.

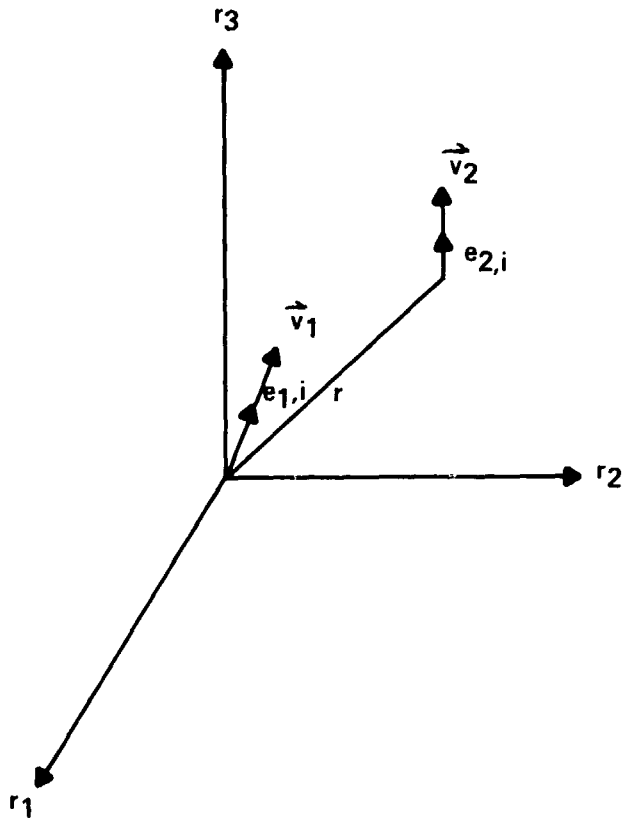


Figure 5. Geometry of the cross-correlation tensor.

As in the case of correlation, Φ_{ij} is a tensor function and a scalar function is obtained as before.

$$\Phi_{12}(\nu_1,\nu_2,\nu_3) = \Phi_{ij}\ e_{1,i}\ e_{2,j} \tag{9}$$

In the above expression $e_{1,i}$ and $e_{2,j}$ are the same unit vectors which correspond to $\vec{v}_1$ and $\vec{v}_2$ in Figure 5, and Φ_{12} is the scalar cross-spectrum corresponding to $\vec{v}_1$ and $\vec{v}_2$.

$E(\nu)$ has representations corresponding to the Dryden and von Karman models. For the Dryden model the equation is

$$E(\nu) = 16\,\sigma^2\,L\,\frac{(2\pi L\nu)^4}{[1 + (2\pi L\nu)^2]^3} \quad . \tag{10}$$

The corresponding von Karman equation is

$$E(\nu) = \frac{110}{9}\,\sigma^2\,L\,\frac{(2\pi aL\nu)^4}{[1 + (2\pi aL\nu)^2]^{17/6}} \quad . \tag{11}$$

From these two equations Φ_{ij} for the two models can be calculated. For the Dryden model the result for the auto-spectra is

$$\Phi_{ii}(\nu_1,\nu_2,\nu_3) = \frac{64\,\sigma^2\,L^5\,\pi^3\,(\nu^2 - \nu_i^2)}{[1 + (2\pi L\nu)^2]^3} \quad . \tag{12}$$

In this equation and the next, the Einstein summation convention is suspended.

The von Karman model is

$$\Phi_{ii}(\nu_1,\nu_2,\nu_3) = \frac{440\,\pi^3}{9}\,\frac{\sigma^2\,a^4\,L^5\,(\nu^2 - \nu_i^2)}{[1 + (2\pi aL\nu)^2]^{17/6}} \quad . \tag{13}$$

The spectra for the vertical velocity component, equations (12) and (13), can be rewritten for the Dryden and von Karman models, respectively, as follows:

Dryden Model

$$\Phi_{33}(\nu_1,\nu_2,\nu_3) = \frac{64\,\sigma^2\,L^5\,\pi^3\,(\nu_1^2 + \nu_2^2)}{[1 + (2\pi L)^2\,(\nu_1^2 + \nu_2^2 + \nu_3^2)]^3} \tag{14}$$

von Karman model

$$\Phi_{33}(\nu_1,\nu_2,\nu_3) = \frac{440\,\pi^3}{9}\,\sigma^2\,a^4\,L^5\,\frac{(\nu_1^2+\nu_2^2)}{[1+(2\pi aL)^2\,(\nu_1^2+\nu_2^2+\nu_3^2)]^{17/6}} \quad . (15)$$

Equations (14) and (15) show rotational symmetry about the ν_3 axis, i.e., they are functions of $\rho^2 = \nu_1^2 + \nu_2^2$ and ν_3. Working with the von Karman spectrum, Equation (15), we can solve for $\nu_3 = \nu_3(\Phi_{33},\rho)$. The result is

$$\nu_3 = \pm \sqrt{\left[\left(\frac{C_1\,\rho^2}{\Phi_{33}}\right)^{6/17} - 1\right]\frac{1}{C_2} - \rho^2} \quad , \qquad (16)$$

where C_1 and C_2 are functions of σ and L.

Equation (16) is an equation for a surface of constant Φ_{33} in wavenumber space. Values of C_1 and C_2 for the corresponding spatial frequency spectrum are developed in Appendix B.

Equation (16) is the equation of a toroid. Cross sections of it are plotted in Figure 6 and a perspective of the surface is depicted in Figure 7. Surfaces of constant Φ_{33} are found if the curves of Figure 6 are rotated about the ν_3 axis. A result similar to (16) can be obtained for Φ_{11} and Φ_{22} with rotational symmetry about the ν_1 and ν_2 axes, respectively.

Figure 6 seems to imply that Φ_{33} is singular, i.e., it approaches infinity near the origin. In fact Φ_{33} does have a finite maximum value.

Three-dimensional correlations and cross-spectral tensors are extremely difficult to measure. Actually, one-dimensional correlations are measured and corresponding one-dimensional spectra calculated from the correlations, i.e.

$$R_{ij}(r_1,o,o) = \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\Phi_{ij}(\nu_1,\nu_2,\nu_3)\,d\nu_2\,d\nu_3\right]e^{-j2\pi\nu_1 r_1}\,d\nu_1$$

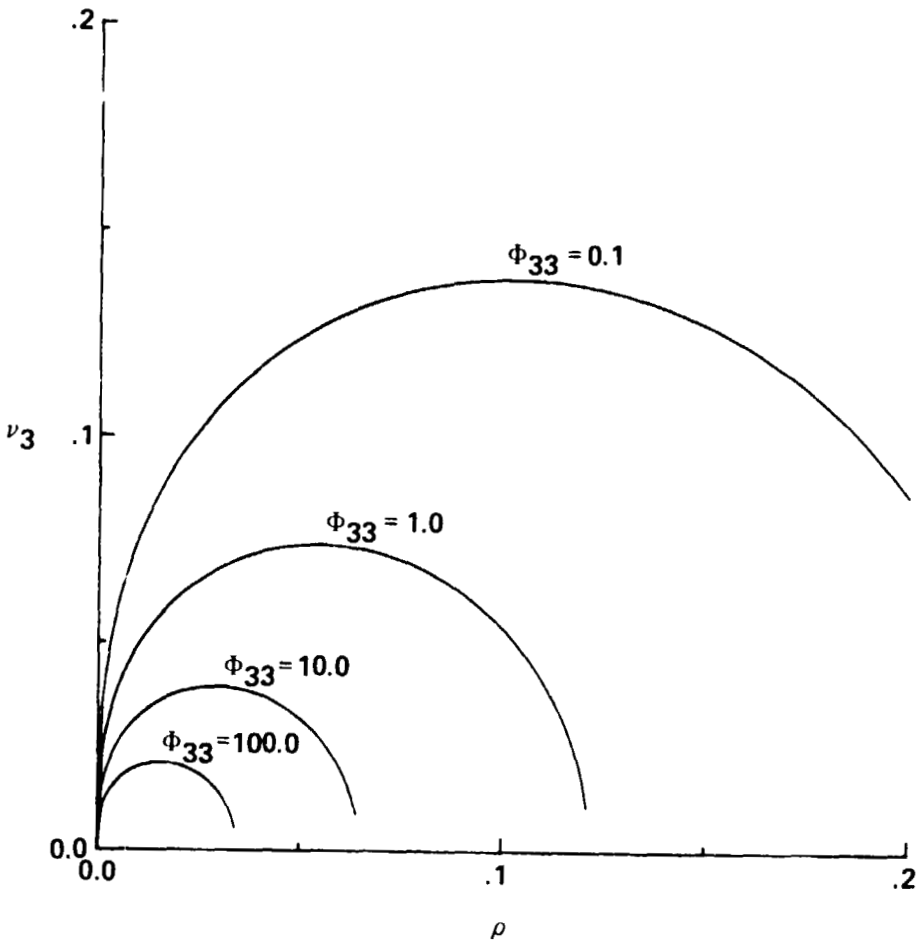


Figure 6. Sections of surfaces of constant Φ_{33} for the von Karman model.

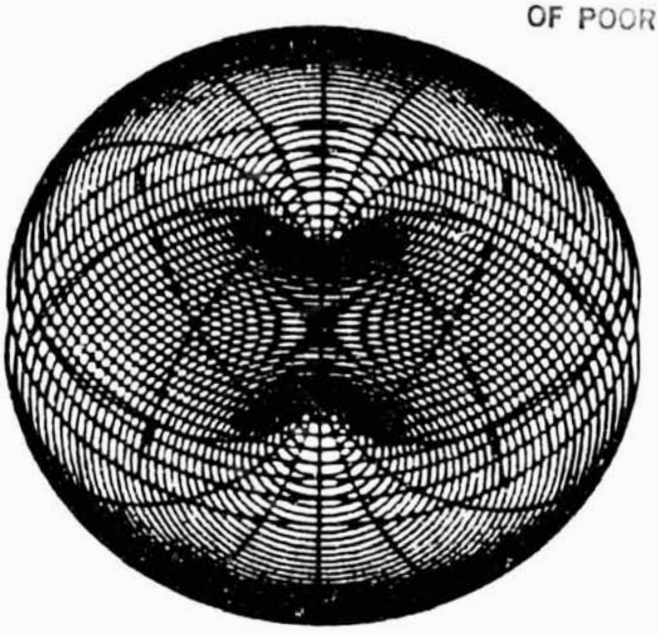


Figure 7. Perspective of surface of constant Φ_{33} for the von Karman model.

Which implies that the one-dimensional spectra are given by

$$\Theta_{ij}(\nu_1) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Phi_{ij}(\nu_1,\nu_2,\nu_3)\, d\nu_2\, d\nu_3 \quad . \tag{17}$$

By similar means, two-dimensional spectra can be deduced.

$$\psi_{ij}(\nu_1,\nu_2) = \int_{-\infty}^{\infty} \Phi_{ij}(\nu_1,\nu_2,\nu_3)\, d\nu_3 \quad . \tag{18}$$

One- and two-dimensional von Karman wave number spectra are given by Etkin [11]. When converted to spatial frequency spectra the results are

$$\psi_{11}(\nu_1,\nu_2) = \frac{2\pi}{3}(\sigma aL)^2 \frac{1 + (2\pi aL\nu_1)^2 + \frac{11}{3}(2\pi aL\nu_2)^2}{[1 + (2\pi aL)^2 (\nu_1^2 + \nu_2^2)]^{7/3}} \tag{19}$$

$$\psi_{22}(\nu_1,\nu_2) = \frac{2\pi}{3}(\sigma aL)^2 \frac{1 + \frac{11}{3}(2\pi aL\nu_1)^2 + (2\pi aL\nu_2)^2}{[1 + (2\pi aL)(\nu_1^2 + \nu_2^2)]^{7/3}} \tag{20}$$

$$\psi_{33}(\nu_1,\nu_2) = \frac{64\pi^3}{9}\ \sigma^2(aL)^4\ \frac{(\nu_1^2+\nu_2^2)}{[1+(2\pi aL)^2\,(\nu_1^2+\nu_2^2)]^{7/3}} \tag{21}$$

$$\Theta_{11}(\nu_1) = 2\sigma^2 L\ \frac{1}{[1+(2\pi aL\nu_1)^2]^{5/6}} \tag{22}$$

$$\Theta_{22}(\nu_1) = \Theta_{33}(\nu_1) = \sigma^2 L\ \frac{1+\frac{8}{3}(2\pi aL\nu_1)^2}{[1+(2\pi aL\nu_1)^2]^{11/6}} \tag{23}$$

The results from this subsection are used in the next subsection to calculate a theoretical cross-spectral model. In Chapter VI results from the preceding discussion are used to calculate filler functions for generating three-dimensional turbulence.

C. A Theoretical Cross-Spectral Model and Comparison with Measurements

Elements of the previous discussion concerning isotropic turbulence will now be used to develop a cross-spectral model of turbulence. The assumption of frozen, von Karman turbulence is made. Previously Houbolt and Sen [12] did a similar analysis to obtain the cross-spectrum corresponding to the vertical velocity component. An equivalent approach is used to extend their analysis to the longitudinal and lateral cross-spectra.

In the analysis that follows it is convenient to non-dimensionalize the spectra. Without non-dimensionalization, annoying conversion constants must be carried along so that the units are correct. Equations (19) through (21) are non-dimensionalized using the transformations $N_i = \nu_i L$.

$$\Psi_{ij}(N_1,N_2) = \psi_{ij}\left(\frac{N_1}{L},\frac{N_2}{L}\right)\Big/L^2 \tag{24}$$

Applying this transformation to (19) through (21) yields:

$$\Psi_{11}(N_1,N_2) = \frac{2\pi}{3}(\sigma a)^2\ \frac{1+(2\pi aN_1)^2+\frac{11}{3}(2\pi aN_2)^2}{[1+(2\pi a)^2\,(N_1^2+N_2^2)]^{7/3}} \tag{25}$$

$$\Psi_{22}(N_1,N_2) = \frac{2\pi}{3}(\sigma a)^2 \frac{1 + \frac{11}{3}(2\pi a N_1)^2 + (2\pi a N_2)^2}{[1 + (2\pi a)^2 (N_1^2 + N_2^2)]^{7/3}} \quad (26)$$

$$\Psi_{33}(N_1,N_2) = \frac{64\pi^3(\sigma a)^2}{9} \frac{(N_1^2 + N_2^2)}{[1 + (2\pi a)^2 (N_1^2 + N_2^2)]^{7/3}} \quad . \quad (27)$$

The frequency transformation, $N_i = \nu_i L$ implies a transformation in space, $X = r_1/L, Y = r_2/L$. Then

$$R_{ij}(X,Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Psi_{ij}(N_1,N_2)\, e^{+j2\pi(N_1 X + N_2 Y)}\, d N_1\, d N_2 \quad . \quad (28)$$

The object we are trying to achieve is the cross-correlation between the velocity at one aircraft wing tip at one moment and the velocity at the other wingtip at a later moment. Assuming frozen turbulence, Figure 8 applies. Making the transformation $\theta = Vt/L$ gives

$$R_{ij}\left(\theta, \frac{\Delta r_2}{L}\right) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Psi_{ij}(N_1,N_2)\, e^{+j2\pi(N_1\theta + N_2 \Delta r_2/L)}\, d N_1\, d N_2 \quad . \quad (29)$$

Equation (29) is an expression for the desired cross-correlation. To obtain the desired cross-spectra, the Fourier transform of (29) is taken.

$$\phi_{ij}\left(N_\theta, \frac{\Delta r_2}{L}\right) = \int_{-\infty}^{\infty} R_{ij}(\theta, \Delta r_2/L)\, e^{-j2\pi N_\theta \theta}\, d\theta$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Psi_{ij}(N_1,N_2)\, e^{+j2\pi N_2\, \Delta r_2/L} \int_{-\infty}^{\infty} e^{+j2\pi\theta(N_1 - N_\theta)}\, d\theta\, dN_1\, dN_2$$

The exponential integral with respect to θ above is the Dirac delta function $\delta(N_1 - N_\theta)$, so the final result is

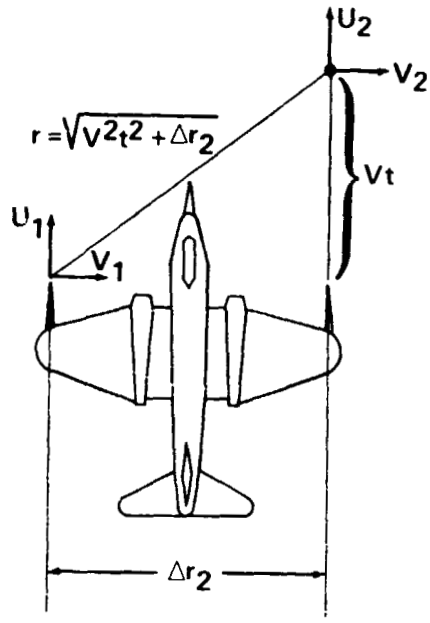


Figure 8. Illustration of cross-correlation distance.

$$\phi_{ij}\left(N_\theta, \frac{\Delta r_2}{L}\right) = \int_{-\infty}^{\infty} \Psi_{ij}(N_\theta, N_2)\, e^{+j2\pi N_2 \Delta r_2/L}\, dN_2 \quad . \tag{30}$$

Using Equation (30) along with Equations (25) through (27) gives the final expressions for the three cross-spectra.

$$\phi_{11}(N_\theta, \Delta r_2/L) = \frac{4\pi a^2}{3} \int_0^{\infty} \frac{\left[1 + \frac{11}{3}(2\pi a N_2)^2 + (2\pi a N_\theta)^2\right]}{[1 + (2\pi a)^2 (N_\theta^2 + N_2^2)]^{7/3}} \cos\left(2\pi N_2 \frac{\Delta r_2}{L}\right) dN_2 \tag{31}$$

$$\phi_{22}(N_\theta, \Delta r_2/L) = \frac{4\pi a^2}{3} \int_0^{\infty} \frac{\left[1 + \frac{11}{3}(2\pi a N_\theta)^2 + (2\pi a N_2)^2\right]}{[1 + (2\pi a)^2 (N_\theta^2 + N_2^2)]^{7/3}} \cos\left(2\pi N_2 \frac{\Delta r_2}{L}\right) dN_2 \tag{32}$$

$$\phi_{33}(N_\theta, \Delta r_2/L) = \frac{128\pi^3 a^4}{9} \int_0^{\infty} \frac{(N_\theta^2 + N_2^2) \cos(2\pi N_2 \Delta r_2/L)}{[1 + (2\pi a)^2 (N_\theta^2 + N_2^2)]^{7/3}} dN_2 \quad . \tag{33}$$

In these expressions the evenness of the integrand was used. The above integrals can be evaluated using Filon's method [11]. The results are plotted in figures 9 through 11.

Comparison is made with data collected during the NASA B-57B Gust Gradient Program. The B-57B is instrumented with three component gust probes on each wing tip and at the nose. Figures 12 through 14 show comparison with data collected during a horizontal flight. Figures 15 through 17 depict a comparison for flight on a simulated ILS approach (three degree glide slope). Both measurements were taken within the planetary boundary layer where turbulence is not really isotropic.

In some of the figures, the measured spectra fall between the expected $\Delta r_3/L$ curve and the $\Delta r_3/L = 0$ curve. This result is quite apparent in figure 14, for example. A possible explanation concerns the assumption of the stationarity of the correlation function. In the real atmosphere, the correlation falls off with time as well as with space. This increased fall off in space and time corresponds to a slower falloff in the frequency domain as observed in the figure.

Some of the measured data show an unexpected foot (flattening out) at the higher frequencies. Possible sources of this error include aliasing and the addition of the white noise to corresponding velocity components measured at each wing tip. The correlations would show a Dirac delta function spike which in the frequency domain corresponds to a small but finite constant. When plotted on log paper the spectra would show a characteristic foot. Simultaneous white noise could be introduced by terms from the INS system used to remove the aircraft motions from the data.

The importance of a cross-spectral model is that moment spectra can be calculated from the cross-spectra [13].

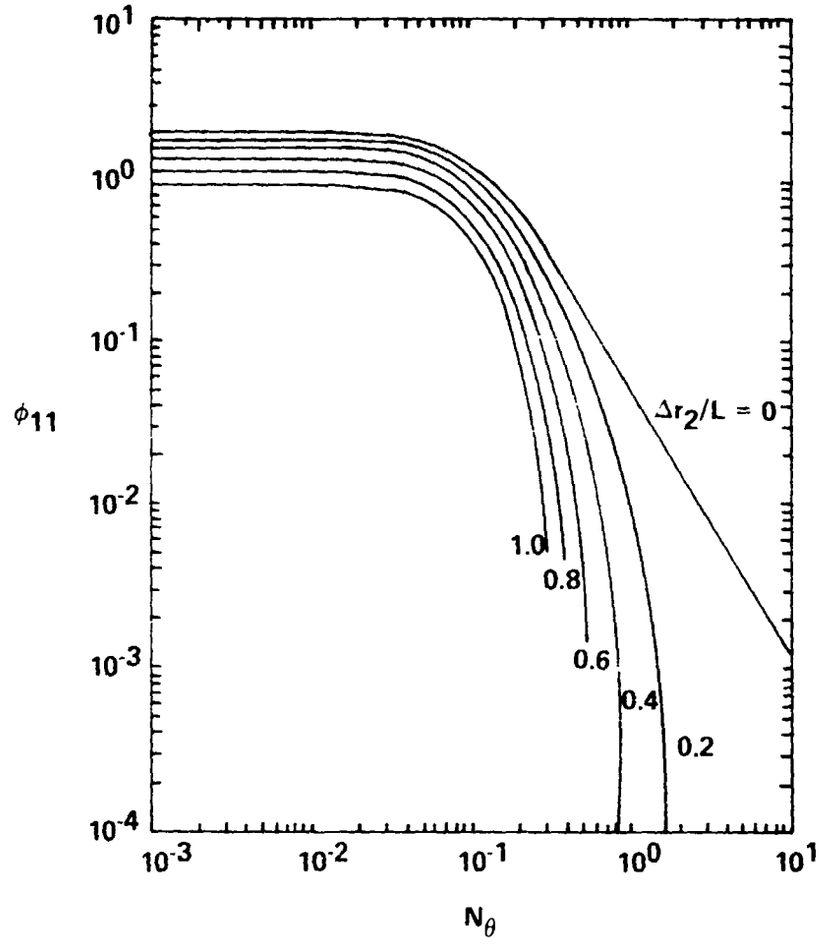


Figure 9. Dimensionless cross-spectra for u-velocity component.

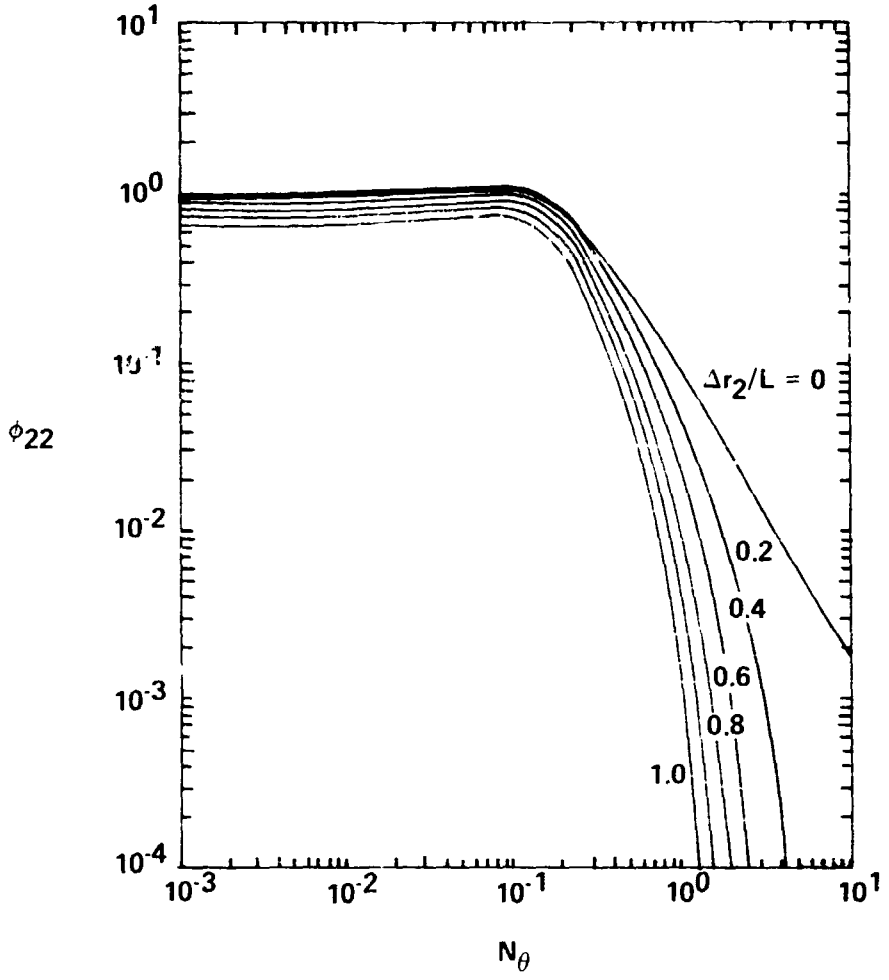


Figure 10. Dimensionless cross-spectra for the V-velocity component.

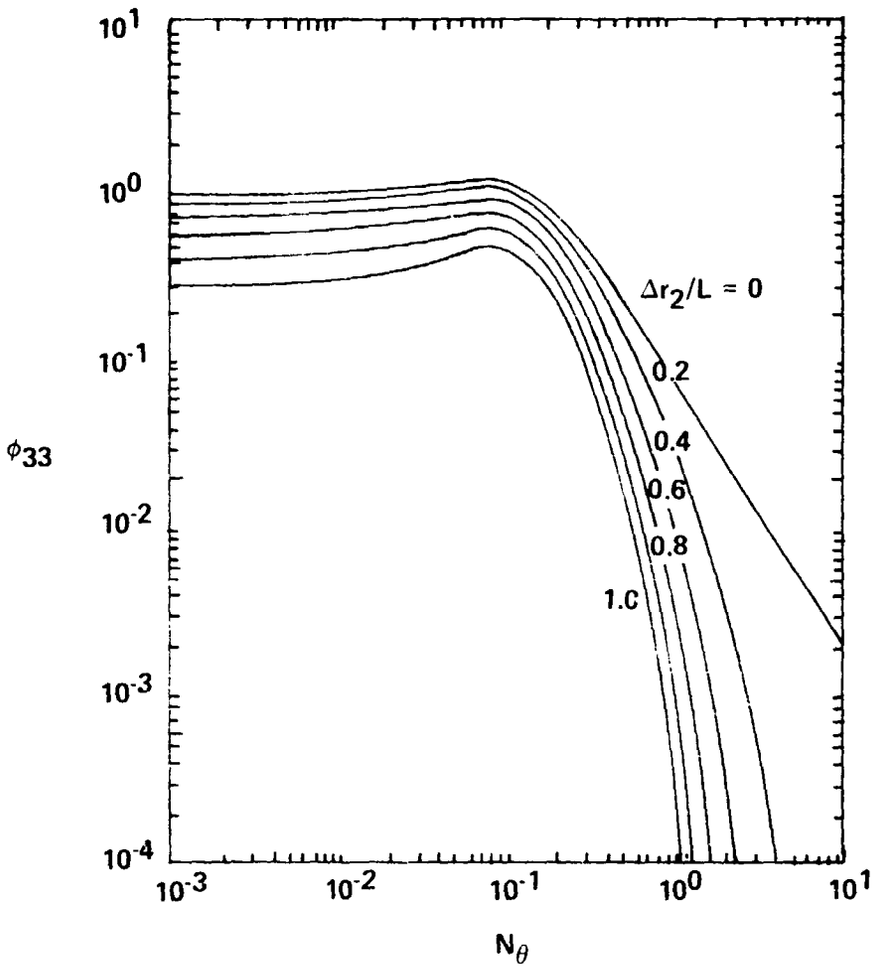


Figure 11. Dimensionless cross-spectra for the w-velocity component.

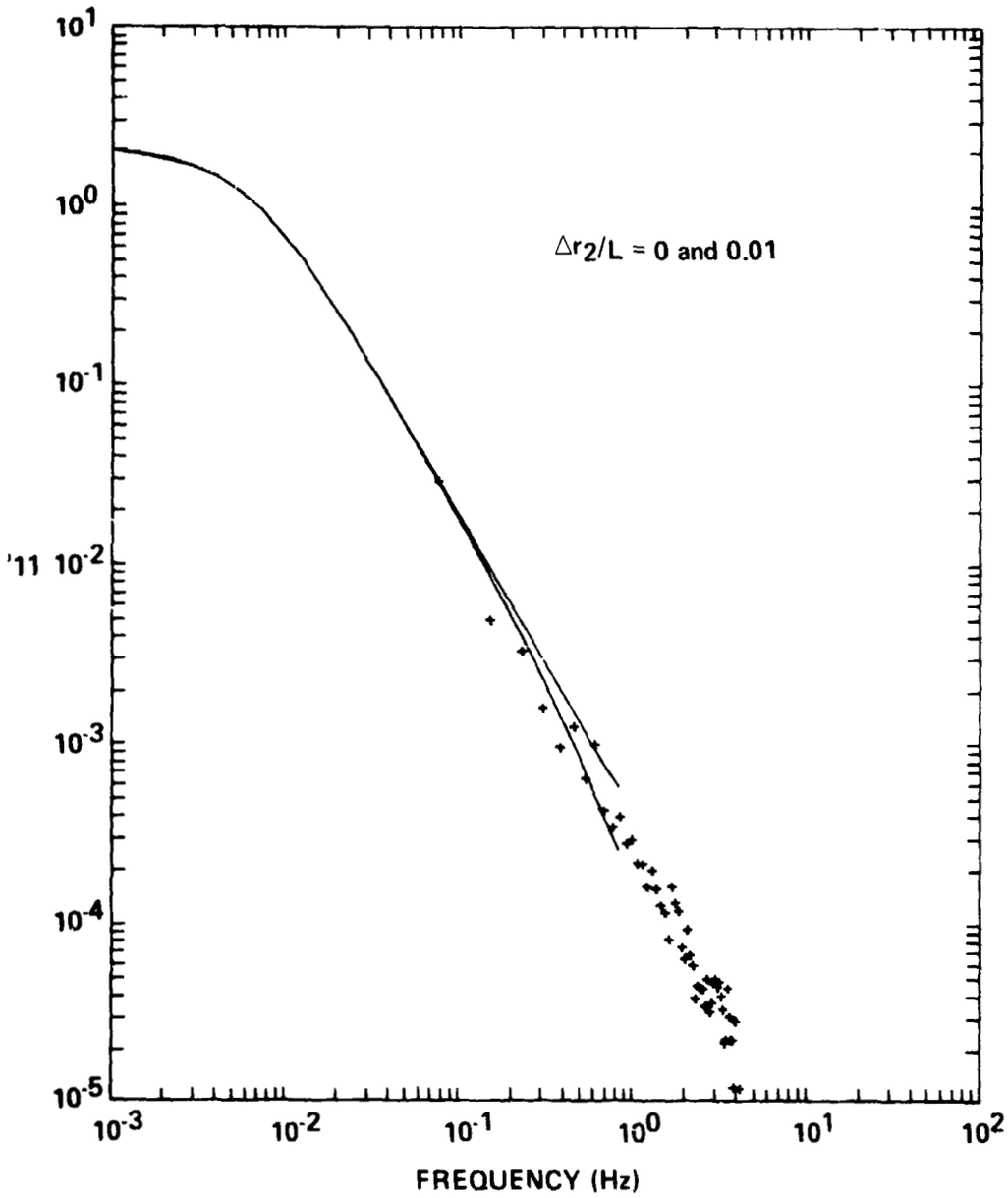


Figure 12. Comparison of computed and measured u-component cross-spectra for a level flight case.

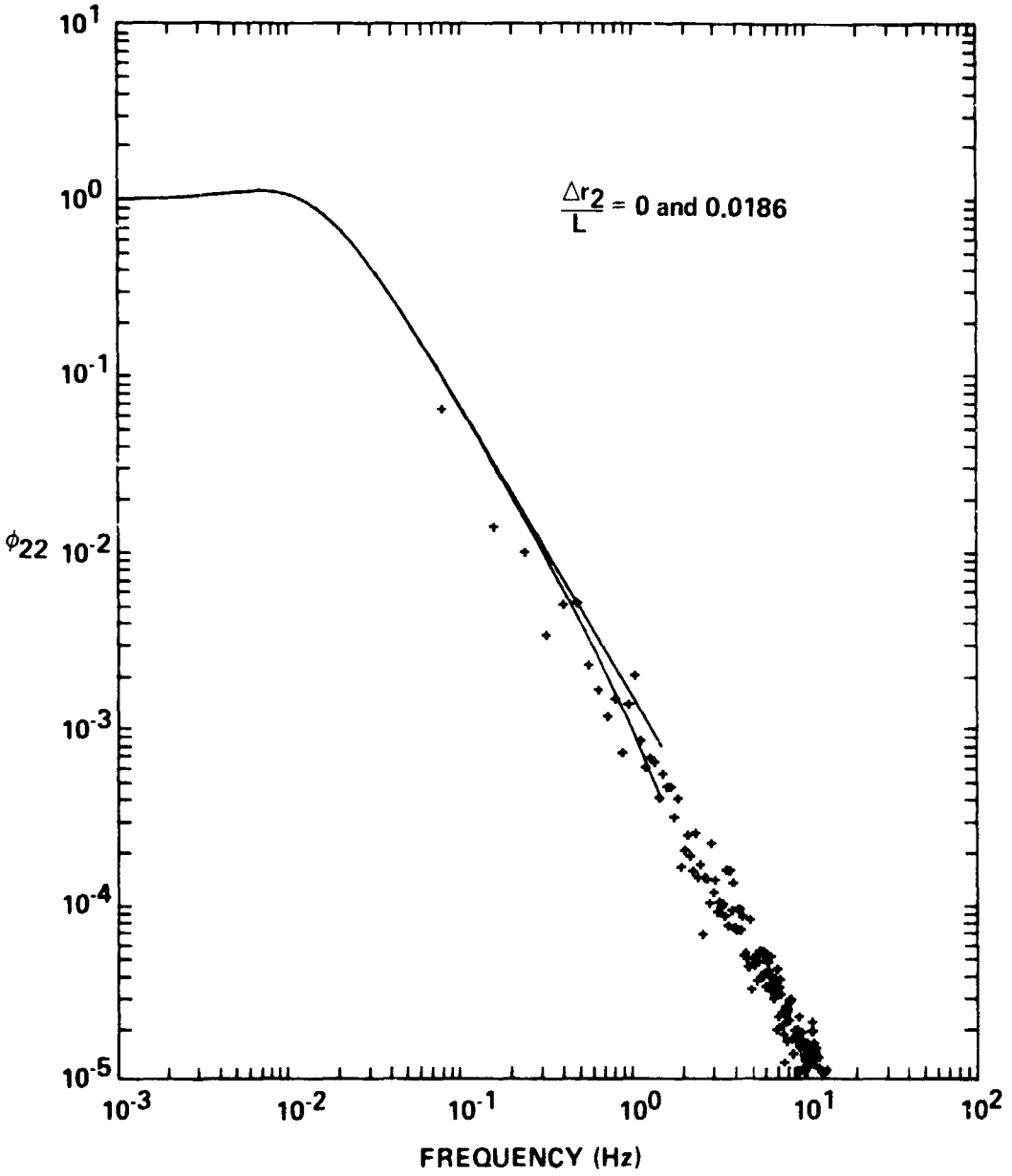


Figure 13. Comparison of measured and computed v-component cross-spectra for a level flight case.

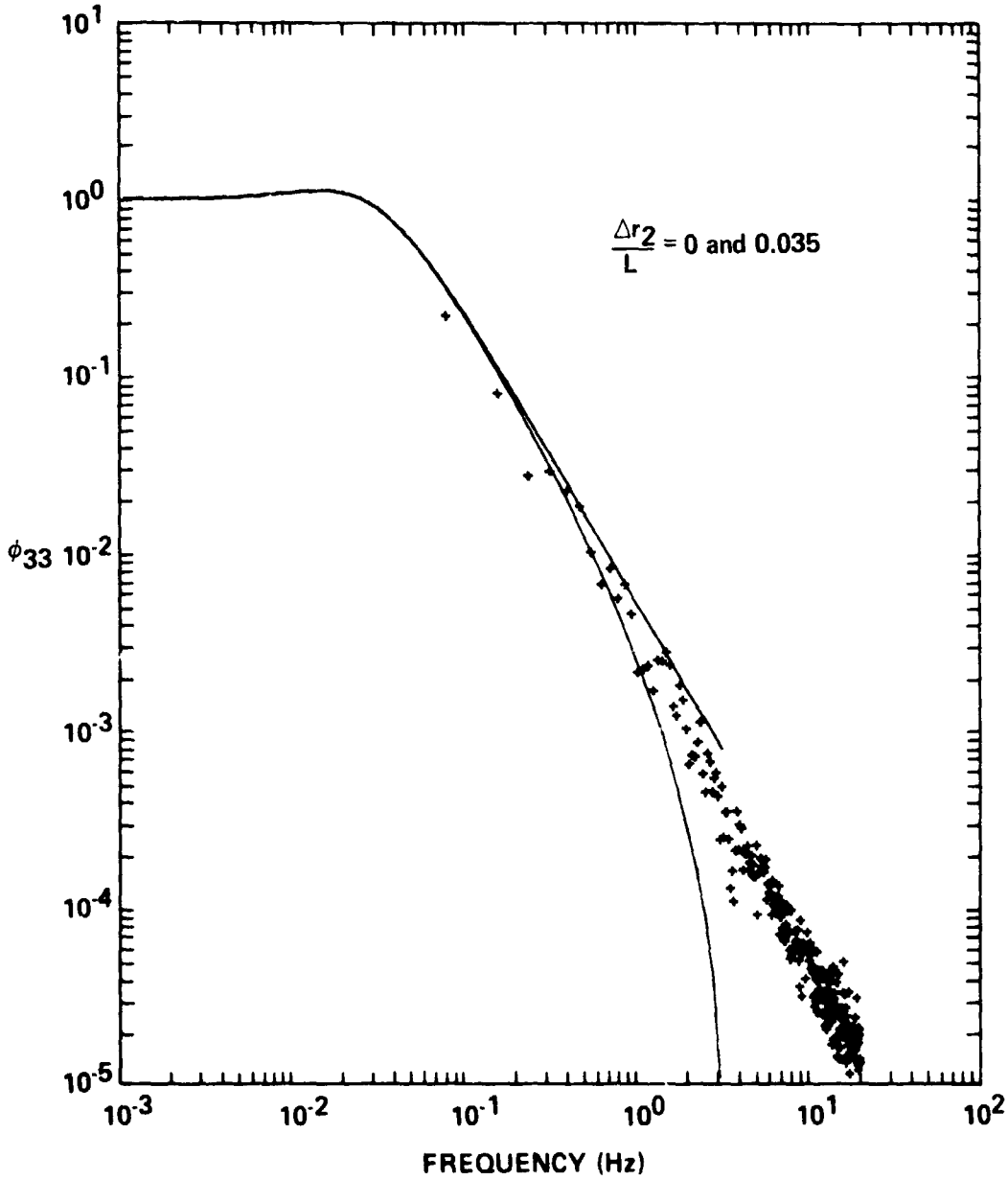


Figure 14. Comparison of measured and computed w-component cross-spectra for a level flight case.

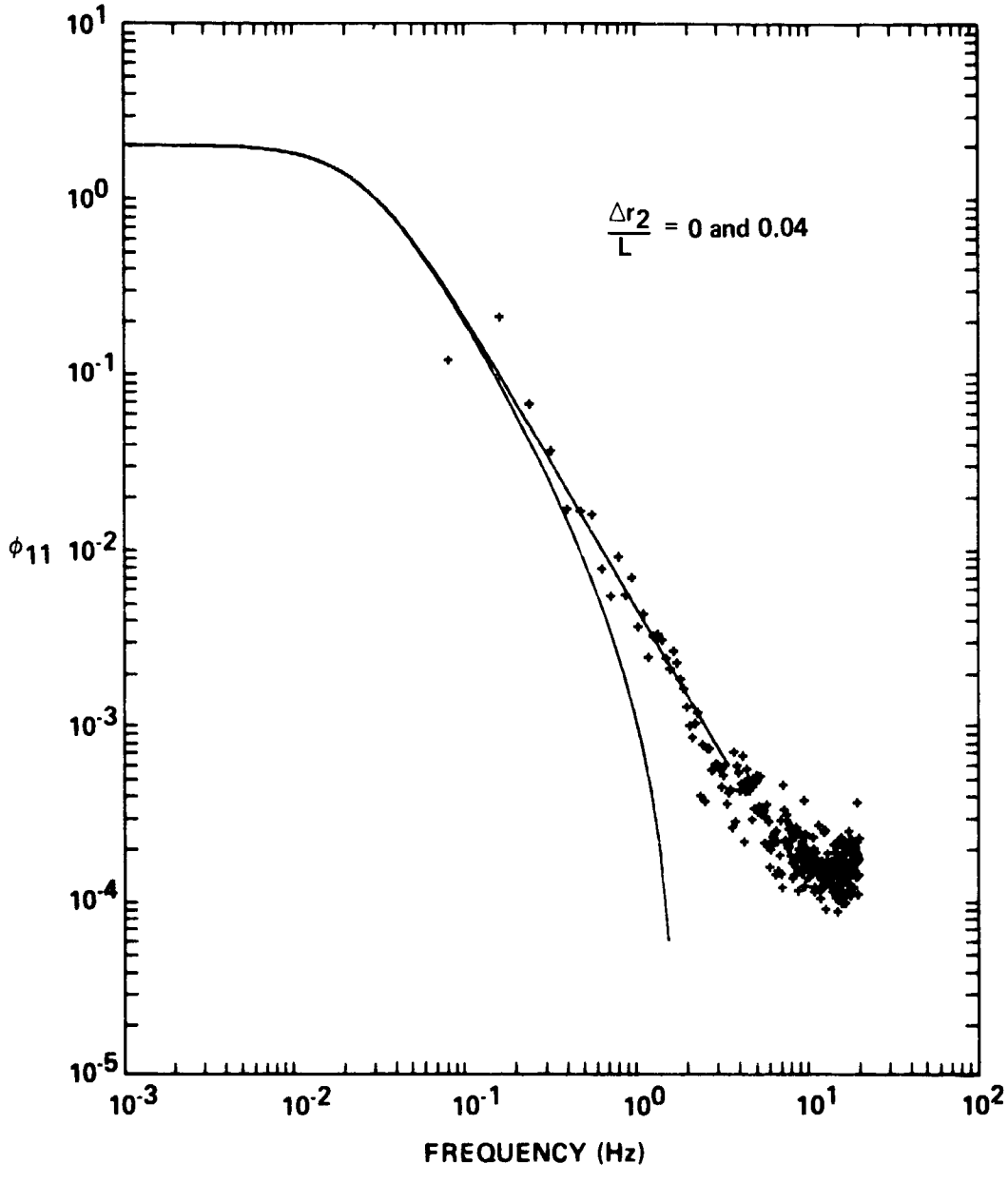


Figure 15. Comparison of measured and computed u-component cross-spectra for the simulated ILS approach case.

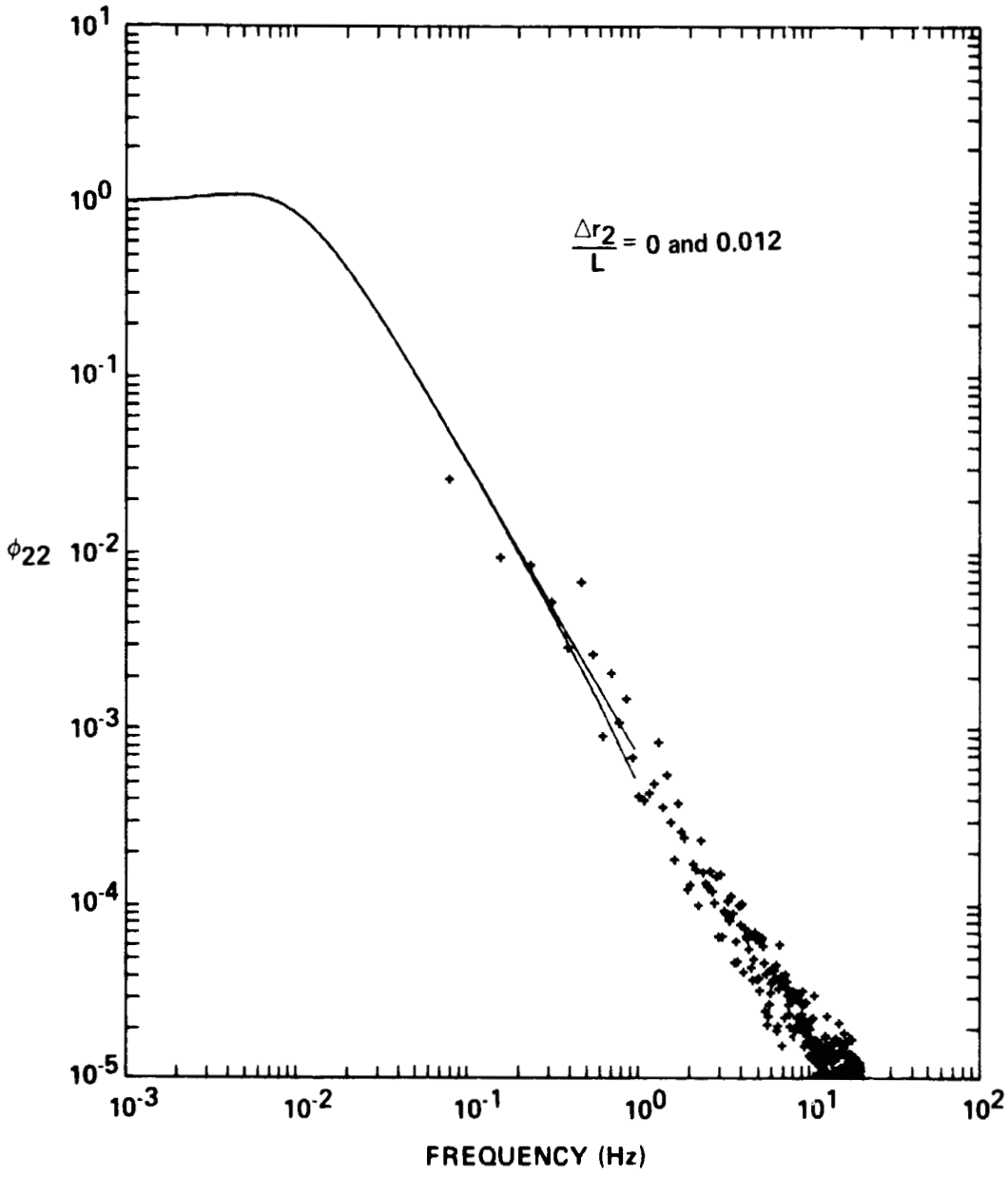


Figure 16. Comparison of measured and computed v-component cross-spectra for the simulated ILS approach case.

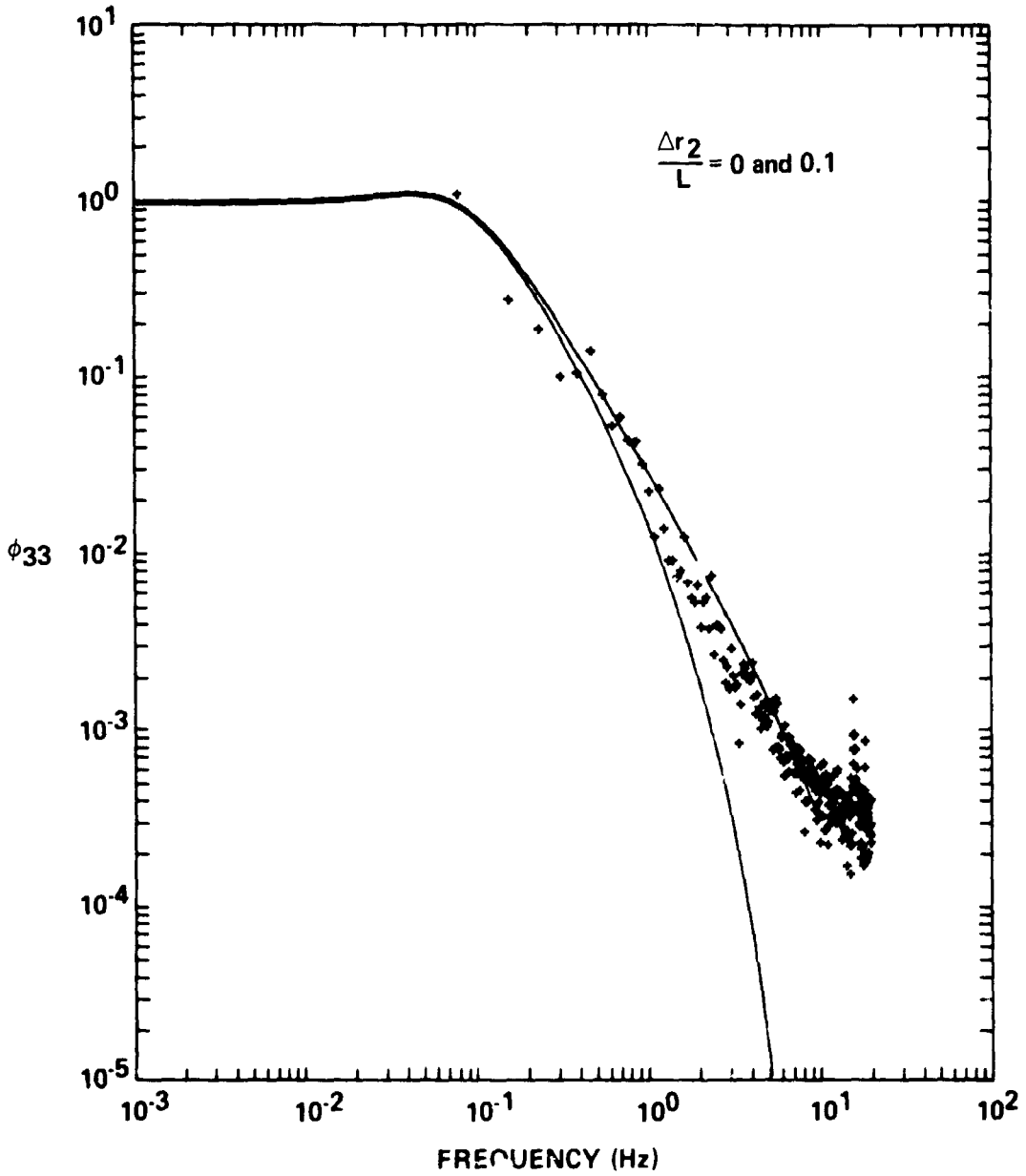


Figure 17. Comparison of measured and computed w-component cross-spectra for the simulated ILS approach case.

CHAPTER III. REVIEW OF MONTE CARLO TURBULENCE SIMULATION

For the purposes of this study, turbulence simulation will mean Monte Carlo turbulence simulation Monte Carlo turbulence simulation should not be confused with efforts to close the time averaged equations of motion. Monte Carlo turbulence simulation is defined as a procedure whereby a noise process is filtered by a linear or nonlinear, analog or digital filter to obtain an output with certain of the statistical properties of turbulence. While Monte Carlo simulation and numerical simulation of turbulence have some things in common, the intent of each is quite different In the case of the former, mean velocity profiles are known and turbulence is to be added to create a realistic flow field. In the latter case mean velocity profiles are to be calculated. Monte Carlo simulation strives to put small scale perturbations back into the flow field while numerical simulation attempts to remove the small scale details. In this sense the two are opposite operations.

Monte Carlo turbulence simulation apparently began in the mid-fifties. The basic idea, which is employed in the present study, is depicted in Figure 18. Gaussian white noise, with Dirac delta function autocorrelation and corresponding constant spectral density is input to a linear filter. The linear filter has a transfer function $H(\nu)$ which is selected to give the desired output spectrum. Because the filter is linear, and the input Gaussian, the output is Gaussian. Using the notation of Figure 18, the output spectrum is

$$\Theta_w(\nu) = |H(\nu)|^2 \, \Theta_n(\nu) \tag{34}$$

In (34), $\Theta_w(\nu)$ is the known, desired form of the output spectrum, and $\Theta_w(\nu)$ is a constant for all frequencies. $H(\nu)$ is the unknown with modulus

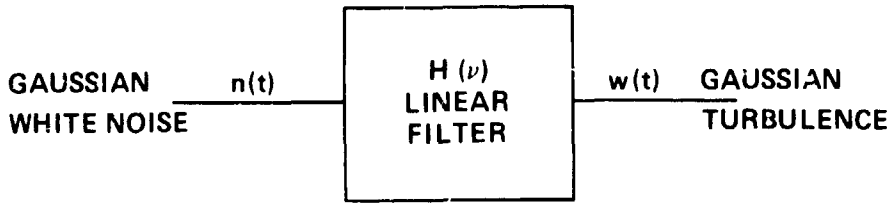


Figure 18. Simple Monte Carlo turbulence simulation.

From Equation (34) the realization of $H(\nu)$ which gives the desired output spectrum is nonunique. In fact, $H(\nu)$ has the general form

$$H(\nu) = |H(\nu)|\, e^{\pm j\alpha(\nu)} \quad . \tag{35}$$

where $\alpha(\nu)$ is the phase of $H(\nu)$. $H(\nu)$ has an infinite number of realizations corresponding to an infinite number of choices for the phase. One choice of phase is $\alpha(\nu) = 0$ for all ν. The choice of the phase does not effect the output spectrum.

Choice of $H(\nu)$ is complicated by the fact that no noise source, neither analog nor digital, is completely white. In the digital case, which is the concern of this report, the designated sampling frequency determines the cutoff frequency of the noise. The nonwhiteness of the noise source results in a factor which is multiplied by the expression for $H(\nu)$. This aspect of the simulation is discussed in more detail in Chapter VI.

Not all turbulence simulations involve the use of linear filters. Reeves and his colleagues in a series of papers [14-16] developed a nonlinear filter model. Figure 19 depicts the block diagram of one version of the nonlinear filter used by Reeves and associates. In the figure, the parameter R can be changed to modify the kurtosis of the output turbulence. In effect R changes the patchiness of the simulated turbulence. The functional forms of the three filters H_a, H_b, and H_c are chosen to give the desired output spectrum.

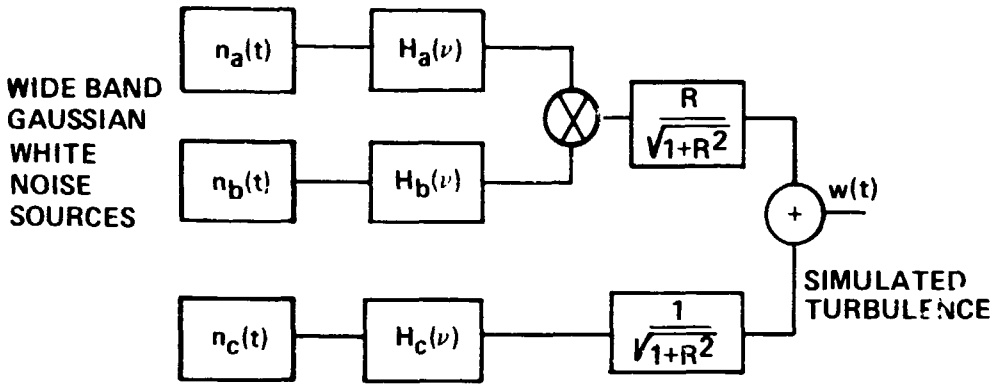


Figure 19. Reeves' non-Gaussian simulation model.

In general, the output form of the probability density cannot be determined for nonlinear filters. Reeves judiciously constructed his filter so that the output probability density function could be calculated.

Another advance in turbulence simulation arose from the need to include interlevel coherence in simulation. Fichtl and Perlmutter [17] developed a multi-filter system to incorporate interlevel coherence. In this model a series of white noise sources is filtered, added together, and the result filtered to obtain the turbulence. A schematic of the method is given in Figure 20. Each of the filters D_{-p} through D_p are effectively height dependent phase shifts. The desired coherence is obtained exactly as the number of filters and noise sources approaches infinity. In practice, the desired coherence can be approximated to any desired accuracy by choosing p large enough. The filter H in Figure 20 gives the desired output spectrum. Since the block diagram of Figure 20 represents a linear filter and since the inputs are Gaussian, the output will also be Gaussian.

For the purposes of this study, one of the more significant advances in turbulence simulation technology was made by Fichtl [18]. In this report, Fichtl generated non-dimensional turbulence. By this approach, one nondimensional turbulence record could be generated, its spectrum checked, and then be used for simulation of any flight profile. Fichtl's method was based on the Dryden spectral model. A similar approach was used by Tatom, et. al. [19] for the von Karman model.

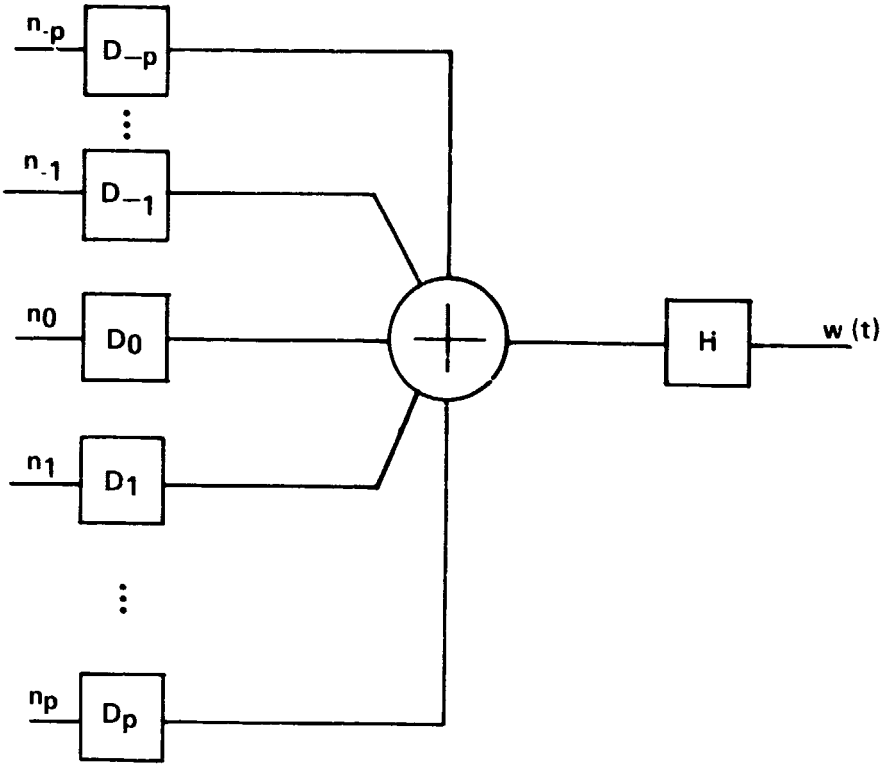


Figure 20. Interlevel coherence model [17].

The preceding discussion was a brief history of some turbulence simulation approaches which have some relevance to the current model. For a broader discussion of Monte Carlo turbulence simulation, the interested reader is referred to three papers by Dutton, et. al. [20], Fichtl, et. al. [21], and Wang and Frost [22]. The report by Dutton, et. al. gives a good historical perspective of turbulence simulation and presents an unusual nonlinear approach. The paper by Fichtl, et. al is a general survey of simulation methods up to 1977. The paper by Wang and Frost presents codes and discussions for several of the methods discussed previously.

Before leaving the area of Monte Carlo turbulence simulation, a brief discussion relevant to the present model is presented. In Chapter II, one-dimensional spectra for the von Karman and Dryden models were given. Real turbulence shows a $\nu^{-5/3}$ rolloff as predicted by Kolmogorov and the von Karman model, however, the von Karman

model lacks the convenience of a rational (ν^{-2} rc' ..) Dryden model. For rational spectral models, the simulated turbulence can be generated with a difference equation of the form

$$w_{i+1} = f_{cn}(w_i, w_{i-1}, \ldots, n_i, n_{i-1}, \ldots) \tag{36}$$

Equation (37) is a digital simulation model where the i+1st turbulence point generated is a function of the previous turbulence points and noise source points. For the Dryden model (37) can be written as

$$w_{i+1} = C_1 w_i + C_2 w_{i-1} + d_1 n_1 + d_2 n_{i-1} \tag{37}$$

where C_1, C_2, d_1, and d_2 are parameters depending on the sampling rate, airspeed, turbulent intensity and length scale of turbulence.

Wang and Frost [22] describe a simple rational model which approximates the more realistic von Karman spectra. This approach can be carried to any level of complexity as long as care is taken so that all poles lie within the unit circle in the Z-transform plane. This stipulation assures system stability.

When a recursion relation such as Equations (36) or (37) can be derived, it is much superior computationally to other available methods. Otherwise the turbulence must be calculated either with a convolution or with a Discrete Fourier Transform (DFT). When turbulence in two or more dimensions is generated, the DFT or convolution must be used because Z transform theory is not well developed in more than one dimension. The only time it can be used is when the function to be transformed is separable so that the two-dimensional Z transform reduces to the product of two one-dimensional transforms. Unfortunately, spectral models are functions of $\nu = (\nu_1^2 + \nu_2^2 + \nu_3^2)^{1/2}$ which is not separable.

In this chapter a brief review of Monte Carlo turbulence simulations was presented. The review was by no means exhaustive, but was confined to those approaches

which are relevant to the current method. With the exception of the paper by Fichtl and Perlmutter [17] none of the reported studies in any way accounted for variations of turbulence in more than one dimension. Fichtl, et. al. [21] discuss in passing the generation of two-dimensional turbulence. For truly realistic turbulence simulation, variation of the turbulence in all three dimensions is necessary. The generation of three-dimensional turbulence is the subject of Chapter VI.

CHAPTER IV. DESIRABLE FEATURES OF A SIMULATION OF ATMOSPHERIC WINDS AND TURBULENCE

Monte Carlo simulation models of atmospheric winds and turbulence have several desirable features. Some of these are listed and discussed in the following paragraphs.

1. Realism: The model should be realistic in the sense that turbulence generated by Monte Carlo methods should have as many of the statistical and spectral characteristics of real turbulence as possible. The turbulence should have the right "feel." A common complaint of pilots who do some of their training on simulators is that the generated turbulence is not realistic. This absence of the correct feel has been attributed to failure to simulate turbulent intermittancy or patchiness [14]. Another factor may be the failure to simulate variation of winds across the body of the aircraft. By this failure, the main contributor to roll and yaw moments are neglected. Six degree of freedom motion simulators (controlled by hydraulic actuators) exist so that roll, pitch, and yaw motions could be simulated if corresponding moment information were available. Finally, a simulation model should contain all frequencies which cause a significant response in the man-aircraft system.

2. Computational Speed: The wind and turbulence model should be computationally efficient so that real time flight simulations are possible. A computer working with a flight simulator must process incoming analog signals, convert digital output signals to analog response, calculate turbulence and aerodynamic loads all in real time. Many operational flight simulators are operated by minicomputers and computational speed is a must for pilot-in-the-loop simulations. Research simulators may have one or more high speed state-of-the-art computers to run them but even these may not be able to handle a real time turbulence simulation if the model is too complex.

3. Flexibility: Several types of flexibility are desirable. First the ability to simulate a wide range of atmospheric phenomena is desirable. The ability to easily implement different microburst events from data sets such as JAWS is required.

Another type of flexibility involves the freedom of the pilot to perform any maneuver desired. Many simulators assume a constant airspeed and commit the pilot to level or glide slope flight. The ability to go around in severe shears is denied. In microburst flights, the airspeed changes can be ±30 m/sec or more. The ability to go around or to test various escape strategies is highly desirable.

The flexibility should extend to the ability to apply the wind and gust model to a wide variety of different aircraft. All frequencies of interest to aircraft response should be in the model. The ability to do this for aircraft of a wide variety of sizes and characteristics implies a nondimensional simulation. The concept of nondimensional simulation is explained in a later section.

4. Easy implementation: Easy implementation implies code clarity, and simplicity, and portability. Portability means the ability to transfer code or data from one computer to another. If these attributes are missing, the method will not be accepted by the aviation community.

While the above list is by no means exhaustive, it does include the major desirable attributes of a gust and wind model.

In creating a simulation, the simulator is in effect creating a world of his own. This shadow world created for engineering purposes should contain enough spectral information so that all factors affecting the phenomenon under study are available. The following sections describe the creation of a turbulence and wind shear "world" which varies in space and time in a realistic manner. In the author's opinion, the differences in the realism of the present model and previously used one-dimensional models are similar to the differences in the creatures inhabiting our three-dimensional world and the one- and two-dimensional inhabitants of Lineland and Flatland [23].

Creating realism for the sake of realism is not the function of engineering simulation, rather the engineering simulation is like a vignette. Relevant features are clear, crisp, and mathematically precise, while features not affecting the simulation are not.

The model described in this document has many of the desirable attributes. Its greatest strengths are realism and flexibility. In these two areas, no currently available wind simulation model can match it. Easy implementation was a goal of this effort. Programs are in FORTRAN and data are stored in easily transportable integer formats. The one possible weakness of the model is in computational speed. The spatial model achieves maximum realism and flexibility but pays some price in speed. Nevertheless, for systems with enough central memory to store both the JAWS data and turbulence, speed should be sufficient to do real time simulations. Some minicomputers satisfy this requirement.

CHAPTER V. OVERVIEW OF THE SPATIAL MODEL

To this point, the aviation hazard posed by microburst-related wind shear was discussed along with the characteristics of the JAWS data sets. Certain aspects of homogeneous turbulence theory relevant to Monte Carlo turbulence simulation were presented. Previously reported turbulence simulation methods and desired characteristics of wind model were discussed. In this chapter all of these threads are pulled together to weave a fabric of computer created reality for the purposes of aircraft design and pilot training and ultimately to save lives.

Creating the final tapestry which achieves the ultimate reality as far as the wind environment is concerned is not a simple problem. The currently proposed technique is an attempt to combine the best available wind shear data set with three-dimensional simulated turbulence. The mixture of reality with simulated reality is the best that can be done within the current state-of-the-art.

The need for the addition of turbulence is documented in this chapter. A recipe for combining measured data and JAWS data is explained, along with the method of implementation. The proposed technique is related to some similar one-dimensional approaches to data analysis and relevant concepts from these simpler cases are introduced. Finally, while the JAWS data sets are the best available in the world, they lack some information which is necessary for realistic flight simulation. Methods for deriving the required parameters from aircraft measurements are presented.

A typical grid spacing for the JAWS data is 200 m whereas a desired grid spacing for the calculation of aerodynamic moments is 10 meters. In Figure 21 a comparison is made between the JAWS grid size and various aircraft. The box surrounding the planes represents vertical plane grid spacing for the July 14 case (200 m x 150 m). Figure 22 shows a desired grid spacing (10 m) compared to the Boeing 747 aircraft.

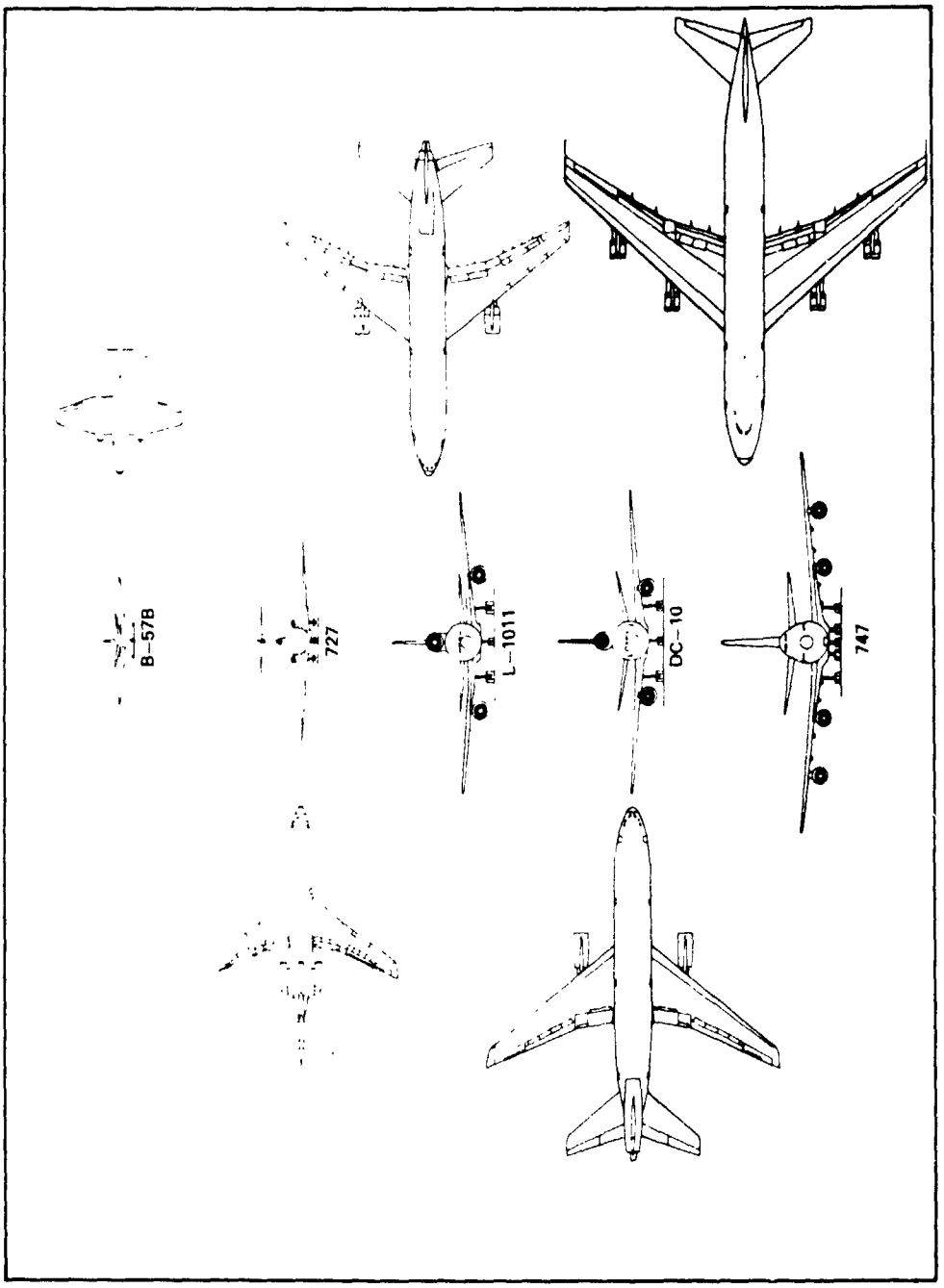


Figure 21. Comparison of JAWS July 14 grid size with various aircraft.

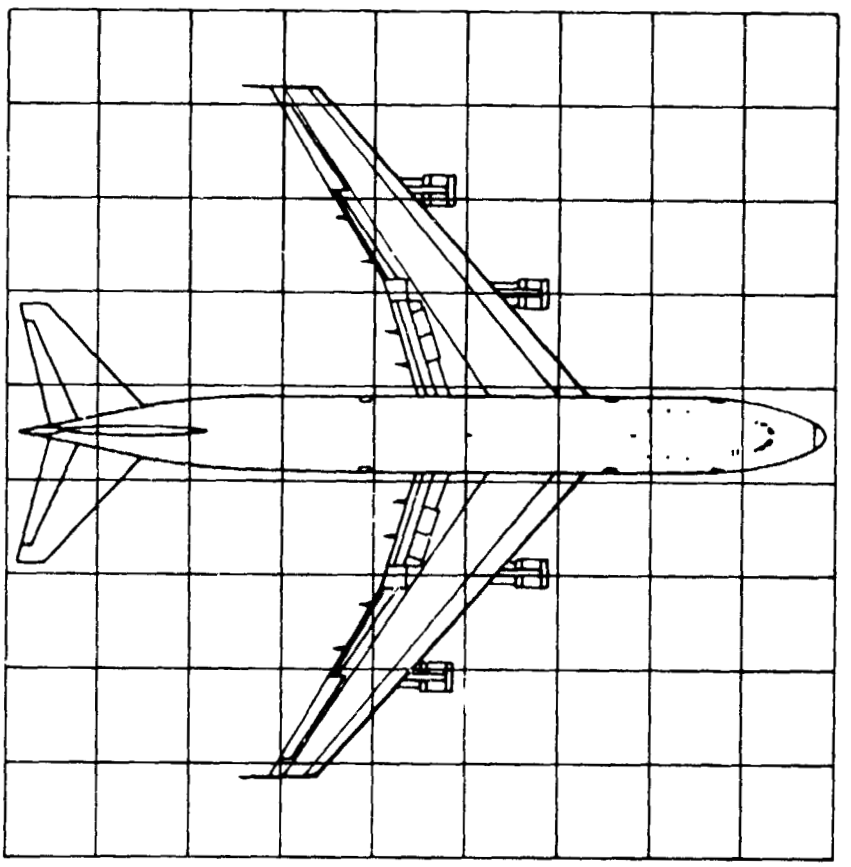


Figure 22. Desired grid spacing compared to the Boeing 747 aircraft.

The spatial sampling frequency ν_s is 1/200 m^{-1} and the resulting Nyquist spatial frequency is $\nu_{Ny} = \nu_{s/2}$ = 1/400 m^{-1}. For an aircraft traveling at 80 m/sec this corresponds to a Nyqiist spatial frequency of f_{Ny} = 0.2 Hz. Typically, the frequency of maximum aircraft response occurs at about 0.1 Hz. According to Etkin [11], structural response of an aircraft extends over a spatial frequency range from about 10^{-4} m^{-1} to 10^{-2} m^{-1}. Short period response occurs over a range from about 10^{-5} m^{-1} to 10^{-3} m^{-1}. Phugoid response cuts off at less than 10^{-4} m^{-1}. The JAWS data contain Phugoid response frequencies, but only part of the short period response frequencies, and even less of the structural response frequency range. With regard to frequencies (airspeed = 80 m/sec), the JAWS cutoff frequency was 0.20 Hz while structural response goes up to near 10 Hz. Some modelers include structural bending modes in their simulators because they feel these modes add the correct "feel" to the turbulence. An aircraft structure encountering turbulence has a ringing response [24]. The conclusion is that since structural and short term response of aircraft to turbulence are important for realistic simulation, and since the complete frequency range is not contained in the JAWS data, the high frequency turbulence must be added.

The obvious question arising from the above discussion is, "how can turbulence be realistically added to JAWS and other data sets?" The addition of these high frequencies are subject to several constraints. First, the model chosen is restricted to using mainly information available from the data set itself. For maximum realism, as little of the required information as possible should be generated from "rules of thumb" or from model equations. The available information includes the three velocity components (over the low frequency range) and spectral width which can be related to a gust standard deviation. Based on this available information the model to be discussed below was developed. The "glue" that holds the model together is the following equation.

$$u_i(x,y,z,t) = \bar{u}_i(x,y,z,t) + \sigma_i(x,y,z,t)\, w_i(x,y,z) \quad . \tag{38}$$

where $u_i(x,y,z,t)$ are the simulated winds, $\bar{u}_i(x,y,z,t)$ are the low frequency "smoothed" winds, $\sigma_i(x,y,z,t)$ are the gust standard deviations, and $w_i(x,y,z)$ are the zero mean, unit standard deviation, frozen turbulent velocities.

The model defined in the above equation is written in its most general form but is specifically tailored for JAWS-type data sets. u_i are the three components of low frequency wind contained in the data sets. σ_i, while written in vector form, in practice is isotropic and a scalar, σ. It can be derived from JAWS second moment information. Notice that u_i and σ_i are written as functions of three spatial coordinates and time. As of this writing, time has not been included in the JAWS data sets. The reason for this is that the time variation may not be important for flight simulation because of the rapid aircraft transit times of wind shear phenomena. For this reason, the JAWS Project has concentrated its resources on other matters.

The final term in Equation (38) is the frozen turbulence data base; frozen because w_i is not a function of time. A three-dimensional block of Monte-Carlo simulated turbulence is created and effectively stacked inside the JAWS data set in order to add the necessary small length scale phenomena to the coarser gridded JAWS data. In the current method the three-dimensional block is assumed to be isotropic turbulence. There is some basis for this assumption, since much of the anisotropy of the winds will be contained in $\bar{u}_i$. w_i is assumed isotropic for convenience. The Monte Carlo simulation of nonisotropic turbulence is not far advanced in the one-dimensional case. In the three-dimensional case it is nearly nonexistant. Etkin, in his classic text [23], recommends for low altitudes (anisotropic turbulence) the three isotropic one-dimensional spectral functions be used with corresponding gust intensities. This is certainly a practical approach in the one-dimensional case, but the corresponding three-dimensional spectra-functions are not available. At least one group of investigators [25] have developed axisymmetric three-dimensional spectrum functions. These functions may prove useful in the near future, but for the present an isotropic model was used.

A question arising out of this discussion is, "Why use a three-dimensional turbulence model?" The reason is that by generating three-dimensional turbulence, all lateral, vertical, and longitudinal correlations are included in the data. Implicitly, when a one-dimensional simulation is done, the assumption is made that the aircraft is a point immersed in turbulence as is depicted in the left half of Figure 23. Turbulence is more complicated than this simple picture and two-dimensional turbulence is depicted in the right half of the figure. In fact, the spanwise variation of gusts can be quite high and gust differences measured in the B-57B Gust Gradient Program exceeded 10 m/sec (20 kts). The B-57B has a 20 meter wing span and its size is compared to some wide-bodied transport aircraft in Figure 21. The Boeing 747 has a wing span of 60 meters. The impact of spanwise variation of gusts is under study, but intuitively it seems that the variations have a significant impact on aircraft response.

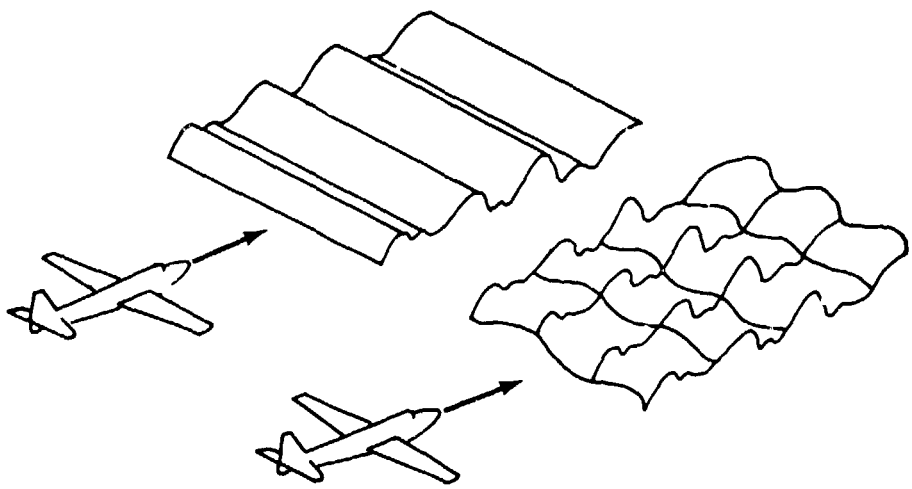


Figure 23. Assumptions of turbulence simulation.

In Chapter II a cross-spectral model was developed. The block of three-dimensional turbulence contains the cross-spectral model as a natural subset of its properties assuming the three-dimensional von Karman spectrum is used to generate the turbulence.

The model as proposed in Equation (38) was in part inspired by a series of papers by Mark, and by Mark and Fischer [26-30]. In these papers a one-dimensional turbulence model defined by the following equation was investigated.

$$w(t) = w_s(t) + w_f(t) = v_s(t) + \sigma_f(t)\, z(t) \quad , \tag{39}$$

where $w_s(t)$ is the slowly varying part of an aircraft-measured velocity, $w_f(t)$ is the high frequency rapidly varying part of the velocity trace, z(t) is a Gaussian unit variance process (von Karman one-dimensional spectrum), and $w_s(t)$, $\sigma_f(t)$, and z(t) are mutually independent processes.

Mark points out that frequently a smoother knee than expected is measured in atmospheric turbulence spectra. He attributes this to the effect of a slowly varying gust intensity $\sigma_f(t)$. The product $\sigma_f(t)z(t)$ corresponds to a convolution in the frequency domain and if σ_f is a well-behaved function (no spikes in its spectrum), then the spectrum of the product will be a smoothed version of the spectrum of z. The result is a rounded knee in the spectrum of the product.

Mark (1981) shows the effect on the autocorrelation function made up of parts as indicated in Equation (39), and presents data to support his model [29]. Figures 24 and 25 are drawn from this report. In Figure 24, σ_{w_s} is the standard derivation of the slowly varying part, $w_s(t)$. The high frequency portion, $w_f(t)$, contributes a rapidly decaying term to the autocorrelation and $w_s(t)$ contributes a slowly decaying term. In Mark's model, $w_s(t)$, $\sigma_f(t)$, and z(t) are presumed independent processes. In Figure 25 a corresponding measurement is presented. In the frequency domain the role is reversed. The spectrum of $w_s(t)$ will decay rapidly while the spectrum of $w_f(t)$ decays slowly.

The summary of the analyses by Mark and by Mark and Fischer was for one-dimensional turbulence measurements, but most of the results carry over to the three-dimensional model of Equation (38). A notable exception concerns the mutual independence of u_i, σ_i, and w_i. The mutual independence assumption was made by Mark to facilitate his analysis, but in fact $\bar{u}_i$ and σ_i should be related. We expect this because

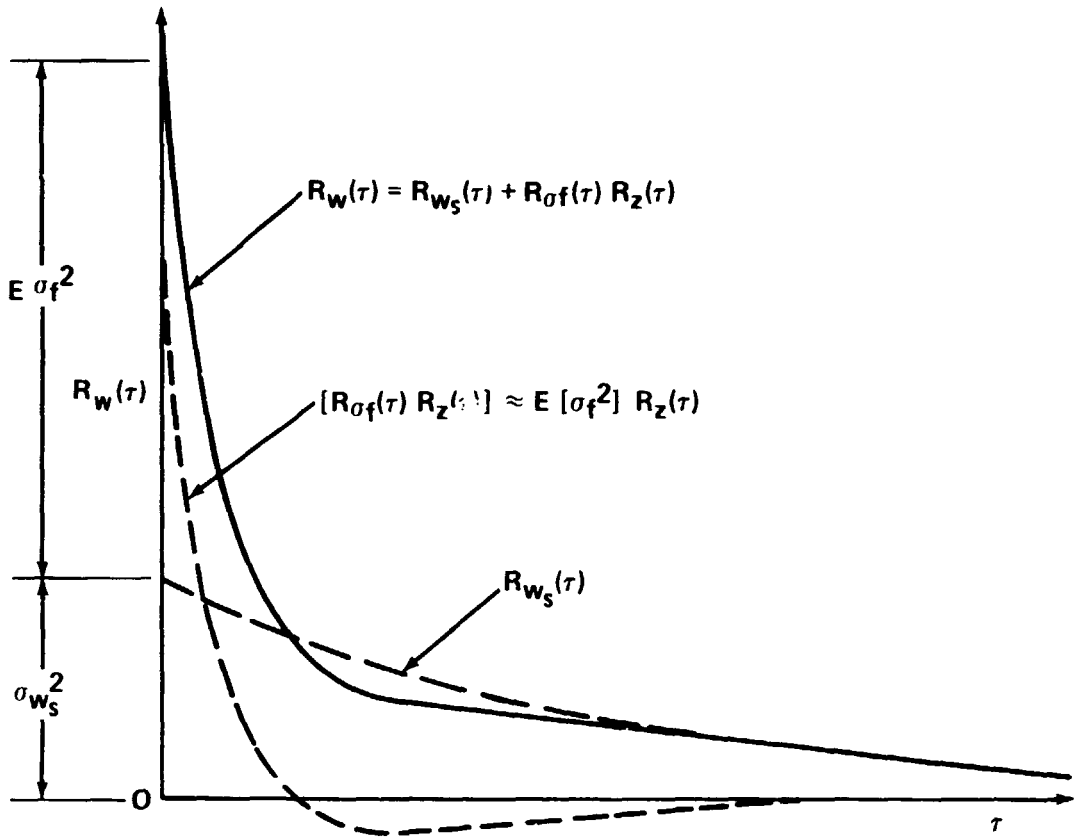


Figure 24. Idealized sketch of auto-correlation functions of atmospheric turbulence and auto-correlation function of its components.

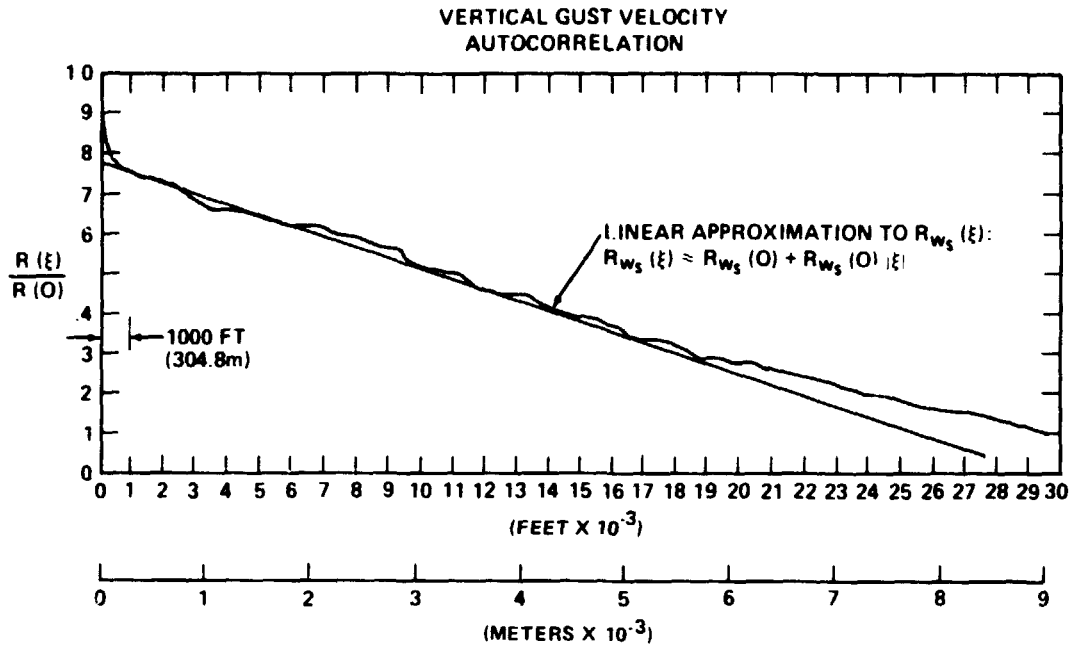


Figure 25. Auto-correlation function of vertical velocity record in mountain wave conditions (airspeed 197 m/sec).

of mechanical production terms in Reynold's equations involving gradients in $\bar{u}_i$. The relationships may not show up in linear statistical relationships, however. In any case, the independence assumption is not required in the present model.

The terms in Equation (38) were discussed above, and now a discussion concerning the assembly of the three terms into a wind model is presented. For the JAWS July 14 data set the grid consisted of 60 x 60 x 11 points. At each of these points is a measured value of the three velocity components, and spectral width. This corresponds to 158,400 data points. These points were measured on a 200 m x 200 m x 150 m grid. A desirable spectral resolution for the turbulence is 10 m x 10 m x 10 m. Generation of a block of turbulence 12 km x 12 km x 1.5 km with 10 m resolution corresponds to 6.5×10^8 data points and is not feasible. Instead, a relatively small block of turbulence can be generated and effectively stacked to fill the JAWS volume. Stacking is equivalent to moving the block around within the JAWS volume as the plane flies through and begins to leave the turbulence. The situation brings to mind a small boy playing with

a caterpillar. As the caterpillar hangs over the edge of his hand the boy obligingly offers his other hand and the worm's journey of exploration continues. In his travel across each hand the caterpillar is unlikely to retrace the exact path of his previous journey because he begins his journey at a slightly different point each time and travels with a slightly different heading. The cycle continues as long as the boy desires and the result is that the caterpillar has his exercise and the boy has the pleasure of his company.

The periodic shifting of the data base is equivalent to stacking blocks of turbulence within the block of JAWS data. Two ways of stacking the blocks come to mind. As the aircraft of the simulated flight passes out one side of the block its motion can be reflected back into the block of turbulence. This procedure is similar to video games in which the electronic ball bounces off the wall (angle of incidence equals angle of reflection). This method, through complicated has the advantage that the aircraft encounters no abrupt discontinuities in the turbulence field. This approach is equivalent to stacking two types of blocks in a special pattern (see Figure 26). The two types of blocks are reflections of each other, the same as left and right hands. The blocks are stacked so that similar sides touch each other.

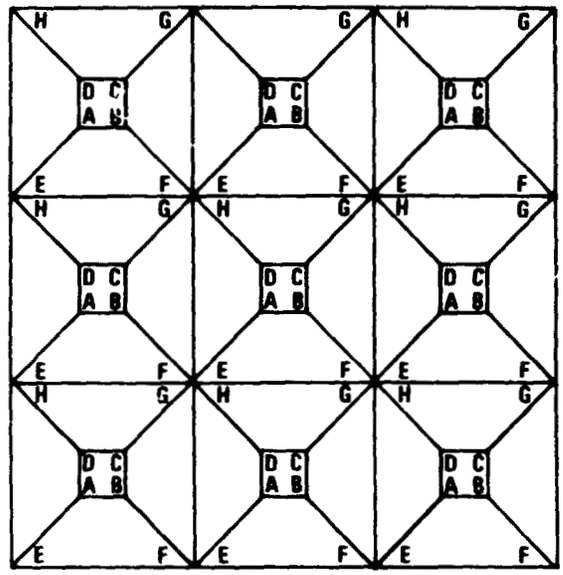
Computationally a simpler approach is to stack one type of block on itself with the same orientation. Position within the block is calculated using congruence arithmetic.

$$X_T = X \bmod (X_{Tmax})$$

$$Y_T = Y \bmod (Y_{Tmax}) \qquad (40)$$

$$Z_T = Z \bmod (Z_{Tmax}) \quad .$$

In this equation, X_T is the X-position within the block of turbulence, X is the real X-position in space, X_{Tmax} is the maximum extent of the frozen turbulence, and so on. This expression means, divide X by X_{Tmax}, determine the remainder which is X_T. If X = 6500 m and X_{Tmax} = 600 m then X_T = 500 m. The value of X_T always lies



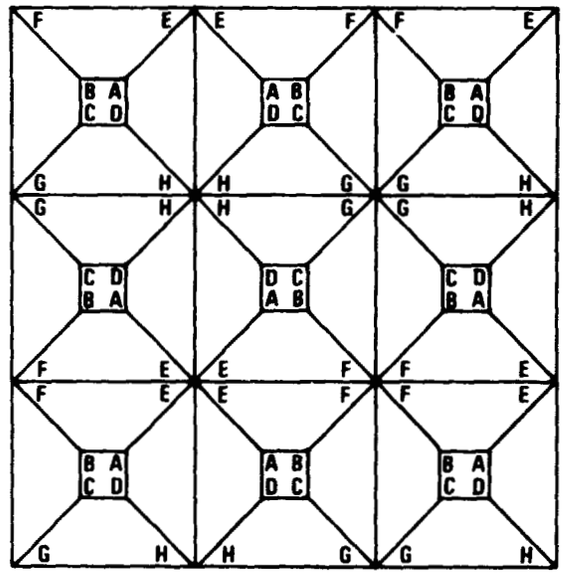


Figure 26. Stacking blocks of frozen turbulence.

between 0 and X_{Tmax}. This method is equivalent to video games in which the electronic "projectile" disappears from one side of the screen and reappears instantaneously on the opposite side. An anticipated disadvantage of this technique is a discontinuity in the turbulence as the plane passes from one side of the block to the other. In practice, the discontinuity is small or nonexistent so the computationally simpler technique of Equation (40) can be used.

One caution on the use of the above equation should be discussed at this time. The frozen turbulence data base generated for use with JAWS data is nondimensional in space. Transformation of a complete block of turbulence must be achieved at one time with one length scale of turbulence. At the same time, the turbulence length scale should vary through the simulated atmosphere. The most commonly studied variation is with height above ground. The presence of microbursts in the planetary boundary layer (PBL) also creates a lateral variation of length scale in the atmosphere. Convective storms imbedded in planetary scale flows have been observed to create an obstruction to the larger scale flow and even shed vortices. Apparently convective obstructions affect turbulent scale lengths in the atmosphere. There is every reason to expect that microbursts which are embedded in larger scale flow in the PBL like chocolate candy kisses will similarly affect length scales in the PBL. In order to account for desired changes in this length scale, the turbulence is generated in dimensionless space. In dimensionless space the block size is constant, but in dimensional space the block size changes isotropically with the length scale. In the space domain an increasing turbulent length scale corresponds to increasing block size, and in the spatial frequency domain a decreasing block size. The problem with indiscriminant use of Equation (40) is that the blocks are stacked on top of each other and as each contracts or expands the whole stack similarly contracts or expands. If the stack contracts along its lower left hand corner the net motion at the right side of the stack is multiplied by the number of blocks in between. The result is an unreal change in aircraft position relative to the turbulence.

Care must be taken that stack contraction is about the location of the aircraft and this is easily implemented. The nondimensional generation of turbulence mentioned above will be described in detail in the following chapter.

The preceding discussion brings us to the one parameter not available from the JAWS data set, i.e., the length scale of turbulence. The distribution of this parameter must be either modeled, or derived from aircraft experiments or from laboratory measurements. This desired length scale is not the overall length scale but rather the length scale associated with $w_i(x,y,z)$ in Equation (38).

Interestingly, some evidence suggests that the spanwise length scale increases through a microburst [31]. The relevant information is presented in Figure 27. Shown are the three velocity components measured at the center boom of the B-57B aircraft during a suspected microburst encounter. The encounter occurred about 78 seconds into the run when a sudden sharp headwind increase was observed. This increase corresponds to the sudden decrease in the longitudinal velocity component, u. During this period the B-57B encountered a 15 m/sec headwind increase over a distance of about 130 m. This drastic windspeed change is believed to occur as the aircraft crossed a microburst front. Aircraft altitude was about 400 m AGL and from Chapter II we recall that the typical outflow depth is 600 m. The increasing headwind was followed by a decreasing headwind and then a tailwind all over a period of about 20 seconds. The feature had an extent on the order of 2 km which is consistent with the microburst hypothesis. The horizontal wind vector shifts were associated with a strong (10 m/sec) downdraft which reinforces the microburst idea.

The interesting part of the figure is the lower three graphs which depict wingtip to wingtip velocity differences. The amplitudes of the velocity difference traces increase gradually from the start of the run but then decrease in all three components through the suspected microburst. This unexpected result can arise from three possible sources: (1) the gust intensity decreases through the microburst, (2) the lateral length scales

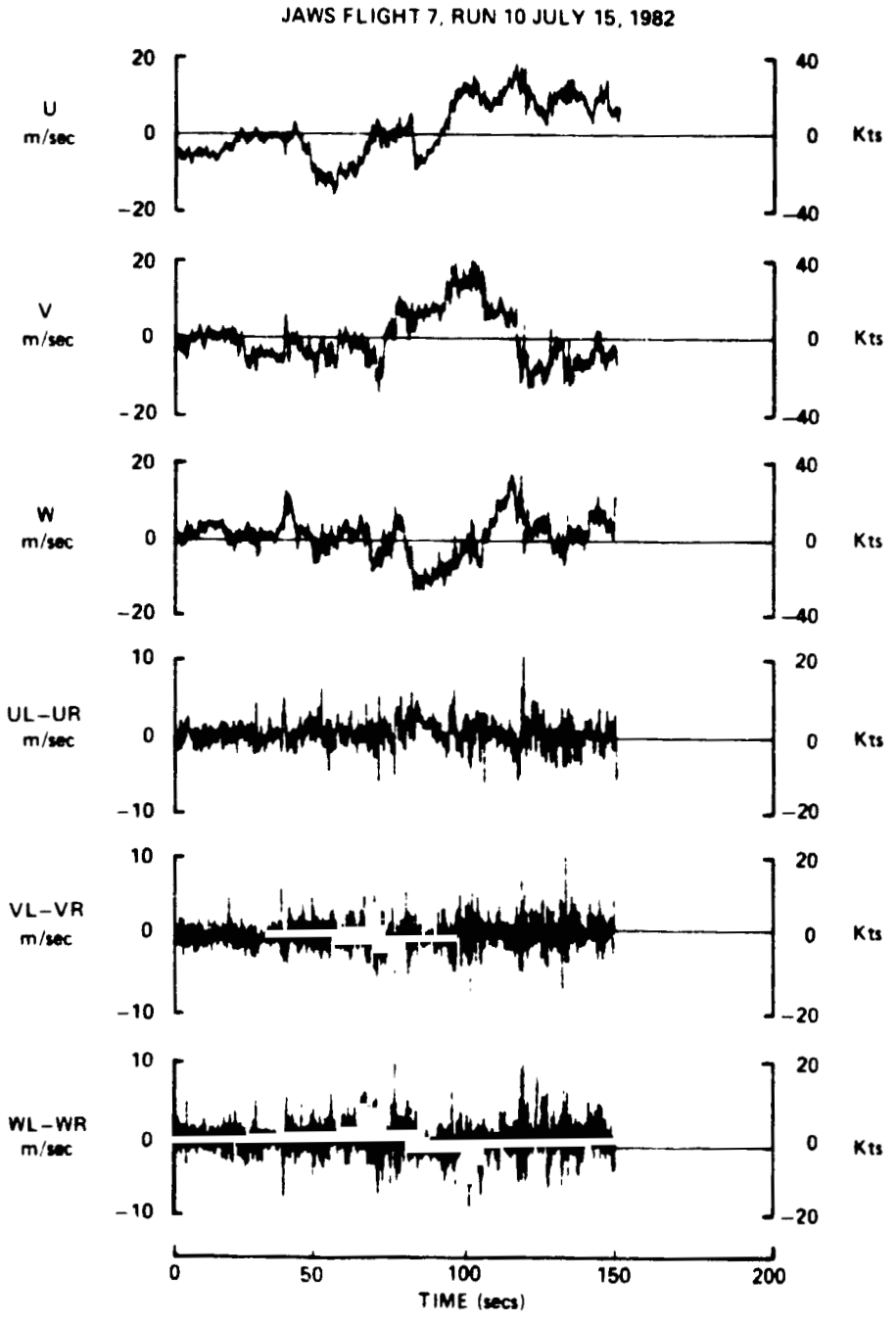


Figure 27. Gusts and wingtip to wingtip gust differences for JAWS Flight 7, Run 10.

increase through the microburst, or (3) a combination of (1) and (2). From the traces at the top of the figure, it is difficult to believe that an intensity decrease is the sole source of the velocity difference decrease. The conclusion then would be that a length scale increase occurs through the microburst. Some caution must be exercised in drawing conclusions from one flight through a microburst, but this type of information is vital for modeling distributions of L.

The problem with obtaining L distribution from aircraft measurements is the difficulty of finding microbursts to fly through even with Doppler radars to guide the aircraft. At the same time, more information is required. Laboratory experiments may help provide the much needed information. Means of obtaining this information in the laboratory are discussed in Appendix A.

The dearth of knowledge on L distributions is similar to the situation turbulence modelers found themselves in prior to the development of two-equation turbulence models [32]. Mixing length distributions had to be specified based on experience. Finally two-equation models were developed which included a partial differential equawion which effectively specified distributions of the mixing length. The close analogy with the present problem suggests that a similar approach to the present problem might be attempted. The most widely utilized two-equation model is the k-ϵ model which contains transport equations for turbulent kinetic energy $k = 1/2\ (\overline{u^2} + \overline{v^2} + \overline{w^2})$ and turbulent kinetic energy dissipation ϵ. The equations for this model are presented by Launder and Spalding [33] in the following form.

$$\frac{Dk}{Dt} = \frac{1}{\rho}\frac{\partial}{\partial x_k}\left(\frac{\mu_{eff}}{\sigma_k}\frac{\partial k}{\partial x_k}\right) + \frac{\mu_{eff}}{\rho}\left(\frac{\partial \bar{U}_i}{\partial x_k} + \frac{\partial \bar{U}_k}{\partial x_i}\right)\frac{\partial \bar{U}_k}{\partial x_i} - \epsilon \qquad (41)$$

$$\frac{D\epsilon}{Dt} = \frac{1}{\rho}\frac{\partial}{\partial x_k}\left(\frac{\mu_{eff}}{\sigma_\epsilon}\frac{\partial \epsilon}{\partial x_k}\right) + \frac{C_1\,\mu_{eff}}{\rho}\frac{\epsilon}{k}\left(\frac{\partial \bar{U}_i}{\partial x_k} + \frac{\partial \bar{U}_k}{\partial x_i}\right)\frac{\partial \bar{U}_i}{\partial x_k} - \frac{C_2\,\epsilon^2}{k} \qquad (42)$$

where $\mu_{eff} = C\mu\rho k^2/\epsilon$ and recommended values for curve fit "constants" are

$$C_\mu = 0.09,\ C_1 = 1.44,\ C_2 = 1.92,\ \sigma_k = 1.0,\ \sigma_c = 1.3 \quad .$$

The length scale of turbulence is defined by

$$L = C_D\, k^{3/2}/\epsilon \tag{43}$$

Here C_D is another constant of proportionality.

Observe that if JAWS estimates of $k = 3/2\ \sigma^2$ based on spectral widths are used then Equation (40) need not be solved. If the $\bar{U}_i$ in Equations (41) and (42) are assumed to be the JAWS velocities, appropriate difference forms of Equation (42) can be developed and with appropriate boundary conditions for ϵ and with the aid of Equation (43), a distribution of L could be calculated. In reality the "constants" in the above equations vary from one flow to another and would need to be defined for microburst type flows. The above equations were developed by making certain assumptions which may not be appropriate for the JAWS microburst situations. Nevertheless, the above provides an engineering approach to a difficult problem. The method can be refined and "tuned" if experimental data are available to tie down the parameters in the above equations.

Flight simulators would use the spatial model as follows. The frozen turbulence data base would be given to the particular simulation group on magnetic tape. The data on tape would be transferred to high speed, random access mass storage, e.g., disc storage. The size of the data base provided would depend upon the available mass storage for the particular computer. For reasons given in the next section, it is desirable to generate the data base in one large block. The total block may be too large for many simulators, and it is not necessary to use the entire data base. The main block can be divided into subblocks which will fit into available mass storage. If additional mass storage became available at a later time, then contiguous blocks of data could be shipped to create a larger data base at the installation.

Finally, Figure 28 shows how drastically the JAWS data truncates assumed spectra. This last figure in this chapter serves to reemphasize some points made herein. First, for realistic wind modeling for flight simulation, turbulence must be added to the JAWS data. Secondly, the model described by Equation (38) is a model based on necessity. It is not a scientifically accurate turbulence model, but it is a realistic engineering approach to a difficult problem. It uses all available information from JAWS and requires only turbulence length scale distributions for completeness. While the frozen turbulence is Gaussian, the winds generated by the model of Equation (38) are not, as real atmospheric winds are not. The model contains all three-dimensional correlations and spectra as any realistic model should. Finally, the model was specifically tailored for use with the JAWS data base, but is not restricted to it. If σ_i and $\bar{U}_i$ were known for winds on Jupiter, it is anticipated that flight vehicle entry into the Jovian atmosphere could be simulated with the proposed model.

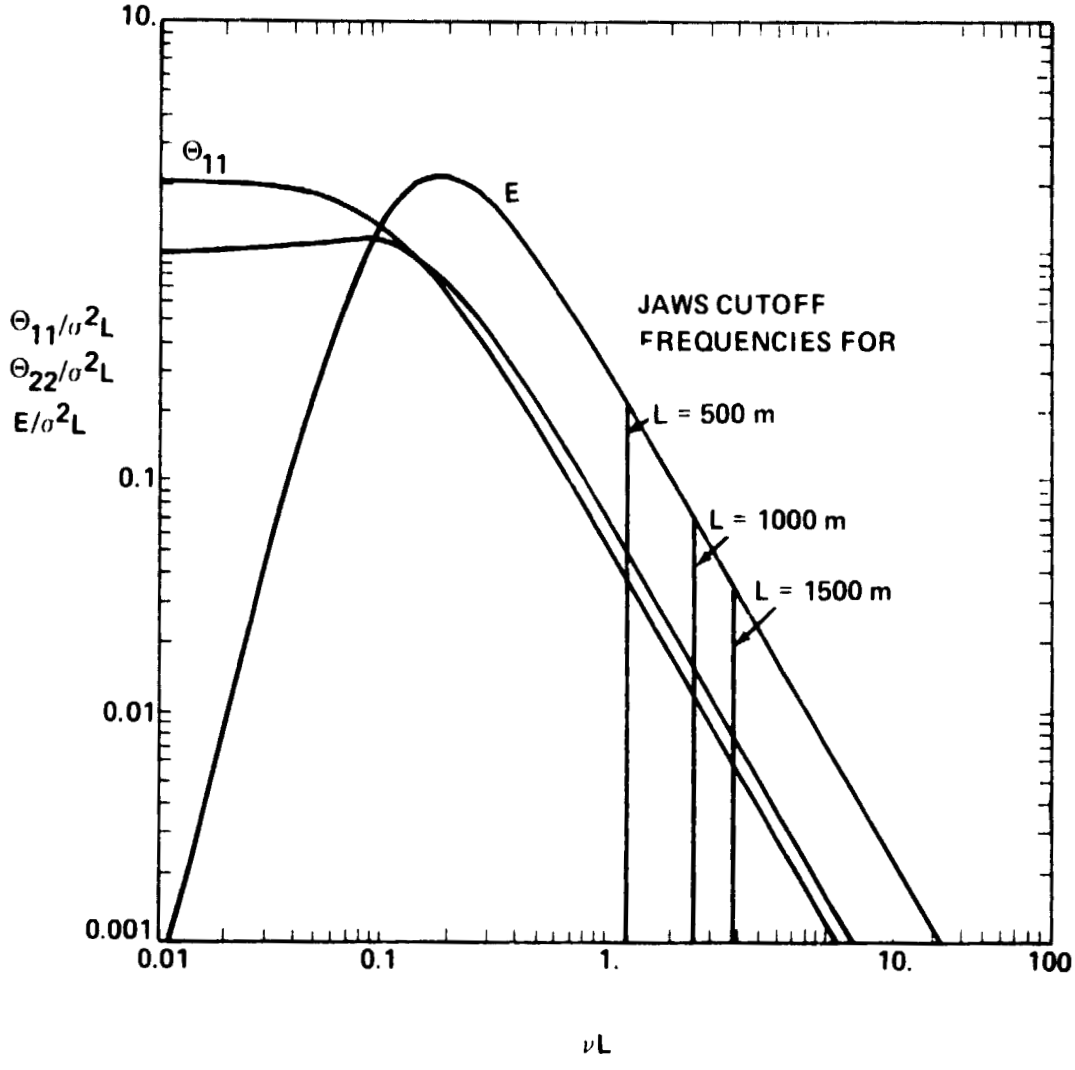


Figure 28. Cutoff frequencies for the JAWS data sets (von Karman spectra).

CHAPTER VI. FROZEN TURBULENCE GENERATION

A. The Fast Fourier Transform Approach

Three-dimensional turbulence generation was achieved using Fast Fourier Transform (FFT) techniques. Figure 29 is similar to Figure 20 but differs in that the filter function, input, and output are functions of three independent variables (four if time is included). If the FFT is used, the Dryden model has no computational advantage over the von Karman model. Hence, the von Karman model was used. Generation of turbulence using the FFT requires some awareness of certain FFT properties. These properties are discussed with some derivations given in the appendices. In this chapter all frequencies, spectra, and filter functions are assumed dimensionless.

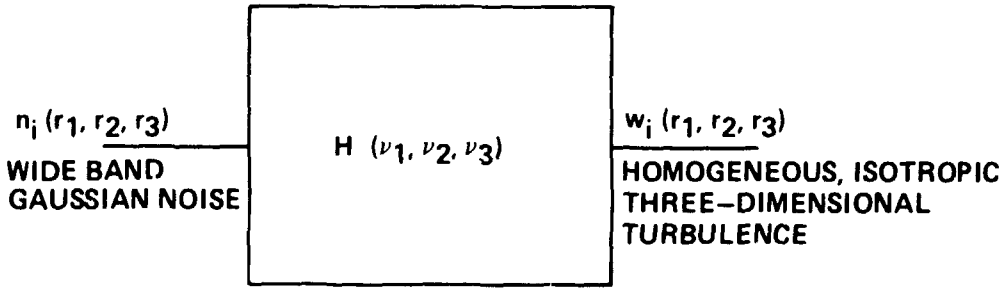


Figure 29. Generation of three-dimensional turbulence.

Since the turbulence is generated digitally, sampling frequencies in the space and spatial frequency domain play an important role. The relationships are given by the following equations

$$\Delta\nu_i = \frac{\nu_{s_i}}{M_i} \qquad i = 1, 2, 3 \tag{44}$$

$$\Delta r_i = \frac{1}{\nu_{s_i}} \qquad i = 1, 2, 3 \tag{45}$$

where ν_{s_i} is the sampling frequency corresponding to the r_i direction, M_i is the number of grid points in the r_i direction, $\Delta\nu_i$ is the grid spacing in the spatial frequency domain corresponding to the r_i direction, and Δr_i is the r_i grid spacing in the space domain.

The FFT is a fast computational implementation of the Discrete Fourier Transform (DFT). One-, two-, and three-dimensional DFTs and their inverses are defined in the following equations.

$$X_k = \sum_{n=0}^{M-1} x_n \, e^{-j2\pi nk/M} \qquad k = 0, 1, \ldots, M-1 \tag{46}$$

$$x_n = \frac{1}{M} \sum_{k=0}^{M-1} X_k \, e^{j2\pi nk/M} \qquad n = 0, 1, \ldots, M-1 \tag{47}$$

$$X_{k_1,k_2} = \sum_{n_1=0}^{M_1-1} \sum_{n_2=0}^{M_2-1} x_{n_1,n_2} \, e^{-j2\pi(n_1k_1/N_1 + n_2k_2/N_2)} \qquad k_i = 0, 1, \ldots, M_i-1 \tag{48}$$

$$x_{n_1,n_2} = \frac{1}{M_1M_2} \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} X_{k_1,k_2} \, e^{+j2\pi(n_1k_1/M_1 + n_2k_2/M_2)} \qquad n_i = 0, 1, \ldots, M_i-1 \tag{49}$$

$$X_{k_1,k_2,k_3} = \sum_{n_1=0}^{M_1-1} \sum_{n_2=0}^{M_2-1} \sum_{n_3=0}^{M_3-1} x_{n_1,n_2,n_3} \, e^{-j2\pi(n_1k_1/M_1 + n_2k_2/M_2 + n_3k_3/M_3)} \qquad k_i = 0, 1, \ldots, M_i-1 \tag{50}$$

$$x_{n_1,n_2,n_3} = \frac{1}{M_1M_2M_3} \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} \sum_{k_3=0}^{M_3-1} X_{k_1,k_2,k_3} \, e^{+j2\pi(n_1k_1/M_1 + n_2k_2/M_2 + n_3k_3/M_3)} \qquad n_i = 0, 1, \ldots, M_i-1 \tag{51}$$

In Equations (46) through (51) n and lower case x refer to the space domain and k and upper case X to the frequency domain. In the above six equations, even numbered equations, i.e., (46), (48), and (50) are the forward transforms and the odd numbered ones are the inverse transforms. The three different transform pairs are represented because in the discussion to follow symmetries associated with the one- and two-dimensional transforms show up in the three-dimensional DFT. These symmetries are of considerable importance to the generation of three-dimensional turbulence.

The three-dimensional continuous Fourier transform and its inverse are given by the following equations. Since all DFT implementations in this report are by means of the FFT, these two terms will be used interchangably

$$X(\nu_1,\nu_2,\nu_3) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(r_1,r_2,r_3)\, e^{-j2\pi(r_1\nu_1 + r_2\nu_2 + r_3\nu_3)}\, dr_1\, dr_2\, dr_3 \qquad (52)$$

$$x(r_1,r_2,r_3) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} X(\nu_1,\nu_2,\nu_3)\, e^{j2\pi(\nu_1 r_1 + \nu_2 r_2 + \nu_3 r_3)}\, d\nu_1\, d\nu_2\, d\nu_3 \qquad (53)$$

By comparison of (52) and (53) with (50) and (51), approximations for the Fourier transform and inverse are given by

$$X(k_1\Delta\nu_1,\, k_2\Delta\nu_2,\, k_3\Delta\nu_3) \approx \Delta r_1\, \Delta r_2\, \Delta r_3\, \text{DFT}\,[x(n_1\Delta r_1,\, n_2\Delta r_2,\, n_3\Delta r_3)] \qquad (54)$$

$$x(n_1\Delta r_1,\, n_2\Delta r_2,\, n_3\Delta r_3) \approx M_1\Delta\nu_1\, M_2\Delta\nu_2\, M_3\Delta\nu_3\, \text{DFT}^{-1}\,[X(k_1\Delta\nu_1,\, k_2\Delta\nu_2,\, k_3\Delta\nu_3)] \qquad (55)$$

Similar expressions apply for the one- and two-dimensional transforms.

Figure 29 seems to imply that two three-dimensional transforms are necessary for turbulence generation. First, noise generated in the space domain is transformed to the frequency domain, multiplied by a filter and then the inverse transformed back to the space domain. Actually, only one transform is required because the noise can be generated in

the frequency domain since transformed noise statistical properties can be calculated. Important to this discussion are the symmetry properties of transformed real noise for the one, two-, and three-dimensional cases. Derivations follow easily from the transform definitions. These properties are presented in Table 2.

Table 2. Symmetry Properties of Transformed One-, Two-, and Three-dimensional Digital Functions

	Expression	Observation
	One-Dimensional Symmetry	
1.	$X(M-k) = X^*(k)$	$Im[X(M/2)] = 0$
	Two-Dimensional Symmetry	
2.	$X(M_1-k_1,0) = X^*(k_1,0)$	$I_m[X(M_1/2,0) = 0]$
3.	$X(0,M_2-k_2) = X^*(0,k_2)$	$I_m[X(0,M_2/2) = 0]$
4.	$X(N_1-k_1,N_2-k_2) = X^*(k_1,k_2)$	$I_m[X(M_1/2, M_2/2) = 0]$
	Three-Dimensional Symmetry	
5.	$X(M_1-k_1,0,0) = X^*(k_1,0,0)$	$I_m[X(M_1/2,0,0)] = 0$
6.	$X(0, M_2-k_2, 0) = X^*(0,k_2,0)$	$I_m[X(0,M_2/2,0)] = 0$
7.	$X(0,0,M_3-k_3) = X^*(0,0,k_3)$	$I_m[X(0,0,M_3/2)] = 0$
8.	$X(M_1-k_1,M_2-k_2,0) = X^*(k_1,k_2,0)$	$I_m[X(M_1/2, M_2/2, 0)] = 0$
9.	$X(M_1-k_1, 0, M_3-k_3) = X^*(k_1,0,k_3)$	$I_m[X(M_1/2,0,M_3/2)] = 0$
10.	$X(0, M_2-k_2, M_3-k_3) = X^*(0,k_2,k_3)$	$I_m[X(0, M_2/2, M_3/2)] = 0$
11.	$X(M_1-k_1, M_2-k_2, M_3-k_3) = X^*(k_1,k_2,k_3)$	$I_m[X(M_1/2, M_2/2, M_3/2)] = 0$

The symmetry relations expressed in Table 2 seem quite complex, but all can be incorporated into a simple geometric interpretation. The geometric interpretation is reflection about the center point. In the one-dimensional case the reflection is about the center point ($k = M/2$) of the line, in the two-dimensional case, points along the two axes are reflected about their respective center points ($k_1 = M_1/2$, $k_2 = 0$; $k_1 = 0$, $k_2 = M_2/2$) and in the grid interior about the center point of the interior ($k_1, k_2 \neq 0$, reflection about $k_1 = M_1/2$, $k_2 = M_2/2$). In the three-dimensional case one-, two-, and three-dimensional reflections must be made about the three axes, the three planes (k_1-k_2, k_1-k_3, k_2-k_3), and about the interior symmetry point (k_1, k_2, k_3) = ($M_1/2, M_2/2, M_3/2$) in the interior of frequency space. The Hermitian symmetries are depicted for the one- and two-dimensonal cases by Figures 30 and 31. Showing the symmetries for the three-dimensional case is very difficult but the principle is the same. Figure 32 indicates the grid points at which the transform of a real function must take real values. If these points fall on an axis, Hermitian symmetry exists about the point on the axis. If the real point falls on the k_1-k_2, k_1-k_3, or k_2-k_3 plane, Hermitian symmetry about the point exists in the plane, and similarly for the interior point. Each point other than these eight real points possesses Hermitian symmetry with another point in the grid through the real point and extended a distance equal to the distance from the original point to

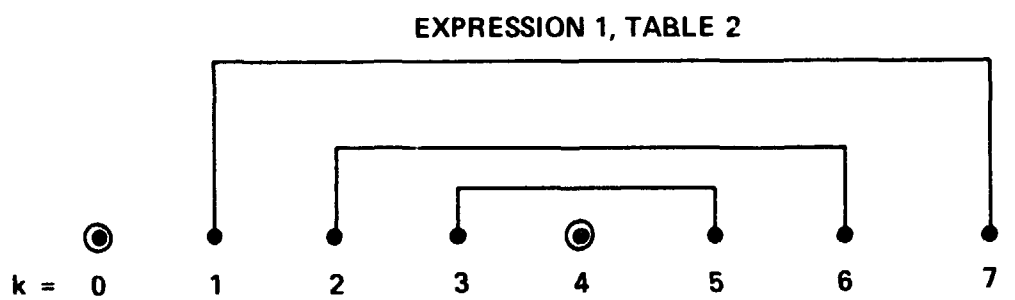


Figure 30. Correspondence of complex conjugate pairs in the one-dimensional transform domain.

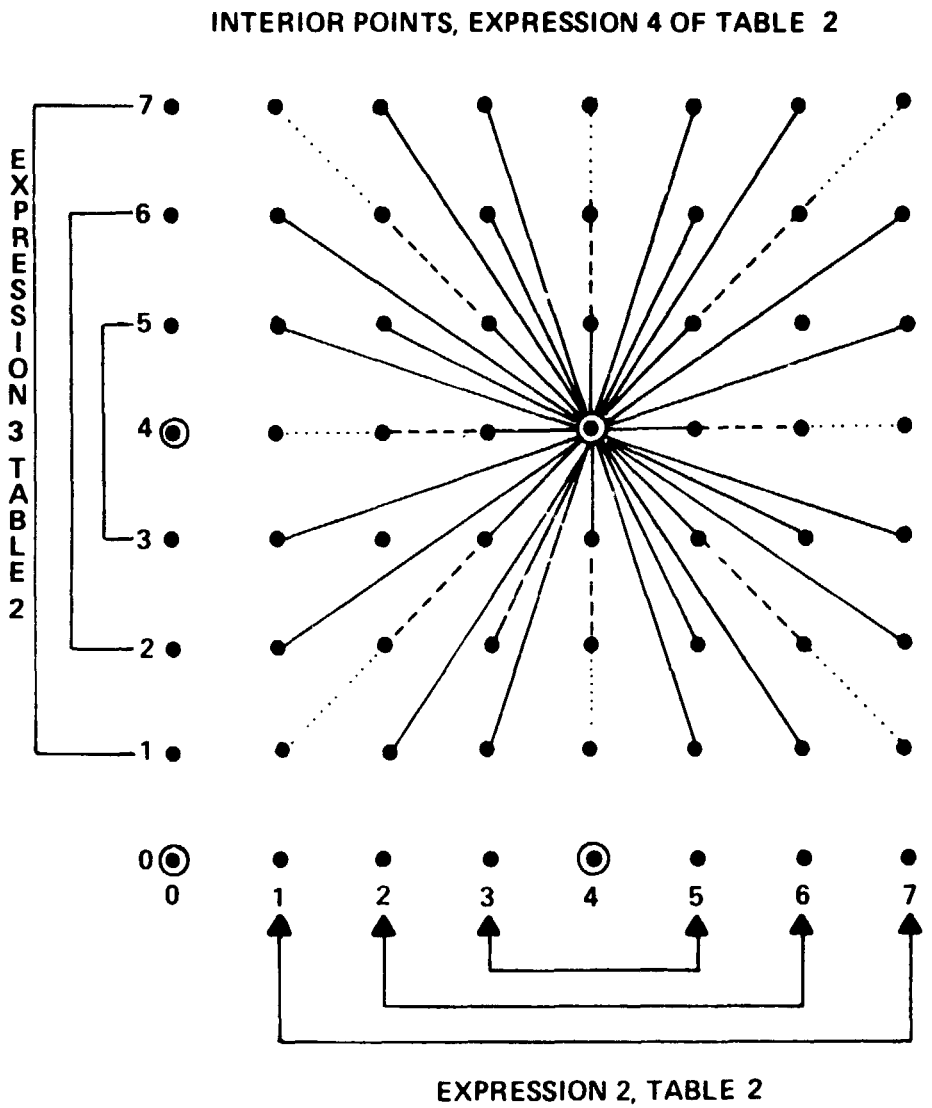


Figure 31. Correspondence of complex conjugate pairs in the two-dimensional transform domain.

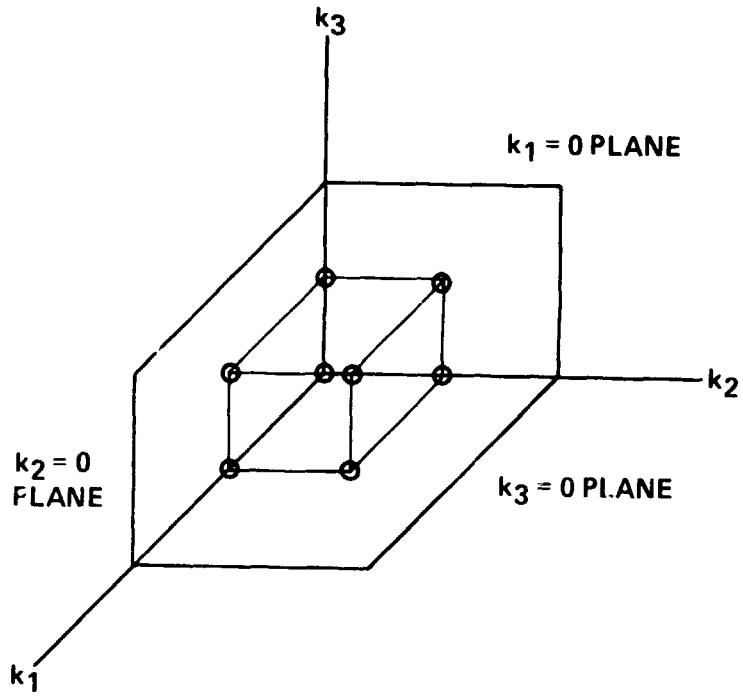


Figure 32. Points in the transform domain where the transforms must be real.

the real point. The end of the line segment falls at the corresponding grid point. Values of the transform at the original point and the new point will be complex conjugates of each other. This fact is a consequence of the requirement for real turbulence, i.e., not complex. In one dimension, there are two real points (k = 0, M/2), in two dimensions, four real points, three dimensions, eight real points, and so on.

Linearity is a well known property of the DFT and DFT^{-1}. Since the wide band random noise to be transformed is Gaussian, then the transformed noise is al. Gaussian. This being the case, if the expected value and variance of the transformed noi.. were known, then all the information required for generating noise in the frequency domain would be known. The mean and variance of transformed noise for the one- and three-dimensional cases are calculated in Appendix D and summarized in Table 3. With the information available in Table 3, generation of three-dimensional transformed noise in the frequency domain is possible with a simple Gaussian random number generator.

Once generated, the noise must be rearranged according to the Hermitian symmetries of Table 2 so that when transformed back to the space domain the resulting turbulence takes real rather than complex values.

Table 3. Expected Value and Variance of Transformed One-, and three-dimensional noise with zero mean value

One-Dimensional Case

$$X_k = \sum_{n=0}^{M-1} x_n e^{-j2\pi nk/N} \Delta r$$

1 $E[X_k] = 0$

2 $\sigma_{X_k}^2 = E[X_k X_k^*] = M(\Delta r)^2 \sigma_{x_n}^2$

3 $E[X_l X_k^*] = M(\Delta r)^2 \sigma_{x_n}^2 \delta_{kl}$

4 $E[Re^2(X_k)] = M(\Delta r)^2 \sigma_{x_n}^2 [1/2 + 1/2\, \delta_{0k} + 1/2\, \delta_{M/2\,k}]$

5 $E[I_m^2(X_k)] = \dfrac{M(\Delta r)^2 \sigma_{x_n}^2}{2} [1 - \delta_{0k} - \delta_{M/2\,k}]$

6 $E[Re[X_k] \cdot I_m[X_l]] = 0$ for all l and k

Three-Dimensional Case

$$X_{k_1,k_2,k_3} = \sum_{n_1=0}^{M_1-1} \sum_{n_2=0}^{M_2-1} \sum_{n_3=0}^{M_3-1} x_{n_1,n_2,n_3} e^{-j2\pi(n_1k_1/M_1 + n_2k_2/M_2 + n_3k_3/M_3)} \Delta r_1 \Delta r_2 \Delta r_3$$

7 $E[X_{k_1,k_2,k_3}] = 0$

8 $\sigma_{X_k}^2 = E[X_{k_1,k_2,k_3} X^*_{k_1,k_2,k_3}] = M_1 M_2 M_3 \sigma_{x_n}^2 (\Delta r_1 \Delta r_2 \Delta r_3)^2$

$= M_1 M_2 M_3 \sigma_{x_n}^2 / (\nu_{s_1} \nu_{s_2} \nu_{s_3})^2$

9 $E[X_{k_1,k_2,k_3} X^*_{l_1,l_2,l_3}] = [M_1 M_2 M_3 \sigma_{x_n}^2 / (\nu_{s_1} \nu_{s_2} \nu_{s_3})] \delta_{k_1 l_1} \delta_{k_2 l_2} \delta_{k_3 l_3}$

10 $E[Re^2(X_{k_1,k_2,k_3})] = \dfrac{M_1 M_2 M_3 \sigma_{x_n}^2}{2(\nu_{s_1} \nu_{s_2} \nu_{s_3})^2} \left[1 + \left(\delta_{0k_1} + \delta_{\frac{M_1}{2}\,k_1}\right)\left(\delta_{0k_2} + \delta_{\frac{M_2}{2}\,k_2}\right)\left(\delta_{0k_3} + \delta_{\frac{M_3}{2}\,k_3}\right)\right]$

11 $E[I_m^2(X_{k_1,k_2,k_3})] = \dfrac{M_1 M_2 M_3 \sigma_{x_n}^2}{2(\nu_{s_1} \nu_{s_2} \nu_{s_3})^2} \left[1 - \left(\delta_{0k_1} + \delta_{\frac{M_1}{2}\,k_1}\right)\left(\delta_{0k_2} + \delta_{\frac{M_2}{2}\,k_2}\right)\left(\delta_{0k_3} + \delta_{\frac{M_3}{2}\,k_3}\right)\right]$

12 $E[Re[X_{k_1,k_2,k_3}] \cdot I_m[X_{l_1,l_2,l_3}]] = 0$ for all $k_1, k_2, k_3, l_1, l_2, l_3$

B. Derivation of Filter Functions

A simple approach for deriving the one-dimensional filter function is presented and then generalized to the three-dimensional case. Since the turbulence is generated digitally, the frequency cutoff, i.e., the Nyquist frequency, is the sampling frequency divided by two ($\nu_s/2$). The DFT is an approximation to the continuous Fourier transform evaluated over these limits (see Figure 33). The magnitude of the constant spectrum was selected as σ_n^2/ν_s so that the following equation was satisfied.

$$\int_{-\nu_s/2}^{\nu_s/2} \Phi_n \, d\nu_s = \sigma_n^2 \tag{56}$$

The expression for the output spectrum for a linear filter in terms of the input spectrum is

$$\Phi_w(\nu) = H(\nu)\, H^*(\nu)\, \Phi_n(\nu) = |H(\nu)|^2\, \Phi_n(\nu) \tag{57}$$

The output spectrum is the desired spectrum and is known while the input spectrum is as shown in Figure 33. The solution for $H(\nu)$ is

$$H(\nu) = \frac{\sqrt{\nu_s \Phi_w}}{\sigma_n} e^{j\alpha(\nu)} \tag{58}$$

where $\alpha(\nu)$ is any arbitrary phase function. For this study, $\alpha(\nu) = 0$ was selected.

The one-dimensional case can be generalized to any number of dimensions. For the three-dimensional case the analog to Equation (56) is

$$\int_{-\nu_{s1}/2}^{\nu_{s1}/2} \int_{-\nu_{s2}/2}^{\nu_{s2}/2} \int_{-\nu_{s3}/2}^{\nu_{s3}/2} \Phi_n \, d\nu_1 \, d\nu_2 \, d\nu_3 = \sigma_n^2 \quad . \tag{59}$$

This equation implies a value of Φ_n of $\sigma_n^2/(\nu_{s1}\ \nu_{s2}\ \nu_{s3})$. Equation (57) still applies with each term being a function of these variables. The new filter function is given by

$$H(\nu_1,\nu_2,\nu_3) = \frac{\sqrt{\nu_{s1}\ \nu_{s2}\ \nu_{s3}\ \Phi_w(\nu_1,\nu_2,\nu_3)}}{\sigma_n} \tag{60}$$

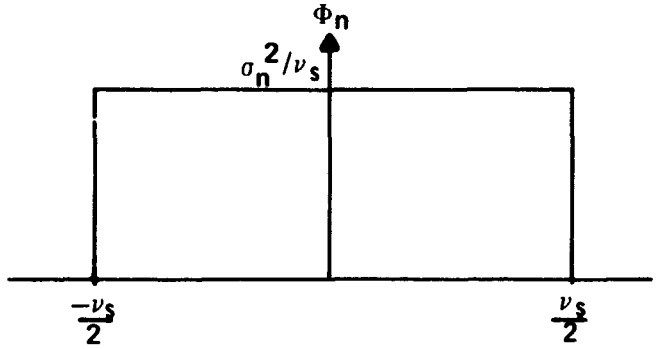


Figure 33. Input noise spectrum.

This equation is the general form of the filter function. Φ_w for this study is the three-dimensional von Karman spectral function for either the u, v, or w velocity component. In Chapter II the dimensional forms of the von Karman spectra were presented. In Chapter V the need for nondimensional spectra was described. The dimensional spectra are transformed by $\nu_i = \hat{\nu}_i L$ where $\hat{\nu}_{si}$ is the dimensional wave number. Then the dimensionless spectrum Φ_w is given in terms of the dimensional spectrum $\hat{\Phi}_w$ by the following relation:

$$\Phi_w(\nu_1,\nu_2,\nu_3)\ d\nu_1\ d\nu_2\ d\nu_3 = \hat{\Phi}_w\left(\frac{\nu_1}{L}, \frac{\nu_2}{L}, \frac{\nu_3}{L}\right)\ \frac{d\nu_1\ d\nu_2\ d\nu_3}{L^3} \tag{61}$$

With this transformation, the nondimensional spectra given in Equation (13) become

$$\frac{\Phi_{ii}}{\sigma^2} = \frac{440\pi^3}{9}\ \frac{a^4\ (\nu^2 - \nu_i^2)}{[1 + (2\pi a\nu)^2]^{17/6}} \tag{62}$$

With Equations (60) and (62) and the results of Tables 2 and 3 all the tools are available for generation of noise in the frequency domain. The sequence is as follows. Generate three-dimensional noise over half of three-dimensional space then fill the rest of the space using the Hermitian symmetry properties of Table 2. Make sure the noise variances satisfy the restrictions of Table 3. Multiply the transformed noise point by point by a sampled version of the dimensionless filter function derived from Equations (60) and (62) above. Apply the inverse FFT and the result in the space domain is three-dimensional dimensionless turbulence which can be added to mean wind data sets. Codes for performing each of these steps are described and listed in the appendices.

The generated turbulence contains all the correlations with lags in each of the three space directions. This being the case, if a line of turbulence, say in the r_1 direction is selected and the one-dimensional spectrum calculated with an FFT the result should be an approximation to the one-dimensional continuous spectrum. To check this, consider the generated turbulence given by

$$w(n_1,n_2,n_3) = \Delta\nu_1\ \Delta\nu_2\ \Delta\nu_3 \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} \sum_{k_3=0}^{M_3-1} H(k_1,k_2,k_3)\ N(k_1,k_2,k_3) \exp\left[+j2\pi\left(\frac{n_1k_1}{M_1} + \frac{n_2k_2}{M_2} + \frac{n_3k_3}{M_3}\right)\right] \quad (63)$$

Then $w(n_1,n_2,n_3)$ is transformed with the following equation.

$$\hat{w}(l_1,n_2,n_3) = \Delta r_1 \sum_{n_1=0}^{N_1-1} w(n_1,n_2,n_3)\ e^{-j2\pi n_1 l_1/M_1} \quad (64)$$

If Equation (63) is substituted into (64), and algebra performed as in Appendix E, the final result is

$$\widehat{\Theta}_w(l_1) = \frac{1}{\Delta\nu_1} \sum_{k_2=0}^{M_2-1} \sum_{k_3=0}^{M_3-1} \Phi_n(l_1,k_2,k_3)\, \Delta\nu_2\, \Delta\nu_3 \tag{65}$$

This is the digital analog to Equation (17). The factor $1/\Delta\nu_1$ must be present because Θ_w is an energy density, i.e., the turbulence energy between frequencies ν_1 and $\nu_1 + d\nu_1$ is given by $\Theta_w(\nu_1)\, d\nu_1$. Equation (65) is consistent with previous derivations.

A description of the means for turbulence generation was presented in this chapter. The technique is a three step procedure, namely (1) generation of three-dimensional noise, (2) multiplication by the sampled spectral function, and (3) inverse transformation by FFT to the space domain. An attempt was made to help the reader understand the important aspects of the problem by presenting geometrical interpretations of the procedure. A clear understanding of these interpretations helps in understanding the codes used for the turbulence generation. Two FORTRAN programs were written, one for the first two steps above, and another for the final step. These programs are described in Appendices F and G, respectively.

CHAPTER VII. USE OF WINDSHEAR DATA SETS WITH SIMULATED TURBULENCE

A description of the addition of turbulence to a JAWS data set is presented. The JAWS July 14 case was selected because some aircraft data from the same day were available for estimating gust intensities and length scales of turbulence. While these aircraft data were not from the Doppler radar measurement region, the results should be applicable in a general way to the aircraft measurements. This conclusion was based on the fact that July 14 was a day when many microbursts occurred over a large area. Any gross variation in turbulence characteristics from one microburst group to another on this day seems unlikely.

The JAWS data set characteristics were described in Chapter II and Table 1 indicates that grid spacing on the July 14 case was 200 m by 200 m by 150 m. For simulated flights using this data, interpolation of the JAWS winds was required. The interpolation procedure was based on bilinear Lagrange polynomial basis functions. A good description of two-dimensional interpolation is presented by Prenter, 1975 [34]. The two-dimensional method is easily generalized to three or n dimensions. For three dimensions, the general interpolation form is

$$\bar{U}(x,y,z) = \sum_{n_1=0}^{M_1-1} \sum_{n_2=0}^{M_2-1} \sum_{n_3=0}^{M_3-1} \bar{U}_{n_1,n_2,n_3} P_{n_1,n_2,n_3}(x,y,z) \quad , \tag{66}$$

where $\bar{U}(x,y,z)$ is the interpolated quantity, $\bar{U}_{n_1,n_2,n_3}$ is the value of the quantity at the grid points and $P_{n_1,n_2,n_3}(x,y,z)$ is a group of basis functions, one for each grid

point. The simplest set of basis functions for our need is the set of bilinear Lagrange polynomials. Each of these basis functions is continuous so any finite sum of the functions is also continuous. Therefore, continuity of the interpolated quantity $\bar{U}(x,y,z)$ is assured.

The Lagrange basis functions P_{n_1,n_2,n_3} are defined at each grid point so that $P_{n_1,n_2,n_3}(x_{n_1},y_{n_2},z_{n_3}) = 1$. x_{n_1}, y_{n_2}, and z_{n_3} are x, y, and z coordinates of grid point (n_1,n_2,n_3). At all surrounding grid points $P_{n_1,n_2,n_3}(x_{n_1},y_{n_2},z_{n_3}) = 0$. In practice the summation of Equation (66) is only over the surrounding eight grid points. In equation form the basis functions associated with a particular cell are defined by

$$P_1(x_c,y_c,z_c) = (\Delta x-x_c)\,(\Delta y-y_c)\,(\Delta z-z_c)/(\Delta x\,\Delta y\,\Delta z) \tag{67}$$

$$P_2(x_c,y_c,z_c) = x_c(\Delta y-y_c)\,(\Delta z-z_c)/(\Delta x\,\Delta y\,\Delta z) \tag{68}$$

$$P_3(x_c,y_c,z_c) = x_c\,y_c\,(\Delta z-z_c)/(\Delta x\,\Delta y\,\Delta z) \tag{69}$$

$$P_4(x_c,y_c,z_c) = (\Delta x-x_c)\,y_c\,(\Delta z-z_c)/(\Delta x\,\Delta y\,\Delta z) \tag{70}$$

$$P_5(x_c,y_c,z_c) = (\Delta x-x_c)\,(\Delta y-y_c)\,z_c/(\Delta x\,\Delta y\,\Delta z) \tag{71}$$

$$P_6(x_c,y_c,z_c) = x_c\,(\Delta y-y_c)\,z_c/(\Delta x\,\Delta y\,\Delta z) \tag{72}$$

$$P_7(x_c,y_c,z_c) = x_c\,y_c\,z_c/(\Delta x\,\Delta y\,\Delta z) \tag{73}$$

$$P_8(x_c,y_c,z_c) = (\Delta x-x_c)\,y_c\,z_c/(\Delta x\,\Delta y\,\Delta z)\quad , \tag{74}$$

where x_c,y_c,z_c are the coordinates within the cell as defined in Figure 34; $\Delta x,\Delta y,\Delta z$ are the x, y, and z grid spacing, respectively; and P_i is the basis function corresponding to grid point i as defined in Figure 34.

Notice that gridpoint one lies at $(x_c,y_c,z_c) = (0,0,0)$, grid point two at $(x_c,y_c,z_c) = (\Delta x,0,0)$, etc. Notice also that P_i is equal to 1 at grid point i, is always nonnegative, and

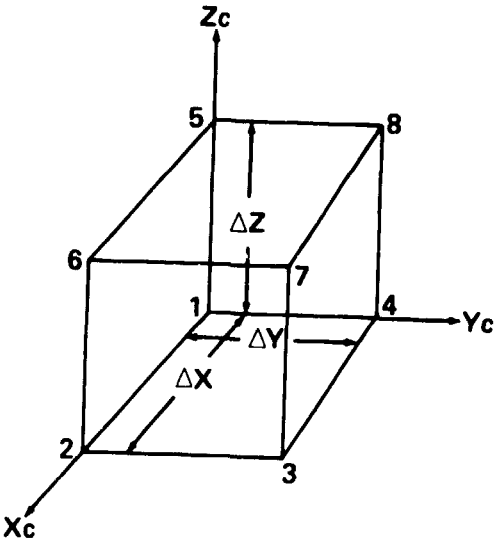


Figure 34. Definition of interpolation variables.

is zero at surrounding grid points. Outside of the cell shown in Figure 34, each of the eight basis functions have different definitions, but these alternate definitions are of no concern to the interpolation of $\bar{U}$ within the cell.

Interpolation within the grid of Figure 34 is achieved with the following equation.

$$\bar{U}(x,y,z) = \sum_{i=1}^{8} \bar{U}_i \, P_i(x,y,z) \quad , \tag{75}$$

where $\bar{U}_i$ is the value of $\bar{U}$ as the ith grid point.

The material in this chapter was implemented for the most part by the computer program listed in Appendix G. This program was written on a minicomputer with limited central memory. As a result, data was swapped from central memory to disc files and back. Three-dimensional arrays could not be conveniently stored on disc. The arrays had to be stored in linear records. For this reason, the JAWS data was stored cell by cell on disc. In other words, the eight values of $\bar{U}$ at each of the eight grid points of Figure 34 were stored contiguously. An adjacent cell would have four of the

same values stored so that memory-wise, this procedure is inefficient. Computation-wise it is efficient because accessing the grid values of the cell requires only one READ statement as opposed to four required by another procedure. More of these details are described in Appendix G.

The addition of turbulence to the JAWS data required information on the variation of gust intensity and turbulent length scale. At the time of writing, JAWS second moment information was not available. An equation for the distribution of gust intensity was necessary for simulation. As a guide, consider Figure 27. This figure shows velocities measured by NASA's B-57B aircraft during flight through an apparent microburst on the day after the July 14 JAWS case. An interpretation of Figure 27 was presented in Chapter V. Evidence of decreased gust intensity and increased turbulent length scale were described. For the purposes of demonstration a combination of both was assumed.

An exponential type decrease through the JAWS microburst was assumed. Lateral variation was written in terms of distance r from the microburst. Moving away from the microburst center, gust intensity increased to a maximum and then decreased to zero exponentially at great distances. A vertical exponential factor was added to make intensity a maximum at 300 m AGL. The functional form is given by

$$\sigma = [(\sigma_{max} - (\sigma_{max}/2)\exp^{-[(r/0.75)^2]}) \exp(-r^2/50) \exp[(z-300/200)]. \quad (76)$$

where r is horizontal distance from the microburst center in km and z is the altitude in meters.

A simple vertical relation for turbulent length scale was selected from Reference 35. The functional form is

$$L = 31.5\,(z/18.3)^{0.64} \text{ meters} \quad (77)$$

Turbulence generated for use with the JAWS data consisted of a 64 x 64 x 64 array generated at dimensionless frequencies of $\nu_{s_1} = 50$, $\nu_{s_2} = 50$, $\nu_{s_3} = 50$. One concern

was the accumulation of error in the calculation of an FFT this large. The three-dimensional FFT does not have an error problem, at least for arrays of this size. Test cases were done and error was on the order of 10^{-6} which is the same as the word precision (32 bits per word, 24 bits for the mantissa).

Interpolation of the turbulence between grid points was required. Tatom and Smith, 1982 [36] offer some guidance on this point. They show that zeroth order interpolation (stair step turbulence) causes aliasing to exactly cancel the effect of digitization of the data as long as the turbulence is sampled at frequencies the same as or lower than generation frequencies. This procedure was followed.

To illustrate model performances a point "airplane" was flown at constant ground speed through the data. The program in Appendix G that implements this model has a pair of "knobs", one of which turns the turbulence up or down, and the other turns the wind shear on or off. Hence, the JAWS data without turbulence, the turbulence generated without JAWS data, or JAWS data with turbulence can be examined.

The point "airplane" path was selected to begin at a point so that a true heading of 35° carried the plane through the microburst center. A simulated ILS approach with three degree glide slope was flown beginning at an altitude of 350 meters.

Figure 35 shows the east-west velocity component encountered in flight without added turbulence. Figure 36 is the same flight path with turbulence added. Notice that the length scales are decreasing toward the end of the flight as the plane approaches the ground. The turbulence in this figure appears quite realistic (compare with Figure 27).

The traces of turbulence only are plotted in Figure 37. The data was sampled at a very high rate so that each data point was shown. Input parameters were arranged for horizontal flight through the block in the east-west direction. The middle curve is a plot of the points ix = 1-64, iy = 1, iz = 1, the upper curve ix = 1-64, iy = 2, iz = 1, and the lower curve, ix = 1-64, iy = 32, iz = 32. The first two adjacent curves are highly

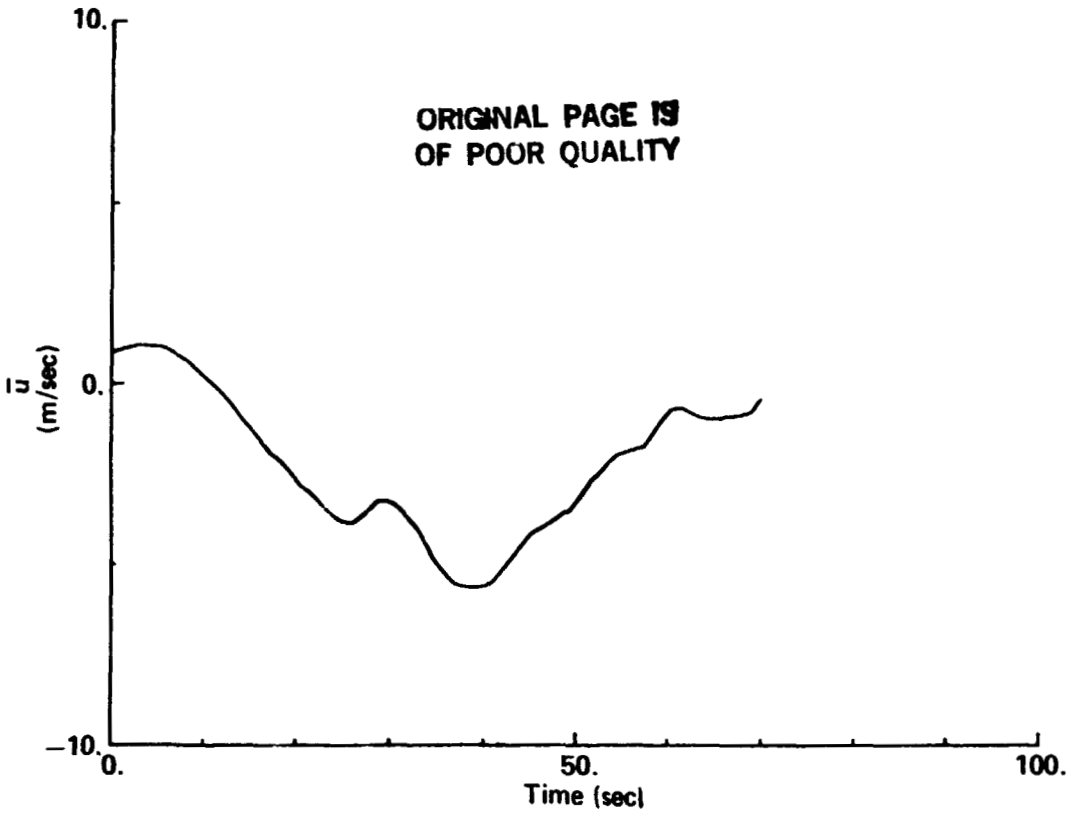


Figure 35. East-west JAWS velocity on a simulated ILS approach (three degree glide slope).

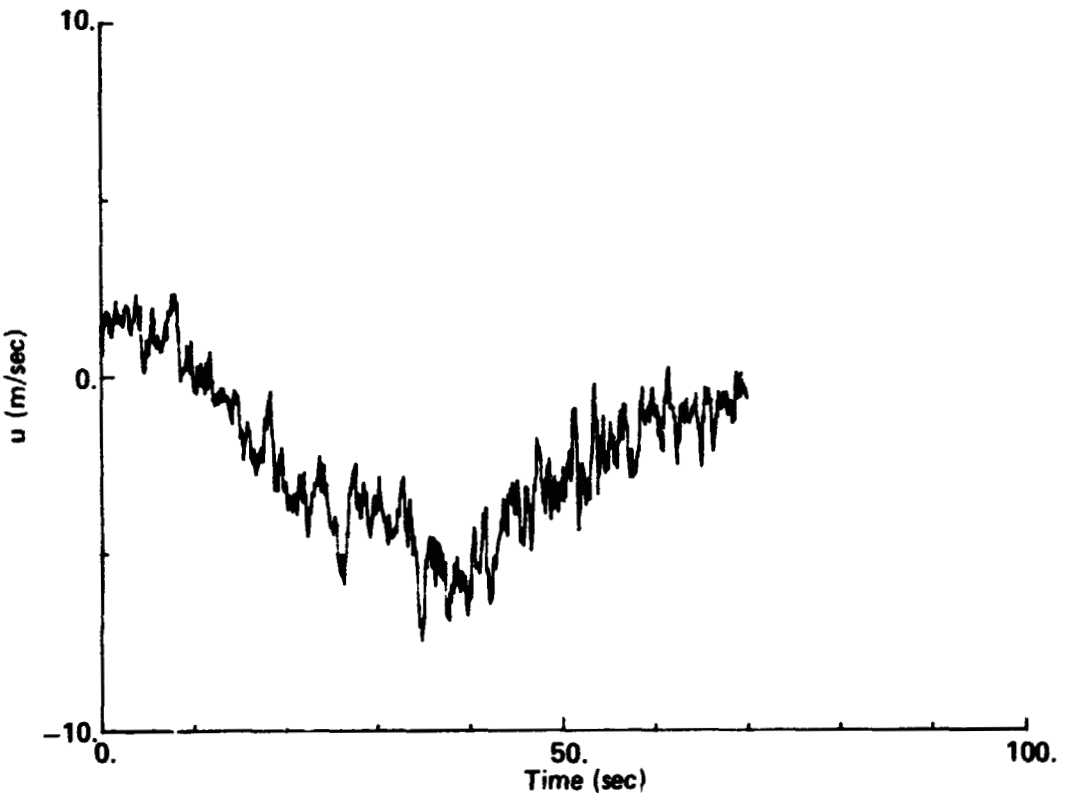


Figure 36. East-west JAWS velocity component plus turbulence on a simulated ILS approach (three degree glide slope).

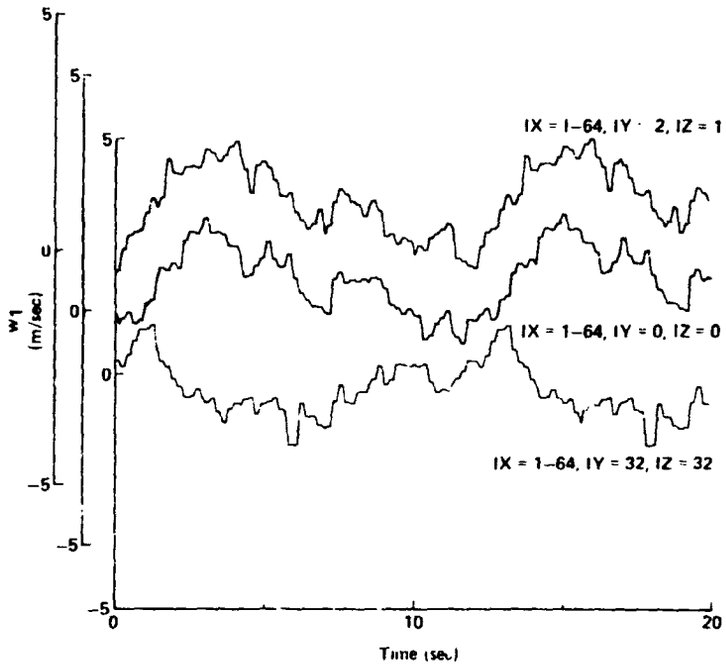


Figure 37. Comparison of lines of turbulence from the simulated turbulence.

correlated. The third curve shows only slight correlation with the first two, as was expected. These three traces show the kind of transverse (lateral and vertical) three-dimensional correlation which was missing from previous one-dimensional turbulence simulations.

The storage requirements for this realistic three-dimensional simulation are not as large as one might think. The July 14 JAWS case was provided on a 61 x 61 x 11 grid. For three velocity components and one second moment, the number of words is 163,724. The turbulence was stored on a 64 x 64 x 64 grid and with three components the number of words is 262,144. The JAWS and the turbulence data were stored in integer form in only two bytes. The total storage requirement is only slightly greater than 850 kilobytes. The Perkin-Elmer 3250 minicomputer can be outfitted with 32 megabytes of central core storage most of which is available to the user. The conclusion is that very sophisticated simulations can be performed on small computers.

CHAPTER VIII. SUMMARY AND CONCLUSION

The characteristics of the spatial model can be summarized as follows:

1. The spatial model contains three components of real wind shear varying over space with corresponding components of simulated turbulence also varying over the three space dimensions. Because wind and gust variation over the body of an aircraft are available, all aerodynamic loads and moments can be calculated.

2. The simulated turbulence is nonlinear, non-Gaussian, and conforms to the von Karman three-dimensional spectral model.

3. Because of the conformance to the three-dimensional spectra, the model contains cross-spectral information for each component.

4. By virtue of its three dimensionality, the spatial model permits flight simulations of any maneuver without diminishing the validity of the simulation.

5. The model is highly flexible. Any flow field about which ensemble average velocities, gust intensities, and turbulent length scales are known can be simulated.

The spatial model was implemented on a Hewlett-Packard F series minicomputer. The three-dimensional turbulence was generated in a 64 x 64 x 64 block and showed the three-dimensional correlation expected of it. The resulting turbulence when added to the JAWS data resulted in a simulated wind trace virtually indistinguishable to the eye from real data measured with the B-57B aircraft.

An extension of the Houbolt-Sen cross-spectral model was presented and compared with data from the Gust Gradient Program. The theoretical curves showed some variation from the measured data. This variation is believed to be a probe effect, aliasing, an effect of the nonfrozen nature of the turbulence, or a combination of the three factors.

The spatial model requires one statistic not available from the JAWS data; the length scale of turbulence, L. The effect of microbursts on distributions of L should be a subject for future research. This research should include flight and laboratory measurements. Flight measurements are required because of the inability to model atmospheric scale Reynolds number in the laboratory. Laboratory measurements are required because of the difficulty of locating and flying through a microburst. Although laboratory measurements cannot provide quantitative results, they should provide trends.

Laboratory measurements can contribute significantly to the understanding of the effect of topography on microbursts. The primary controlling parameter for microburst shape is the Froude number which can be modeled very well in the laboratory. Topographic effects in the JAWS data should be identified and carefully considered where the data are used with the spatial model.

REFERENCES

1. Fujita, T. T., "Manual of Downburst Identification for Project NIMROD," SMRP Res. Paper 156, University of Chicago, 104 pp.

2. Wilson, James, and Roberts, Rita, "Evaluation of Doppler Radar for Airport Windshear Detection," Paper presented at the AMS 21st Conference on Radar Meteorology, Edmonton, Alberta, Canada, September 19-23, 1983.

3. Fujita, T. T., "Microbursts as an Aviation Wind Shear Hazard," Presented at the AIAA 19th Aerospace Sciences Meeting, January 12-15, 1981, St. Louis, Missouri, AIAA-81-0386.

4. Campbell, Warren, Camp, Dennis W., and Frost, Walter, "An Analysis of Spanwise Gust Gradient Data," Paper presented at the Ninth Conference on Aerospace and Aeronautical Meteorology, June 6-9, 1983, Omaha, Nebraska.

5. Caracena, Fernando, Scientist from National Center for Atmospheric Research, Boulder, Colorado, Personal Communication, October, 1982.

6. Fujita, T. Theodore, and Wakimoto, Roger M., "Five Scales of Airflow Associated with a Series of Downbursts on 16 July, 1980," Monthly Weather Review, Volume 109, No. 7, July, 1981.

7. Elmore, Kim, "Description and Critique of an Outflow Windshear Model Developed by NASA Ames," As yet an unpublished report of the National Center for Atmospheric Research, September, 1983.

8. Zhu, Shang Xiang and Etkin, Bernard, "Fluid Dynamic Model of a Microburst," Institute for Aerospace Studies, University of Toronto, Report UTIAS 271, April, 1983.

9. Anonymous, "The JAWS Project Preliminary Data Description," NCAR internal document released September, 1983.

10. von Karman, T., and Howarth, L., "The Fundamentals of the Statistical Theory of Turbulence," Journal of the Aeronautical Sciences, Volume 4, 1937, pp. 131-138.

11. Etkin, B., Dynamics of Atmospheric Flight, John Wiley and Sons, Inc., New York, 1972.

12. Filon, L. N. G., "On a Quadrature Formula for Trigonometric Integrals," Proceedings of the Royal Society of Edinburgh, Volume 49, 1928, pp. 38-47.

13. Houbolt, J. C., Steiner, R., and Pratt, K. G., "Dynamic Response of Airplanes to Atmospheric Turbulence Including Flight Data on Input and Response," NASA TR-R-199, 1964.

14. Reeves, Paul M., "A Non-Gaussian Turbulence Simulation," Air Force Flight Dynamics Laboratory, Air Systems Command, Wright Patterson AFB, Ohio, AFFDL-TR-69-67, November, 1969.

15. Reeves, Paul M., Campbell, G. S., Ganzer, V. M., and Joppa, R. G., "Development and Application of a Non-Gaussian Atmospheric Turbulence Model for Use in Flight Simulators," NASA CR 2451, September 1974.

16. Reeves, Paul M., Joppa, Robert G., and Ganzer, Victor M., "A Non-Gaussian Model of Continuous Atmospheric Turbulence for Use in Aircraft Design," NASA CR 2639, January, 1976.

17. Fichtl, George H., and Perlmutter, Morris, "Nonstationary Atmospheric Boundary-Layer Turbulence Simulation," Journal of Aircraft, Volume 12, Number 8, August, 1975, pp. 639-647.

18. Fichtl, George H., "A Technique for Simulating Turbulence for Aerospace Vehicle Flight Simulation," NASA TM 78141, November, 1977.

19. Tatom, F. B., Smith, S. R., Fichtl, G. H., and Campbell, C. W., "Simulation of Atmospheric Turbulent Gusts and Gust Gradients," Journal of Aircraft, Volume 19, April, 1982, pp. 264-271.

20. Dutton, John A., Kerman, Bryan R., and Petersen, Erik L., "Contributions to the Simulation of Turbulence," NASA CR 2762, December, 1976.

21. Fichtl, G. H., Perlmutter, Morris, and Frost, Walter, "Monte Carlo Turbulence Simulation," from Handbook of Turbulence, Volume I, Edited by Walter Frost and Trevor H. Moulden, Plenum Publishing Corporation, 1977, pp. 433-473.

22. Wang, Show-Tien, and Frost, Walter, "Atmospheric Turbulence Simulation Techniques with Application to Flight Analysis," NASA CR 3309, September, 1980.

23. Abbot, Edwin A., Flatland, Dover Publications, Inc., New York, 1952.

24. Terry, Carl, Engineer, United Airlines, Denver, Colorado, Personal Communication, September, 1983.

25. Kristensen, L., Kirkegaard, P., and Lenschow, D. H., "Squashed Atmospheric Turbulence," Riso National Laboratory, Report No. Riso-R-478. January, 1983.

26. Mark, W. D., "Characterization of Stochastic Transients and Transmission Media: The Method of Power-Moments Spectra," Journal of Sound and Vibration, Volume 22, Number 3, 1972, pp. 249-295.

27. Mark, William D., and Fischer, Raymond W., "Investigation of the Effects of Nonhomogeneous (or nonstationary) Behavior on the Spectra of Atmospheric Turbulence." NASA Contractor Report CR-2745, Langley Research Center, October, 1976.

28. Mark, William D., "Characterization of Nongaussian Atmospheric Turbulence for Prediction of Aircraft Response Statistics," NASA Contractor Report 2913, Langley Research Center, December, 1977.

29. Mark, William D., "Characterization, Parameter Estimation and Aircraft Response Statistics of Atmospheric Turbulence," NASA Contractor Report 3463, Langley Research Center, September, 1981.

30. Mark, William D., and Fischer, Raymond W., "Statistics of Some Atmospheric Turbulence Records Relevant to Aircraft Response Calculations," NASA Contractor Report 3464, Langley Research Center, September, 1981.

31. Campbell, Warren, Camp, Dennis W., and Frost, Walter, "An Analysis of Spanwise Gust Gradient Data," Paper presented at the Ninth Conference on Aerospace and Aeronautical Meteorology, June 6-9, 1983, Omaha, Nebraska.

32. Launder, B. E., and Spalding, D. B., Lectures in Mathematical Models of Turbulence, Academic Press, New York, 1972.

33. Launder, B. E., and Spalding, D. B., "The Numerical Computation of Turbulent Flows," Computer Methods in Applied Mechanics and Engineering, Volume 3, 1974, pp. 269-289.

34. Prenter, P. M., Splines and Variational Methods, John Wiley and Sons, New York, 1975, pp. 127-129.

35. Kaufman, John W. (editor), "Terrestrial Environment (Climatic) Criteria Guidelines for Use in Aerospace Vehicle Development, 1977 Revision," NASA TM 78118, November, 1977.

36. Tatom, Frank B., and Smith, S. Ray, "Advanced Space Shuttle Simulation Model," NASA CR 3541, April, 1982.

APPENDIX A. MICROBURST RESEARCH IN THE LABORATORY

The dimensionless parameters governing microburst-type flows are the Reynolds and Froude numbers ($Re = VD/\nu$, $Fr = V/\sqrt{g(\beta\Delta T)D}$), where β is the volumetric expansion coefficient. For real-atmosphere microbursts typical velocity and length scales are 10 m/sec (downdraft) and 600 m (depth of out-flow). The atmospheric Reynolds number is then on the order of 4×10^8. A typical microburst ΔT is 3 deg C which implies $g\beta\Delta T \approx 10$ cm/sec^2 = 0.1 m/sec^2. The atmospheric Froude number is then on the order of 1.3. The Froude number exerts a strong influence on microburst shape while the Reynolds number affects turbulent details of the flow. In the laboratory, the Froude number can be modeled quite well, but the Reynolds number cannot. Let V_{LAB} = 100 cm/sec, $g\beta\Delta T$ = 100 cm/sec^2, ν = 0.01 cm^2/sec, and D = 0.5 cm. Then Fr = 14, and Re = 5000. Hence very high Froude numbers, or any smaller value can be achieved so that the shape of the microburst can be accurately simulated in the laboratory. Turbulence in the flow will not correspond quantitatively to that in the atmosphere but trends should be observable.

Intuitively, the higher the Froude number, the shallower the outflow depth. The shallower outflow depth corresponds to an increased outflow velocity. The increased outflow velocity and decreased depth creates a zone of intense shear which results in increased turbulence production. Hence the Froude number has a secondary influence on turbulence. The Reynolds number affects turbulence and therefore turbulent mixing and thus the momentum diffusivity. Higher values of momentum diffusivity result in smoother velocity profiles (except near surfaces where viscous effects predominate) and thus the Reynolds number has a secondary effect on microburst shape.

Figure 38 depicts a laboratory apparatus for studying microbursts. Saline solutions with densities 19% greater than fresh water can be achieved at 20°C. A reasonable

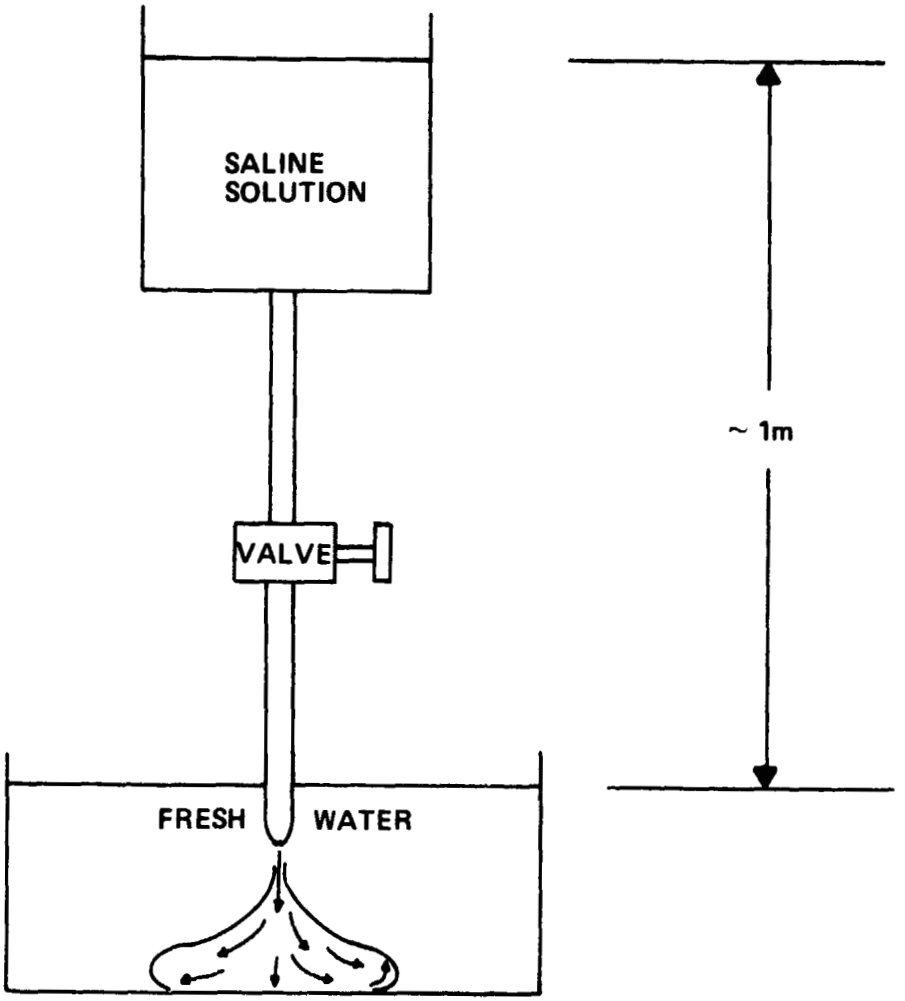


Figure 38. Laboratory "microburst" apparatus.

working value for $\beta\Delta T$ would be 10%. For normal laboratory work, one meter head seems reasonable. This gives a maximum velocity of $V = \sqrt{2gh} = 4$ m/sec. If the injector is constructed with a contraction, higher values of velocity can be achieved. The valve controls the strength of the momentum source. The apparatus of Figure 38 can be of any size and complexity depending on resources. The valve could be microprocessor controlled with flowmeter feedback, for example. Velocity measurements could be made with hot wire or split film, laser Doppler velocimeter, or by particle photography methods. Flow visualization could be achieved by means by Schlieren, shadowgraph, interferometer, particle photography, hydrogen bubbles, or dye.

Many of the questions posed previously could be studied with the apparatus of Figure 4. The effect of topography on microburst spreading should be modeled quite well with the apparatus since Froude number scaling can be achieved. Some idea of the effect of surface roughness can be accomplished also. Surface roughness affects the surface layer immediately but eventually could affect higher layers. Consider for the moment Figure 39. The laboratory microburst center is at the center of the circle. The microburst front after a time is retarded by the increased roughness in the rough sector. The retardation creates shear layers in the transition area between the rough and smooth surfaces. If the effect is marked enough, Kelvin-Helmholz instabilities could be created. Variations of vorticity are observed in blowing dust associated with microbursts.

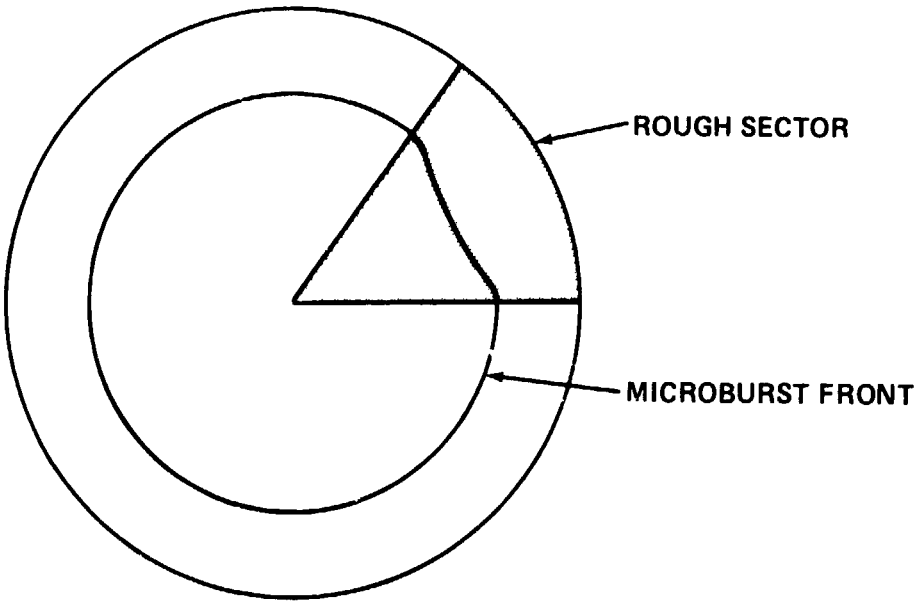


Figure 39. Effects of surface roughness on microburst spreading.

By increasing head and hence velocities, variation of Reynold's number can be achieved in the laboratory model. The effect on turbulence characteristics could be observed for changing Reynolds number. The turbulence variations, i.e., length scales and intensities have crucial relevance to the wind simulation model. While differences

will be observed between the laboratory turbulence and atmospheric turbulence, the trends should be there. The importance of the trends or functional forms was explained in Chapter V.

APPENDIX B. THE THREE-DIMENSIONAL CHARACTER OF THE VON KARMAN SPECTRA

As an example, consider the three-dimensional von Karman spectrum for the vertical velocity component.

$$\Phi_{33} = \frac{C_1 (\nu_1^2 + \nu_2^2)}{[1 + C_2 (\nu_1^2 + \nu_2^2 + \nu_3^2)^{17/6}} \tag{78}$$

where

$$C_1 = \frac{440\pi^3}{9} \sigma^2 a^4 L^5$$

and

$$C_2 = (2\pi aL)^2 \quad .$$

Defining $\rho^2 = \nu_1^2 + \nu_2^2$, substituting into (78) and solving for ν_3 gives the following result.

$$\nu_3 = \pm \sqrt{\left[\left(\frac{C_1 \rho^2}{\Phi_{33}}\right)^{6/17} - 1\right] \frac{1}{C_2} - \rho^2} \quad . \tag{79}$$

Equation (79) is the equation for surfaces of constant ϕ_{33}.

If Equation (79) is differentiated with respect to ρ, set to zero, and solved for ρ the result is

$$\rho = \left(\frac{6}{17C_2}\right)^{17/22} \left(\frac{C_1}{\Phi_{33}}\right)^{3/17} \quad . \tag{80}$$

For this value of ρ for a given value of Φ_{33}, ν_3 achieves its maximum value.

The maximum value of Φ_{33} can also be determined for given C_1 and C_2. By inspection of (78), the maximum value of Φ_{33} occurs at $\nu_3 = 0$, since f_3 is a monitonically decreasing function of Φ_{33}. The problem of finuing a maximum for Φ_{33} is reduced to finding the maximum value of Φ_{33} defined by

$$\Phi_{33} = \frac{C_1 \rho^2}{[1 + C_2 \rho^2]^{17/6}} \tag{81}$$

Differentiating (81) with respect to ρ equating the result to zero, and solving for ρ gives

$$\rho = \sqrt{\frac{6}{11C_2}} \quad . \tag{82}$$

Substituting this into the expression for Φ_{33} gives

$$\hat{\Phi}_{33} = \frac{6C_1/11C_2}{[1 + 6/11]^{17/6}} \quad . \tag{83}$$

Using σ = 1 m/sec, L = 500 m in Equations (82) and (83) gives

$$\rho = 1.76 \times 10^{-4}\ m^{-1} \quad \text{and} \quad \hat{\Phi}_{33} = 1.37 \times 10^{9}\ m^5/sec^2 \quad .$$

The maximum value of Φ_{33} occurs at a small but nonzero value of ρ and has a large but not infinite value.

APPENDIX C. SUM OF COMPLEX EXPONENTIAL SERIES

In this study the following sum occurs frequently in derivations.

$$S = \sum_{n=0}^{N-1} (e^{i\theta})^n \quad . \tag{84}$$

The above summation is recognized immediately as a partial sum of the geometric series. Equation (84) can l e rewritten in the following form.

$$S = \sum_{n=0}^{\infty} (e^{i\theta})^n - \sum_{n=N}^{\infty} (e^{i\theta})^n = \sum_{n=0}^{\infty} (e^{i\theta})^n - e^{i\theta N} \sum_{n=0}^{\infty} (e^{i\theta})^n$$

$$S = \sum_{n=0}^{\infty} (e^{i\theta})^n \, [1 - e^{i\theta N}] \tag{85}$$

Equation (85) is true if the infinite series converges. From complex variable theory it is known that the geometric series below converges if $|z| < 1$.

$$G = \sum_{n=0}^{\infty} z^n = \frac{1}{1-z} \tag{86}$$

In actual fact (86) converges everywhere in the closed region $|z| \leq 1$ except at $z = \pm 1$. Hence

$$S = \frac{1 - e^{i\theta N}}{1 - e^{i\theta}} \qquad \theta \neq 0, \pi \tag{87}$$

By direct substitution into (84) S can be calculated at $\theta = 0$, and π.

$$S = N \qquad \theta = 0$$
$$S = 0 \qquad \theta = 0,\ N \text{ even} \tag{88}$$

APPENDIX D. DERIVATION OF PROPERTIES OF TRANSFORMED NOISE

The purpose of this appendix is the derivation of the expressions summarized in Table 3. In these derivations, x_n and x_{n_1,n_2,n_3} are assumed to be real white noise such that $E[x_n] = 0$, $E[x_n^2] = \sigma_{x_n}^2$, $E[x_n x_m] = \sigma_{x_n}^2 \delta_{mn}$, $E[x_{n_1,n_2,n_3}] = 0$, $E[x_{n_1}^2,_{n_2,n_3}] = \sigma_{x_n}^2$, and $E[x_{n_1,n_2,n_3} x_{m_1,m_2,m_3}] = \sigma_{x_n}^2 \delta_{n_1 m_1} \delta_{m_2 n_2} \delta_{m_3 n_3}$. The operator $E[\bullet]$ means the expected value of the variable in the parentheses and δ_{mn} is the Kronecker delta, i.e., $\delta_{mn} = 0$ for $m \neq n$, and $\delta_{mn} = 1$ if $m = n$.

Expression 1.

$$E[X_k] = 0$$

$$X_k = \sum_{n=0}^{M-1} x_n \, e^{-j2\pi nk/M} \, \Delta r$$

$$E[X_k] = E\left[\sum_{n=0}^{M-1} x_n \, e^{-j2\pi nk/M}\right] = \sum_{n=0}^{M-1} E[x_n] \, e^{-j2\pi nk/M} = 0$$

Expression 2.

$$E[X_k X_k^*] = \sum_{n_1=0}^{M-1} \sum_{n_2=0}^{M-1} E[x_{n_1} x_{n_2}] \, e^{-j2\pi(n_1-n_2)k/M} \, (\Delta r)^2$$

$$= \sum_{n_1=0}^{M-1} \sum_{n_2=0}^{M-1} \sigma_{x_n}^2 \, \delta_{n_1 n_2} \, e^{-j2\pi(n_1-n_2)k/M} \, (\Delta r)^2$$

$$= \sigma_{x_n}^2 \, M \, (\Delta r)^2$$

Expression 3.

$$E[X_k X_l^*] = E\left[\sum_{n_1=0}^{M-1} x_{n_1} e^{-j2\pi n_1 k/M} \sum_{n_2=0}^{M-1} x_{n_2} e^{+j2\pi n_2 l/M} (\Delta r)^2\right]$$

$$= (\Delta r)^2 \sum_{n_1=0}^{M-1} \sum_{n_2=0}^{M-1} E[x_{n_1} x_{n_2}] e^{-j2\pi(n_1 k - n_2 l)/M}$$

$$= (\Delta r)^2 \sum_{n_1=0}^{M-1} \sum_{n_2=0}^{M-1} \sigma_{x_n}^2 \delta_{n_1 n_2} e^{-j2\pi(n_1 k - n_2 l)/M}$$

$$= (\sigma_{x_n} \Delta r)^2 \sum_{n=0}^{M-1} e^{-j2\pi(k-l)n/M} = (\sigma_{x_n} \Delta r)^2 \delta_{kl}$$

Expression 4.

$$Re^2 [X_k] = \sum_{n_1=0}^{M-1} x_{n_1} \cos(2\pi n_1 k/M) \sum_{n_2=0}^{M-1} x_{n_2} \cos(2\pi n_2 k/M)(\Delta r)^2$$

$$E[Re^2 [X_k]] = \sum_{n_1=0}^{M-1} \sum_{n_2=0}^{M-1} E[x_{n_1} x_{n_2}] \cos(2\pi n_1 k/M) \cos(2\pi n_2 k/M) (\Delta r)^2$$

$$= \sum_{n=0}^{M-1} \cos^2(2\pi nk/M) \sigma_{x_n}^2 (\Delta r)^2$$

$$= \sigma_{x_n}^2 (\Delta r)^2 \sum_{n=0}^{M-1} \left(\frac{1}{2} + \cos\frac{4\pi nk}{M}\right)$$

$$= \frac{\sigma_{x_n}^2 (\Delta r)^2}{2} \left(M + Re\left[\sum_{n=0}^{M-1} e^{j4\pi nk/M}\right]\right)$$

$$= \frac{M\sigma_{x_n}^2 (\Delta r)^2}{2} \left(1 + \delta_{0k} + \delta\, M/2\, k\right)$$

Expression 5.

$$E[Im^2\,[X_k]] = \sigma_{x_k}{}^2\,(\Delta r)^2 \sum_{n=0}^{M-1} \sin^2\,(2\pi nk/M)$$

$$= \sigma_{x_k}{}^2\,(\Delta r)^2 \sum_{n=0}^{M-1} \left(\frac{1}{2} - \frac{\cos\,(4\pi nk/M)}{2}\right)$$

$$= \frac{M\sigma_{x_n}{}^2\,(\Delta r)^2}{2} \left(1 - \delta_{0k} - \delta\,M/2\,k\right)$$

Expression 6.

$$E[Re[X_k]Im[X_l]] = \sum_{n_1=0}^{M-1} \sum_{n_2=0}^{M-1} E[x_{n_1} x_{n_2}] \cos\,(2\pi n_1 k/M) \sin\,(2\pi n_2 l/M)\,(\Delta r)^2$$

$$= \sum_{n_1=0}^{M-1} \sum_{n_2=0}^{M-1} \delta_{n_1 n_2}\,\sigma_{x_n}{}^2 \cos\,(2\pi n_1 k/M) \sin\,(2\pi n_2 l/M)\,(\Delta r)^2$$

$$= \sigma_{x_n}{}^2\,(\Delta r)^2 \sum_{n=0}^{M-1} \cos\,(2\pi nk/M) \sin\,(2\pi nl/M) = 0$$

By the orthogonality property of the sine and cosine functions.

Expression 7.

$$E[X_{k_1,k_2,k_3}] = \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} \sum_{k_3=0}^{M_3-1} E[x_{n_1,n_2,n_3}]\, e^{-j2\pi\left(\frac{n_1k_1}{M_1} + \frac{n_2k_2}{M_2} + \frac{n_3k_3}{M_3}\right)} \times \Delta r_1\,\Delta r_2\,\Delta r_3$$

$$= 0$$

Expression 8:

$$E[X_{k_1,k_2,k_3}\, X^*_{k'_1,k'_2,k'_3}] = (\Delta r_1\, \Delta r_2\, \Delta r_3)^2 \sum_{n_1=0}^{M_1-1} \sum_{n_2=0}^{M_2-1} \sum_{n_3=0}^{M_3-1} E[x_{n_1,n_2,n_3}\, x_{\hat{n}_1,\hat{n}_2,\hat{n}_3}] \exp\left\{-j2\pi\left[\frac{(n_1-\hat{n}_1)k_1}{M_1} + \frac{(n_2-\hat{n}_2)k_2}{M_2} + \frac{(n_3-\hat{n}_3)k_3}{M_3}\right]\right\}$$

$$= \sigma_{x_n}^2\, (\Delta r_1\, \Delta r_2\, \Delta r_3)^2 \sum_{n_1=0}^{M_1-1} \sum_{n_2=0}^{M_2-1} \sum_{n_3=0}^{M_3-1} \delta_{n_1\hat{n}_1}\, \delta_{n_2\hat{n}_2}\, \delta_{n_3\hat{n}_3} \exp\left\{-j2\pi\left[\frac{(n_1-\hat{n}_1)k_1}{M_1} + \frac{(n_2-\hat{n}_2)k_2}{M_2} + \frac{(n_3-\hat{n}_3)k_3}{M_3}\right]\right\}$$

$$= \sigma_{x_n}^2\, (\Delta r_1\, \Delta r_2\, \Delta r_3)^2\, M_1\, M_2\, M_3$$

Expression 9:

$$E[X_{k_1,k_2,k_3}\, X^*_{l_1,l_2,l_3}] = \sigma_{x_n}^2\, (\Delta r_1\, \Delta r_2\, \Delta r_3)^2 \sum_{n_1=0}^{M_1-1} \sum_{n_2=0}^{M_2-1} \sum_{n_3=0}^{M_3-1} \exp\left[-j2\pi\, \frac{n_1(k_1-l_1)}{M_1} + \frac{n_2(k_2-l_2)}{M_2} + \frac{n_3(k_3-l_3)}{M_3}\right]$$

$$= (\Delta r_1\, \Delta r_2\, \Delta r_3)^2\, \sigma_{x_n}^2\, \delta_{k_1l_1}\, \delta_{k_2l_2}\, \delta_{k_3l_3}\, M_1\, M_2\, M_3$$

$$= [M_1\, M_2\, M_3\, \sigma_{x_n}^2/(\nu_{s_1}\, \nu_{s_2}\, \nu_{s_3})^2]\, \delta_{k_1l_1}\, \delta_{k_2l_2}\, \delta_{k_3l_3}$$

Expression 10:

$$E[Re^2(X_{k_1,k_2,k_3})] = (\Delta r_1\,\Delta r_2\,\Delta r_3\,\sigma_{x_n})^2 \sum_{n_1=0}^{M_1-1}\sum_{n_2=0}^{M_2-1}\sum_{n_3=0}^{M_3-1} \cos^2\left(2\pi n_1 k_1/M_1 + \frac{2\pi k_2 n_2}{M_2} + \frac{2\pi k_3 n_3}{M_2}\right.$$

$$= (\Delta r_1\,\Delta r_2\,\Delta r_3\,\sigma_{x_n})^2 \sum_{n_1=0}^{M_1-1}\sum_{n_2=0}^{M_2-1}\sum_{n_3=0}^{M_3-1} \left\{\frac{1}{2} - \frac{1}{2}\cos\left[4\pi\left(\frac{n_1k_1}{M_1} + \frac{n_2k_2}{M_2} + \frac{n_3k_3}{M_3}\right)\right]\right\}$$

$$= (\Delta r_1\,\Delta r_2\,\Delta r_3\,\sigma_{x_n})^2 \left[\frac{M_1\,M_2\,M_3}{2} + \frac{1}{2}\,R_e \sum_{n_1=0}^{M_1-1} \exp\left(+j4\pi\left(\frac{n_1k_1}{M_1}\right)\right) \sum_{n_2=0}^{M_2-1} \exp\left(+j4\pi\left(\frac{n_2k_2}{M_2}\right)\right) \sum_{n_3=0}^{M_3-1} \exp\left(+j4\pi\left(\frac{n_3k_3}{M_3}\right)\right)\right]$$

$$= (\Delta r_1\,\Delta r_2\,\Delta r_3\,\sigma_{x_n})^2 \frac{M_1\,M_2\,M_3}{2}\left[1 + \left(\delta_{0k_1} + \delta_{M_1/2\,k_1}\right)\left(\delta_{0k_2} + \delta_{\frac{M_2}{2}k_2}\right)\left(\delta_{0k_3} + \delta_{M_3/2\,k_3}\right)\right]$$

Expression 11:

$$E[I_m^2(X_{k_1k_2k_3})] = (\Delta r_1\,\Delta r_2\,\Delta r_3\,\sigma_{x_n})^2 \sum_{n_1=0}^{M_1-1}\sum_{n_2=0}^{M_2-1}\sum_{n_3=0}^{M_3-1} \sin^2\left[2\pi\left(\frac{n_1k_1}{M_1} + \frac{n_2k_2}{M_2} + \frac{n_3k_3}{M_3}\right)\right]$$

$$= (\Delta r_1\,\Delta r_2\,\Delta r_3\,\sigma_{x_n})^2 \sum_{n_1=0}^{M_1-1}\sum_{n_2=0}^{M_2-1}\sum_{n_3=0}^{M_3-1} \left\{\frac{1}{2} - \frac{1}{2}\cos\left[4\pi\left(\frac{n_1k_1}{M_1} + \frac{n_2k_2}{M_2} + \frac{n_3k_3}{M_3}\right)\right]\right\}$$

$$= \frac{(\Delta r_1\,\Delta r_2\,\Delta r_3\,\sigma x_n)^2\,M_1\,M_2\,M_3}{2}\left[1 - \left(\delta_{0k_1} + \delta_{\frac{M_1}{2}k_2}\right)\left(\delta_{0k_2} + \delta_{\frac{M_2}{2}k_2}\right)\left(\delta_{0k_3} + \delta_{\frac{M_3}{2}k_3}\right)\right]$$

Expression 12:

$$E[Re(X_{k_1k_2k_3})\,Im(X_{l_1l_2l_3})] = (\sigma_{x_n}\,\Delta r_1\,\Delta r_2\,\Delta r_3)^2 \sum_{n_1=0}^{M_1-1}\sum_{n_2=0}^{M_2-1}\sum_{n_3=0}^{M_3-1} \sin\left[2\pi\left(\frac{n_1k_1}{M_1}+\frac{n_2k_2}{M_2}+\frac{n_3k_3}{M_3}\right)\right] \times \cos\left[2\pi\left(\frac{n_1l_1}{M_1}+\frac{n_2l_2}{M_2}+\frac{n_3l_3}{M_3}\right)\right]$$

$$= \frac{(\sigma_{x_n}\,\Delta r_1\,\Delta r_2\,\Delta r_3)^2}{4j} \sum_{n_1=0}^{M_1-1}\sum_{n_2=0}^{M_2-1}\sum_{n_3=0}^{M_3-1} \exp\left[+j2\pi\left(\frac{n_1(k_1-l_1)}{M_1}+\frac{n_2(k_2-l_2)}{M_2}+\frac{n_3(k_3-l_3)}{M_3}\right)\right]$$

$$+ \exp\left[+j2\pi\left(\frac{n_1(k_1+l_1)}{M_1}+\frac{n_2(k_2+l_2)}{M_2}+\frac{n_3(k_3+l_3)}{M_3}\right)\right] - \exp\left[+j2\pi\left(\frac{n_1(l_1-k_1)}{M_1}+\frac{n_2(l_2-k_2)}{M_2}+\frac{n_3(l_3-k_3)}{M_3}\right)\right]$$

$$- \exp\left[-j2\pi\left(\frac{n_1(k_1+l_1)}{M_1}+\frac{n_2(k_2+l_2)}{M_2}+\frac{n_3(k_3+l_3)}{M_3}\right)\right]$$

$$= \frac{(\sigma_{x_n}\,\Delta r_1\,\Delta r_2\,\Delta r_3)^2}{4j}\Big[\delta_{k_1l_1}\,\delta_{k_2l_2}\,\delta_{k_3l_3} + \delta_{0k_1}\,\delta_{0l_1}\,\delta_{0k_2}\,\delta_{0l_2}\,\delta_{0k_3}\,\delta_{0l_3} - \delta_{k_1l_1}\,\delta_{k_2l_2}\,\delta_{k_3l_3} - \delta_{0k_1}\,\delta_{0l_1}\,\delta_{0k_2}\,\delta_{0l_2}\,\delta_{0k_3}\,\delta_{0l_3}\Big]$$

$$= 0$$

APPENDIX E. ESTIMATE OF THE ONE-DIMENSIONAL SPECTRUM FROM THREE-DIMENSIONAL SIMULATED TURBULENCE

This Appendix derives the result presented in Equation (65). The notation $\Delta^3\nu = \Delta\nu_1\ \Delta\nu_2\ \Delta\nu_3$ is used and the convention $\sum_{ki}$ is accepted to mean $\sum_{k_i=0}^{M_1-1}$

$$w(n_1,n_2,n_3) = \Delta^3\nu \sum_{k_1}\sum_{k_2}\sum_{k_3} H(k_1,k_2,k_3)\, N(k_1,k_2,k_3)\, e^{j2\pi\left[\frac{n_1k_1}{M_1}+\frac{n_2k_2}{M_2}+\frac{n_3k_3}{M_3}\right]} \tag{89}$$

In the above expression $N(k_1,k_2,k_3)$ is the transformed noise and $H(k_1,k_2,k_3)$ is the filter function corresponding to the desired output spectrum.

Define W as follows.

$$\widehat{W}(l_1,n_2,n_3) = \sum_{n_1} w(n_1,n_2,n_3)\, e^{-j2\pi(l_1n_1/N_1)}\, \Delta r_1 \tag{90}$$

Substituting Equation (89) into Equation (90) and rearranging gives

$$\widehat{W} = \Delta^3\nu\ \Delta r_1 \sum_{k_1}\sum_{k_2}\sum_{k_3} H(k_1,k_2,k_3)\, N(k_1,k_2,k_3)\, e^{j2\pi\left[\frac{n_2k_2}{M_2}+\frac{n_3k_3}{M_3}\right]} \sum_{n_1} e^{j2\pi\left[\frac{n_1(k_1-l_1)}{N_1}\right]}$$

$$\widehat{W} = \Delta^3\nu\ \Delta r_1 \sum_{k_2}\sum_{k_3} H(l_1,k_2,k_3)\, N(l_1,k_2,k_3)\, e^{j2\pi\left[\frac{n_2k_2}{M_2}+\frac{n_3k_3}{M_3}\right]}$$

With this result the variance of $\widehat{W}$ becomes

$$E[|\widehat{W}(l_1,n_2,n_3)|^2] = (\Delta^3\nu\ \Delta r_1\ \sigma_k)^2 \sum_{k_2}\sum_{k_3} |H(l_1,k_2,k_3)|^2 \tag{91}$$

where σ_k is the variance of the transformed noise and is given in Table 3 as

$$\sigma_k^2 = \frac{M_1 M_2 M_3 \sigma_n^2}{(\nu_{s_1} \nu_{s_2} \nu_{s_3})^2} \tag{92}$$

The filter function is given in Equation (60) and is repeated here for convenience in digital form.

$$H(l_1,k_2,k_3) = \frac{\sqrt{\nu_{s_1} \nu_{s_2} \nu_{s_3} \Phi_w(l_1,k_2,k_3)}}{\sigma_n} \tag{93}$$

Here $H(l_1,k_2,k_3)$ and $\Phi_w(l_1,k_2,k_3)$ are taken to mean $H(l_1\Delta\nu_1,k_2\Delta\nu_2,k_3\Delta\nu_3)$ and $\Phi_w(l_1\Delta\nu_1,k_2\Delta\nu_2,k_3\Delta\nu_3)$ respectively.

Using Equations (92) and (93) in Equation (91) along with relations between sampling frequencies and grid spacing yields the following desired result.

$$E[|\widehat{W}(l_1,n_2,n_3)|^2] = \widehat{\Theta}_w(l_1) = \frac{1}{\Delta\nu_1} \sum_{k_2} \sum_{k_3} \Phi_W(l_1,k_2,k_3)\, \Delta\nu_2\, \Delta\nu_3 \tag{94}$$

APPENDIX F. FROZEN TURBULENCE GENERATION IN THE FREQUENCY DOMAIN

FTURB generates a three-dimensional block of turbulent velocities for either the u(II=1), v(II=2), or w(II=3) components. Dimensionless sampling frequencies are selected for each coordinate direction. ν_{s_1} = FS1, ν_{s_2} = FS2, and ν_{s_3} = FS3 in the program. A typical value for these parameters is 50. The first executable statement in FTURB is a call to a library routine LGBUF. The HP computer system this program was run on has a limited I/O buffer and a call to LGBUF creates a buffer LBUF 128 - 16 bit words long so that records of this length can be read by the system. For other systems without buffer problems the call to LGBUF can be removed along with the dimensional array LBUF(128).

The file of transformed turbulence is created in a direct access disc file in records 128 - 16 bit words. The array being created is complex, and each complex number is made up of four words, so each block contains 32 complex numbers. Storage can be visualized as a one-dimensional array with index IX changing fastest, IY, next fastest, and IZ slowest. Figure 40 depicts storage for a small block. In the block, each cell corresponds to a storage location (logically) for one complex number. The 128 words are stored physically in 4 - 32 complex word records.

Generation of the turbulence in the frequency domain requires access to lines of the block of logical turbulence. For example, in the figure, say we want to get a line of turbulence in the Z direction, say IX = 3, IY = 2, and IZ = 1 to 4. These indices correspond to words 11, 43, 75, and 107. Routine FCHRP handles this chore.

Within the main program, the turbulence is generated according to the symmetry relations of Table 2. Variances of the transformed turbulence real and imaginary parts come from expressions 10 and 11 in Table 3. The (F1,0,0) loop corresponds to

symmetry relation 5 in Table 3. The (0,F2,0) loop corresponds to relation 6, the (0,0,F3) loop to 7, the (F1,F2,0) loop to 8, the (F1,0,F3) to 9, the (0,F2,F3) loop to 10, and the (F1,F2,F3) loop to expression 11.

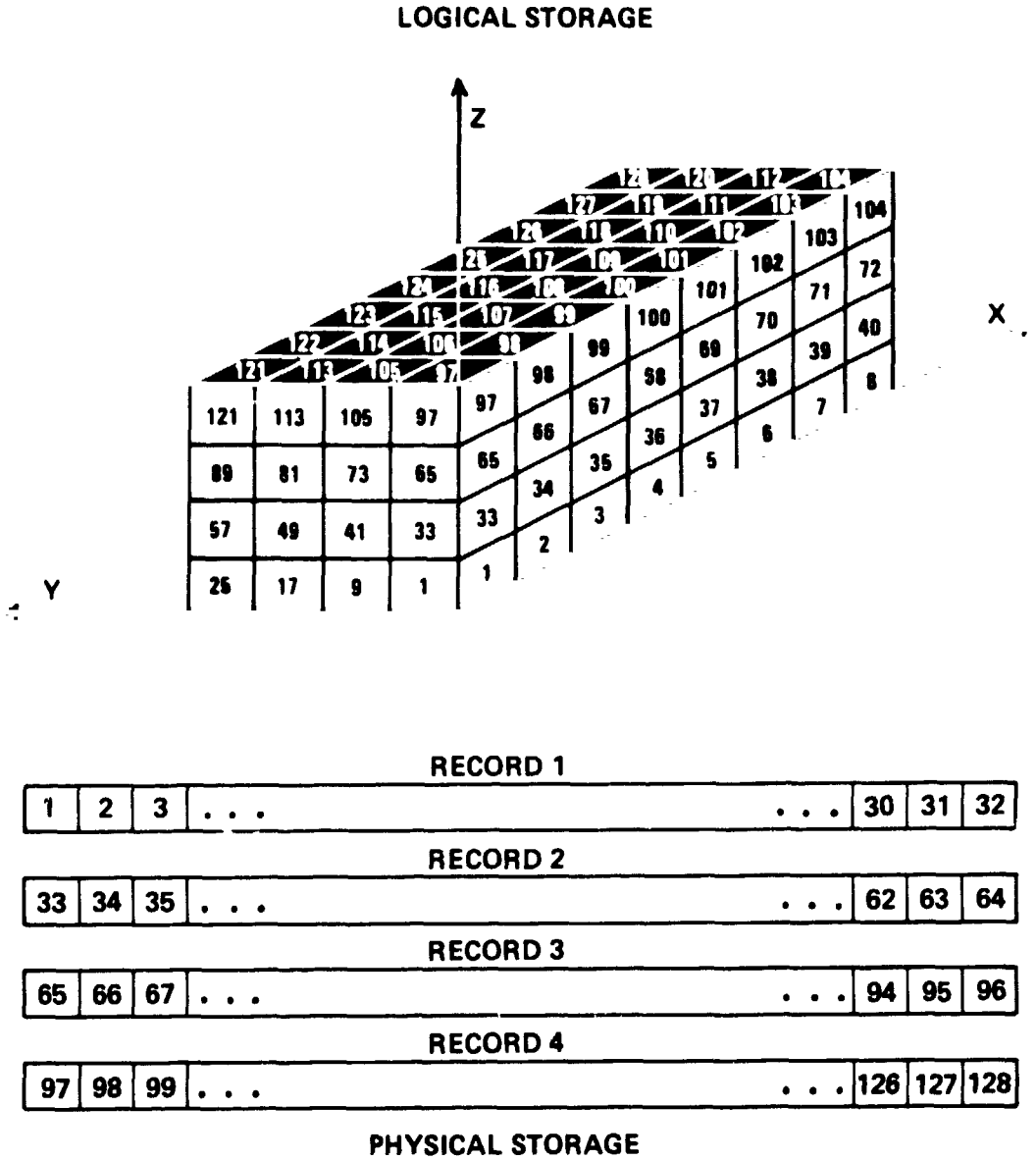


Figure 40. Example of logical and physical storage for program FTURB.

```
: T=00004 IS ON CR00033 USING 00049 BLKS R=0000

FTN4X,L
#FILES(0,2)
      PROGRAM FTURB
C
C-----------------------------------------------------------------
C PROGRAM FTURB: THIS PROGRAM CREATES TRANSFORMED 3-D TURBULENCE BASED
C ON THE VON KARMAN SPECTRAL MODEL. ITS PURPOSE IS TO CREATE THE FROZ-
C EN TURBULENCE DATA BASE FOR THE TURBULENCE SIMULATION MODEL
C
C VARIABLE DEFINITION:
C   N1MAX = MAXIMUM NUMBER OF F1 FREQUENCIES
C   N2MAX = MAXIMUM NUMBER OF F2 FREQUENCIES
C   N3MAX = MAXIMUM NUMBER OF F3 FREQUENCIES
C   FS1   = F1 DIRECTION SAMPLING RATE (DIMENSIONLESS)
C   FS2   = F2 DIRECTION SAMPLING RATE (DIMENSIONLESS)
C   FS3   = F3 DIRECTION SAMPLING RATE (DIMENSIONLESS)
C   ICART = LOGICAL UNIT NUMBER OF THE DISC DRIVE
C   II = VELOCITY COMPONENT (1=U, 2=V, 3=W)
C-----------------------------------------------------------------
C
      DIMENSION LABL(10),LBUF(128)
      COMPLEX X(512),Y(512)
      COMMON N1M,N2M,N3M,INPUT,ICART
      INTEGER*4 N1M,N2M,N3M
C
C  THIS HP LIBRARY ROUTINE IS NECESSARY TO INCREASE I/O BUFFER SIZE.
C
      CALL LGBUF(LBUF,128)
      INPUT = 1
      ICART = 34
      ROOT2 = SQRT(2.)
      WRITE(INPUT,9999)
9999  FORMAT(11HOUTPUT LU=?)
      READ(INPUT,9998) LUOUT
9998  FORMAT(I4)
      WRITE(INPUT,9997)
9997  FORMAT(7HN1MAX=?)
      READ(INPUT,9998) N1MAX
      WRITE(INPUT,9996)
9996  FORMAT(7HN2MAX=?)
      READ(INPUT,9998) N2MAX
      WRITE(INPUT,9995)
9995  FORMAT(7HN3MAX=?)
      READ(INPUT,9998) N3MAX
      WRITE(INPUT,9994)
9994  FORMAT(5HFS1=?)
      READ(INPUT,9993) FS1
9993  FORMAT(F10.0)
      WRITE(INPUT,9992)
9992  FORMAT(5HFS2=?)
      READ(INPUT,9993) FS2
      WRITE(INPUT,9991)
9991  FORMAT(5HFS3=?)
      READ(INPUT,9993) FS3
      WRITE(INPUT,9986)
9986  FORMAT(30H ENTER VELOCITY COMPONENT (1, 2, OR 3))
      READ(1,9985) II
9985  FORMAT(I1)
```

```
      N1M = N1MAX
      N2M = N2MAX
      N3M = N3MAX
      MREC = N1MAX*N2MAX*N3MAX/32
      MREC = N1M*N2M*N3M/32
      WRITE(1,8887) MREC
8887  FORMAT(7HMAXREC=,I7,25H  ENTER FILE TO BE OPENED)
      READ(1,8886) (LABL(I),I=1,10)
8886  FORMAT(10A2)
      OPEN(ICART,FILE=LABL,IOSTAT=IOS,STATUS='NEW',ERR=99,RECL=256,
     *     FORM='UNFORMATTED',ACCESS='DIR',MAXREC=MREC)
      DF1 = FS1/FLOAT(N1MAX)
      DF2 = FS2/FLOAT(N2MAX)
      DF3 = FS3/FLOAT(N3MAX)
      N1D2 = N1MAX/2
      N2D2 = N2MAX/2
      N3D2 = N3MAX/2
      DDDF = DF1*DF2*DF3
      X1MAX = FLOAT(N1MAX)
      X2MAX = FLOAT(N2MAX)
      X3MAX = FLOAT(N3MAX)
      FFF = FS1*FS2*FS3
      SIGXK = SQRT(X1MAX*X2MAX*X3MAX/2.)/FFF
      P = PHIU(0.,0.,0.,II)
      SUM = P
      H = SQRT(FFF*P)
      X(1) = SIGXK*ROOT2*H*CMPLX(GRAN(),0.)
C
C (F1,0,0) LOOP
C
      DO 25 K1 = 2,N1D2
        F1 = (FLOAT(K1)-1.)*DF1
        P = PHIU(F1,0.,0.,II)
        SUM = SUM +2.*P
        H  = SQRT(FFF*P)
        X(K1) = SIGXK*H*CMPLX(GRAN(),GRAN())
        X(N1MAX-K1+2) = CONJG(X(K1))
25    CONTINUE
      F1 = F1 + DF1
      P = PHIU(F1,0.,0.,II)
      SUM = SUM + P
      H = SQRT(FFF*P)
      X(N1D2+1) = SIGXK*ROOT2*H*CMPLX(GRAN(),0.)
      CALL FOURP(2,1,N1MAX,1,1,1,1,X)
C
C (0,F2,0) LOOP
C
      DO 30 K2 = 2,N2D2
        F2 = (FLOAT(K2)-1.)*DF2
        P = PHIU(0.,F2,0.,II)
        SUM = SUM + 2.*P
        H = SQRT(FFF*P)
        X(K2) = SIGXK*H*CMPLX(GRAN(),GRAN())
        X(N2MAX-K2+2) = CONJG(X(K2))
30    CONTINUE
      F2 = F2 + DF2
      P = PHIU(0.,F2,0.,II)
      SUM = SUM + P
      H = SQRT(FFF*P)
      X(N2D2+1) = SIGXK*ROOT2*H*CMPLX(GRAN(),0.)
```

```
      CALL FCHRP(2,1,1,2,N2MAX,1,1,X(2))
C
C (0,0,F3) LOOP
C
      DO 35 K3=2,N3D2
        F3 = (FLOAT(K3)-1.)*DF3
        P = PHIU(0.,0.,F3,II)
        SUM = SUM + 2.*P
        H = SQRT(FFF*P)
        X(K3) = SIGXK*H*CMPLX(GRAN(),GRAN())
        X(N3MAX-K3+2) = CONJG(X(K3))
35    CONTINUE
      F3 = F3 + DF3
      P = PHIU(0.,0.,F3,II)
      H = SQRT(FFF*P)
      X(N3D2+1) = SIGXK*ROOT2*H*CMPLX(GRAN(),0.)
      CALL FCHRP(2,1,1,1,2,N3MAX,X(2))
C
C (F1,F2,0) LOOP
C
      DO 45 K1=2,N1D2
        F1 = (FLOAT(K1)-1.)*DF1
        DO 40 K2 = 2,N2MAX
          F2 = (FLOAT(K2)-1.)*DF2
          P = PHIU(F1,F2,0.,II)
          SUM = SUM + 2.*P
          H = SQRT(FFF*P)
          X(K2) = SIGXK*H*CMPLX(GRAN(),GRAN())
          Y(N2MAX-K2+2) = CONJG(X(K2))
40      CONTINUE
        CALL FCHRP(2,K1,K1,2,N2MAX,1,1,X(2))
        CALL FCHRP(2,N1MAX-K1+2,N1MAX-K1+2,2,N2MAX,1,1,Y(2))
45    CONTINUE
      F1 = F1 + DF1
      DO 47 K2=2,N2D2
        F2 = (FLOAT(K2)-1.)*DF2
        P = PHIU(F1,F2,0.,II)
        SUM = SUM + 2.*P
        H = SQRT(FFF*P)
        X(K2) = SIGXK*H*CMPLX(GRAN(),GRAN())
        X(N2MAX-K2+2) = CONJG(X(K2))
47    CONTINUE
      F2 = F2 + DF2
      P = PHIU(F1,F2,0.,II)
      SUM = SUM + P
      H = SQRT(FFF*P)
      X(N2D2+1) = SIGXK*ROOT2*H*CMPLX(GRAN(),0.)
      CALL FCHRP(2,N1D2+1,N1D2+1,2,N2MAX,1,1,X(2))
C
C (F1,0,F3) LOOP
C
      DO 55 K1=2,N1D2
        F1 = (FLOAT(K1)-1.)*DF1
        DO 50 K3=2,N3MAX
          F3 = (FLOAT(K3)-1.)*DF3
          P = PHIU(F1,0.,F3,II)
          SUM = SUM + 2.*P
          H = SQRT(FFF*P)
          X(K3) = SIGXK*H*CMPLX(GRAN(),GRAN())
          Y(N3MAX-K3+2) = CONJG(X(K3))
```

```
50      CONTINUE
        CALL FCHRP(2,K1,K1,1,1,2,N3MAX,X(2))
        CALL FCHRP(2,N1MAX-K1+2,N1MAX-K1+2,1,1,2,N3MAX,Y(2))
55    CONTINUE
      F1 = F1 + DF1
      DO 57 K3 = 2,N3D2
        F3 = (FLOAT(K3)-1.)*DF3
        P = PHIU(F1,0.,F3,II)
        SUM = SUM + 2.*P
        H = SQRT(FFF*P)
        X(K3) = SIGXK*H*CMPLX(GRAN(),GRAN())
        X(N3MAX-K3+2) = CONJG(X(K3))
57    CONTINUE
      F3 = F3 + DF3
      P = PHIU(F1,0.,F3,II)
      SUM = SUM + P
      H = SQRT(FFF*P)
      X(N3D2+1) = SIGXK*ROOT2*H*CMPLX(GRAN(),0.)
      CALL FCHRP(2,N1D2+1,N1D2+1,1,1,2,N3MAX,X(2))
C
C (0,F2,F3) LOOP
C
      DO 65 K2=2,N2D2
        F2 = (FLOAT(K2)-1.)*DF2
        DO 60 K3=2,N3MAX
          F3 = (FLOAT(K3)-1.)*DF3
          P = PHIU(0.,F2,F3,II)
          SUM = SUM + 2.*P
          H = SQRT(FFF*P)
          X(K3) = SIGXK*H*CMPLX(GRAN(),GRAN())
          Y(N3MAX-K3+2) = CONJG(X(K3))
60      CONTINUE
        CALL FCHRP(2,1,1,K2,K2,2,N3MAX,X(2))
        CALL FCHRP(2,1,1,N2MAX-K2+2,N2MAX-K2+2,2,N3MAX,Y(2))
65    CONTINUE
      F2 = F2 + DF2
      DO 67 K3=2,N3D2
        F3 = (FLOAT(K3)-1.)*DF3
        P = PHIU(0.,F2,F3,II)
        SUM = SUM + 2.*P
        H = SQRT(FFF*P)
        X(K3) = SIGXK*H*CMPLX(GRAN(),GRAN())
        X(N3MAX-K3+2) = CONJG(X(K3))
67    CONTINUE
      F3 = F3 + DF3
      P = PHIU(0.,F2,F3,II)
      SUM = SUM + P
      X(N3D2+1) = SIGXK*H*ROOT2*CMPLX(GRAN(),0.)
      CALL FCHRP(2,1,1,N2D2+1,N2D2+1,2,N3MAX,X(2))
C
C (F1,F2,F3) LOOP
C
      DO 300 K1=2,N1D2
        F1 = (FLOAT(K1)-1.)*DF1
        DO 200 K2=2,N2MAX
          F2 = (FLOAT(K2)-1.)*DF2
          DO 100 K3=2,N3MAX
            F3 = (FLOAT(K3)-1.)*DF3
            P = PHIU(F1,F2,F3,II)
            SUM = SUM + 2.*P
```

```
            H = SQRT(FFF*P)
            X(K3) = SIGXK*H*CMPLX(GRAN(),GRAN())
            Y(N3MAX-K3+2) = CONJG(X(K3))
100       CONTINUE
          CALL FCHRP(2,K1,K1,K2,K2,2,N3MAX,X(2))
          CALL FCHRP(2,N1MAX-K1+2,N1MAX-K1+2,N2MAX-K2+2,N2MAX-K2+2,
     *                2,N3MAX,Y(2))
200     CONTINUE
300   CONTINUE
      F1 = F1 + DF1
      DO 350 K2=2,N2D2
        F2 = (FLOAT(K2)-1.)*DF2
        DO 325 K3=2,N3MAX
          F3 = (FLOAT(K3)-1.)*DF3
          P = PHIU(F1,F2,F3,II)
          SUM = SUM + 2.*P
          X(K3) = SIGXK*H*CMPLX(GRAN(),GRAN())
          Y(N3MAX-K3+2) = CONJG(X(K3))
325     CONTINUE
        CALL FCHRP(2,N1D2+1,N1D2+1,K2,K2,2,N3MAX,X(2))
        CALL FCHRP(2,N1D2+1,N1D2+1,N2MAX-K2+2,N2MAX-K2+2,2,N3MAX,Y(2))
350   CONTINUE
      F2 = F2 + DF2
      DO 375 K3=2,N3D2
        F3 = (FLOAT(K3)-1.)*DF3
        P = PHIU(F1,F2,F3,II)
        SUM = SUM + 2.*P
        H = SQRT(FFF*P)
        X(K3) = SIGXK*H*CMPLX(GRAN(),GRAN())
        X(N3MAX-K3+2) = CONJG(X(K3))
375   CONTINUE
      F3 = F3 + DF3
      P = PHIU(F1,F2,F3,II)
      SUM = SUM + P
      H = SQRT(FFF*P)
      X(N3D2+1) = SIGXK*H*CMPLX(GRAN(),0.)
      CALL FCHRP(2,N1D2+1,N1D2+1,N2D2+1,N2D2+1,2,N3MAX,X(2))
C
      VAR = SUM*DDDF
      WRITE(LUOUT,9990) N1MAX,N2MAX,N3MAX
9990  FORMAT(7H N1MAX=,I4,2X,6HN2MAX=,I4,2X,6HN3MAX=,I4)
      WRITE(LUOUT,9989)
9989  FORMAT(/)
      WRITE(LUOUT,9988) FS1,FS2,FS3
9988  FORMAT(5H FS1=,F10.4,2X,4HFS2=,F10.4,2X,4HFS3=,F10.4)
      WRITE(LUOUT,9989)
      WRITE(LUOUT,9987) VAR
9987  FORMAT(5H VAR=,F10.4)
      CLOSE(ICART,IOSTAT=IOS,ERR=99,STATUS='KEEP')
      STOP
99    WRITE(1,8888) IOS
8888  FORMAT("IOSTAT ERROR& "I4)
      STOP
      END
C
C-----------------------------------------------------------------------
C FUNCTION PHIU:
C THIS FUNCTION CALCULATES THE 3-D VON KARMAN SPECTRAL FUNCTION PHIII
C IN NONDIMENSIONAL FORM
C
```

```
C     F1,F2,F3 ARE THE 3 COMPONENTS OF SPATIAL FREQUENCY
C     TLS IS THE TURBULENT LENGTH SCALE
C-----------------------------------------------------------------------
C
      FUNCTION PHIU(F1,F2,F3,II)
      A = 1.339
      A4 = A**4
      PI = 4.*ATAN(1.)
      FSQ = F1*F1 + F2*F2 + F3*F3
      ARG = FSQ*(2.*PI*A)**2
      DENOM = (1. + ARG)**(17./6.)
      IF(II .EQ. 1) THEN
        FF = F1
      ELSE
        IF(II .EQ. 2) THEN
          FF = F2
        ELSE
          FF = F3
        ENDIF
      ENDIF
      PHIU = (440./9.)*PI**3*A4*(FSQ-FF**2)/DENOM
      RETURN
      END
C
C-----------------------------------------------------------------------
C SUBROUTINE: FCHRP (FETCH REPLACE) THIS ROUTINE READS OR WRITES
C             THE FOLLOWING                (X(N1,N2,N3),N1=N1MIN,N1MAX)
C                                       OR (X(N1,N2,N3),N2=N2MIN,N2MAX)
C                                       OR (X(N1,N2,N3),N3=N3MIN,N3MAX)
C             FROM A RANDOM ACCESS DISC FILE.
C-----------------------------------------------------------------------
C
C
      SUBROUTINE FCHRP(IFOR,N1L,N1H,N2L,N2H,N3L,N3H,T)
      COMMON N1M,N2M,N3M,INPUT,ICART
      COMPLEX T(1),ZBUFF(32)
      INTEGER*4 IL,IH,N1M,N2M,N3M
      INTEGER DIDN
C
      IL = N1L+(N2L-1)*N1M+(N3L-1)*N2M*N1M
      IH = N1H+(N2H-1)*N1M+(N3H-1)*N2M*N1M
C
      NRECL = (IL-1)/32 + 1
      NRECH = (IH-1)/32 + 1
C
      IBL = IL - (NRECL-1)*32
C
      DIDN = 0
      IF(N1L .NE. N1H)  DIDN = 1
      IF(N2L .NE. N2H)  DIDN = N1M
      IF(N3L .NE. N3H)  DIDN = N1M*N2M
      IF(DIDN .NE. 0) GO TO 25
        WRITE(INPUT,9999)
9999    FORMAT(13HERROR FETCH 1)
        STOP
C
25    IS = 1
      IB = IBL
      NSTEP = DIDN/32
      IF(NSTEP .LT. 1) NSTEP = 1
```

```
      DO 100 IREC = NRECL,NRECH,NSTEP
        READ(ICART,REC=IREC,IOSTAT=IOS,ERR=99) (ZBUFF(I),I=1,32)
50      IF(IFOR .EQ. 1) T(IS) = ZBUFF(IB)
        IF(IFOR .EQ. 2) ZBUFF(IB) = T(IS)
        IS = IS + 1
        IB = IB + DIDN
        IF(IB .LE. 32) GO TO 50
        IB = MOD(IB,32)
        IF(IB .EQ. 0) IB = 32
        IF(IFOR .EQ. 2) WRITE(ICART,REC=IREC,IOSTAT=IOS,ERR=99)
     *                   (ZBUFF(I),I=1,32)
100   CONTINUE
      RETURN
99    WRITE(IMPUT,9998) IOS
9998  FORMAT(19HFETCH IOSTAT ERROR ,I3)
      STOP
      END
      END$
```

APPENDIX G. TRANSFORMATION OF FREQUENCY DOMAIN "TURBULENCE" TO THE SPACE DOMAIN

Program FFT3D transforms the file created by FTURB to the space domain via the inverse FFT. FFT3D contains a call to library routine LGBUF which was explained in Appendix F. The transform is calculated by operating on small lines (one-dimensional) from the data in the previously created file. This program is extremely memory efficient but time inefficient. Core storage is minimal but numerous disc file reads and writes are required.

The three-dimensional FFT is calculated using a simple one-dimensional FFT routine. The three-dimensional inverse FFT is defined by

$$x(n_1,n_2,n_3) = \frac{1}{M_1M_2M_3} \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} \sum_{k_3=0}^{M_3-1} X(k_1,k_2,k_3)\, e^{j2\pi\left(\frac{n_1k_1}{M_1} + \frac{n_2k_2}{M_2} + \frac{n_3k_3}{M_3}\right)} \tag{95}$$

This equation is broken up into three groups of one-dimensional transforms as follows

$$A(k_1,k_2,k_3) = \frac{1}{M_1} \sum_{k_3=0}^{M_3-1} X(k_1,k_2,k_3)\, e^{j2\pi\ n_3k_3/M_3} \tag{96}$$

The above procedure must be performed $M_1 \times M_2$ times and is implemented in the K3 loop of the main program. The next operation is given in the following equation

$$B(k_1,n_2,n_3) = \frac{1}{M_2} \sum_{n_2=0}^{M_2-1} A(k_1,k_2,k_3)\, e^{j2\pi\ n_2k_2/M_2} \tag{97}$$

This transform is performed $M_1 \times M_3$ times. The third operation is given by

$$x(n_1,n_2,n_3) = \frac{1}{M_1} \sum_{k_1=0}^{M_1-1} B(k_1,n_2,n_3)\, e^{j2\pi\, n_1 k_1/M_1} \qquad (98)$$

and is performed $M_2 \times M_3$ times in K1 loop. For a 64 x 64 x 64 array only 64 complex words must be in memory at a given time.

```
3D T=00004 IS ON CR00033 USING 00037 BLKS R=0000

  FTN4X,L
  #FILES(0,2)
        PROGRAM FFT3D
  C
  C-----------------------------------------------------------------------
  C PROGRAM FFT3D: THIS PROGRAM CALCULATES A THREE DIMENSIONAL FFT IN
  C PLACE FOR DATA RESIDING IN MASS STORAGE (TYPE 1 DIRECT ACCESS FILE)
  C-----------------------------------------------------------------------
  C
        DIMENSION LABL(10),LBUF(128)
        COMPLEX X(512)
        COMMON N1MAX,N2MAX,N3MAX,INPUT,ICART,K1,K2,K3
        INTEGER*4 N1MAX,N2MAX,N3MAX
        CALL LGBUF(LBUF,128)
        INPUT = 1
        ICART = 34
        ROOT2 = SQRT(2.)
        WRITE(INPUT,9999)
  9999  FORMAT(11HOUTPUT LU=?)
        READ(INPUT,9998) LUOUT
  9998  FORMAT(I4)
        WRITE(INPUT,9997)
  9997  FORMAT(7HN1MAX=?)
        READ(INPUT,9998) N1MAX
        WRITE(INPUT,9996)
  9996  FORMAT(7HN2MAX=?)
        READ(INPUT,9998) N2MAX
        WRITE(INPUT,9995)
  9995  FORMAT(7HN3MAX=?)
        READ(INPUT,9998) N3MAX
        WRITE(INPUT,9994)
  9994  FORMAT(5HFS1=?)
        READ(INPUT,9993) FS1
  9993  FORMAT(F10.0)
        WRITE(INPUT,9992)
  9992  FORMAT(5HFS2=?)
        READ(INPUT,9993) FS2
        WRITE(INPUT,9991)
  9991  FORMAT(5HFS3=?)
        READ(INPUT,9993) FS3
        MREC = N1MAX*N2MAX*N3MAX/32
        WRITE(1,8887) MREC
  8887  FORMAT(7HMAXREC=,I7,25H  ENTER FILE TO BE OPENED)
        READ(1,8886) (LABL(I),I=1,10)
  8886  FORMAT(10A2)
        OPEN(ICART,FILE=LABL,IOSTAT=IOS,STATUS='OLD',ERR=99,RECL=256,
       *      FORM='UNFORMATTED',ACCESS='DIR',MAXREC=MREC)
        X1MAX = FLOAT(N1MAX)
        X2MAX = FLOAT(N2MAX)
        X3MAX = FLOAT(N3MAX)
        DF1 = FS1/X1MAX
        DF2 = FS2/X2MAX
        DF3 = FS3/X3MAX
        N1D2 = N1MAX/2
        N2D2 = N2MAX/2
        N3D2 = N3MAX/2
        DDDF = DF1*DF2*DF3
        FFF = FS1*FS2*FS3
```

```
      N1M = N1MAX
      N2M = N2MAX
      N3M = N3MAX
      WRITE(1,6999)
6999  FORMAT("BEGIN K3 LOOP")
C
C K3 LOOP
C
      DO 200 K1=1,N1MAX
        DO 100 K2=1,N2MAX
          WRITE(1,6998) K1,K2
6998      FORMAT(" K1="I3"  K2="I3"$A")
          CALL FCHRP(1,K1,K1,K2,K2,1,N3M,X)
          CALL FFT(X,N3MAX,1)
          CALL FCHRP(2,K1,K1,K2,K2,1,N3M,X)
100     CONTINUE
200   CONTINUE
C
C K2 LOOP
C
      WRITE(1,6997)
6997  FORMAT("BEGIN K2 LOOP")
      DO 400 K1=1,N1MAX
        DO 300 K3=1,N3MAX
          WRITE(1,6996) K1,K3
6996      FORMAT("K1="I3"  K3="I3"$A")
          CALL FCHRP(1,K1,K1,1,N2M,K3,K3,X)
          CALL FFT(X,N2MAX,1)
          CALL FCHRP(2,K1,K1,1,N2M,K3,K3,X)
300     CONTINUE
400   CONTINUE
C
C K1 LOOP
C
      WRITE(1,6995)
6995  FORMAT("BEGIN K1 LOOP")
      DO 600 K2=1,N2MAX
        DO 500 K3=1,N3MAX
        WRITE(1,6994) K2,K3
6994      FORMAT("K2="I3"  K3="I3"$A")
          CALL FCHRP(1,1,N1M,K2,K2,K3,K3,X)
          CALL FFT(X,N1MAX,1)
          CALL FCHRP(2,1,N1M,K2,K2,K3,K3,X)
500     CONTINUE
600   CONTINUE
C
C MULTIPLY BY CONSTANT
C
      WRITE(1,6993)
6993  FORMAT("MULTIPLY BY CONSTANT")
      CONST = FFF
      WRITE(1,5554) CONST
5554  FORMAT("CONST=$"E12.5)
      DO 950 K3=1,N3MAX
        DO 900 K2=1,N2MAX
          CALL FCHRP(1,1,N1M,K2,K2,K3,K3,X)
          DO 800 J=1,N1MAX
            X(J) = X(J)*CONST
800       CONTINUE
          CALL FCHRP(2,1,N1M,K2,K2,K3,K3,X)
```

```
900     CONTINUE
950   CONTINUE
      CLOSE(ICART,STATUS='KEEP',IOSTAT=IOS,ERR=99)
      STOP
99    WRITE(1,5555) IOS
5555  FORMAT(17HIOSTAT ERROR NO. ,I4)
      STOP
      END
C
C-------------------------------------------------------------------
C SUBROUTINE: FCHRP (FETCH REPLACE) THIS ROUTINE READS OR WRITES
C             THE FOLLOWING              (X(N1,N2,N3),N1=1,N1MAX)
C                                       OR (X(N1,N2,N3),N2=1,N2MAX)
C                                       OR (X(N1,N2,N3),N3=1,N3MAX)
C             FROM A RANDOM ACCESS DISC FILE.
C-------------------------------------------------------------------
C
C
      SUBROUTINE FCHRP(IFOR,N1L,N1H,N2L,N2H,N3L,N3H,T)
      COMMON N1MAX,N2MAX,N3MAX,INPUT,ICART,K1,K2,K3
      INTEGER*4 N1MAX,N2MAX,N3MAX,IL,IH
      COMPLEX T(1),ZBUFF(32)
      INTEGER DIDN
C
      IL = N1L+(N2L-1)*N1MAX+(N3L-1)*N2MAX*N1MAX
      IH = N1H+(N2H-1)*N1MAX+(N3H-1)*N2MAX*N1MAX
C
      NRECL = (IL-1)/32 + 1
      NRECH = (IH-1)/32 + 1
C
      IBL = IL - (NRECL-1)*32
C
      DIDN = 0
      IF(N1L .NE. N1H)  DIDN = 1
      IF(N2L .NE. N2H)  DIDN = N1MAX
      IF(N3L .NE. N3H)  DIDN = N1MAX*N2MAX
      IF(DIDN .NE. 0) GO TO 25
        WRITE(INPUT,9999)
9999    FORMAT(13HERROR FETCH 1)
        STOP
C
25    IS = 1
      IB = IBL
      NSTEP = DIDN/32
      IF(NSTEP .LT. 1) NSTEP = 1
      DO 100 IREC = NRECL,NRECH,NSTEP
        READ(ICART,REC=IREC,IOSTAT=IOS,ERR=99) (ZBUFF(I),I=1,32)
50      IF(IFOR .EQ. 1) T(IS) = ZBUFF(IB)
        IF(IFOR .EQ. 2) ZBUFF(IB) = T(IS)
        IF(IB.GT.32) GO TO 500
        IF(IB.LT.1) GO TO 500
        IF(IS.LT.1) GO TO 500
        IF(IS.GT.512) GO TO 500
        IF(CABS(T(IS)).GT.1.E5) GO TO 500
        IF(CABS(ZBUFF(IB)).GT.1.E5) GO TO 500
        GO TO 600
500       WRITE(1,7998) K1,K2,K3
          WRITE(6,7998) K1,K2,K3
          WRITE(1,7999) IB,ZBUFF(IB),IS,T(IS)
          WRITE(6,7999) IB,ZBUFF(IB),IS,T(IS)
```

```
7999      FORMAT(" ZBUFF("I5")="2(X,E12.5)"  T("I5")="2(X,E12.5))
7998      FORMAT(" K1="I5"  K2="I5"  K3="I5)
600     IS = IS + 1
        IB = IB + DIDN
        IF(IB .LE. 32) GO TO 50
        IB = MOD(IB,32)
        IF(IB .EQ. 0) IB = 32
        IF(IFOR .EQ. 2) WRITE(ICART,REC=IREC,IOSTAT=IOS,ERR=99)
     *                  (ZBUFF(I),I=1,32)
100   CONTINUE
      RETURN
99    WRITE(INPUT,9998) IOS
9998  FORMAT(19HFETCH IOSTAT ERROR ,I3)
      STOP
      END
C
C-----------------------------------------------------------------------
C SUBROUTINE:   FFT
C JIM COOLEY'S SIMPLE FFT PROGRAM--USES DECIMATION IN TIME ALGORITHM
C X IS AN N=2**M POINT COMPLEX ARRAY THAT INITIALLY CONTAINS THE INPUT
C AND ON OUTPUT CONTAINS THE TRANSFORM
C THE PARAMETER INV SPECIFIED DIRECT TRANSFORM IF 0 AND INVERSE IF 1
C-----------------------------------------------------------------------
C
      SUBROUTINE FFT(X, N, INV)
      COMPLEX X(1)
      COMPLEX U, W, T, CMPLX
      INTEGER*4 N
C
C   X = COMPLEX ARRAY OF SIZE N--ON INPUT X CONTAINS
C       THE SEQUENCE TO BE TRANSFORMED
C       ON OUTPUT X CONTAINS THE DFT OF THE INPUT
C   N = SIZE OF THE FFT TO BE COMPUTED--N=2**M FOR 1.LE.M.LE.15
C INV = PARAMETER TO DETERMINE WHETHER TO DO A DIRECT TRANSFORM (INV=0)
C       OR AN INVERSE TRANSFORM (INV=1)
C
      M = ALOG(FLOAT(N))/ALOG(2.) + .1
      NV2 = N/2
      NM1 = N - 1
      J = 1
      DO 40 I=1,NM1
         IF  (I.GE.J) GO TO 10
         T = X(J)
         X(J) = X(I)
         X(I) = T
10     K = NV2
20     IF (K.GE.J) GO TO 30
       J = J - K
       K = K/2
       GO TO 20
30     J = J + K
40     CONTINUE
       PI = 4.*ATAN(1.)
       DO 70 L=1,M
         LE = 2**L
         LE1 = LE/2
         U = (1.0,0.0)
         W = CMPLX(COS(PI/FLOAT(LE1)),-SIN(PI/FLOAT(LE1)))
         IF (INV.NE.0) W = CONJG(W)
         DO 60 J=1,LE1
```

```
          DO 50 I=J,N,LE
            IP = I + LE1
            T = X(IP)*U
            X(IP) = X(I) - T
            X(I) = X(I) + T
50        CONTINUE
          U = U*W
60      CONTINUE
70    CONTINUE
      IF(INV.EQ.0) RETURN
      DO 80 I=1,N
         X(I) = X(I)/CMPLX(FLOAT(N),0.)
80    CONTINUE
      RETURN
      END
      END$
```

APPENDIX H. FLIGHT OF A POINT "AIRPLANE" THROUGH WINDSHEAR AND TURBULENCE

Program JAWS2 moves a point through the JAWS wind shear and generated turbulence at a constant selectable ground speed. The "airplane" can fly a straight horizontal, three degree glide slope approach, or three degree glide slope departure flight path. Initial positions within both the JAWS data and the turbulence are selectable through flight mode variables MODEF. Two real variable "knobs", JCON and TCON permit an increase in turbulence or turning on or off of the wind shear so that separate components can be examined separately.

Interpolation procedures for this program were described in Chapter VII. The interpolation for the JAWS data is handled by subroutine NTERP. Functions for the calculation of turbulent length scales and gust intensity are handled by functions TLS and SIGX, respectively. The demonstration functional forms of TLS and SIGX are also described in Chapter VII.

Storage for the JAWS data is in cell form as described in Chapter VII. The turbulence is stored in 16 bit integers and are converted to real numbers by dividing by 10000. Storage of the JAWS data is on disc unit 34 and the turbulence on disc unit 37.

```
2 T=00004 IS ON CR00033 USING 00058 BLKS R=0000

 FTN4X,L
 $FILES(0,2)
       PROGRAM JAWS2
 C-----------------------------------------------------------------
 C PROGRAM JAWS2: THIS PROGRAM CALCULATES WINDS FOR A PLANE
 C MOVING THROUGH THE JAWS DATA SET.
 C
 C PROGRAMMER: WARREN CAMPBELL
 C
 C VARIABLE DEFINITION:
 C
 C       NBLPF = NUMBER OF BLOCKS PER FIELD
 C       NBLPL = NUMBER OF BLOCKS PER LINE
 C       ILINE = LINE NUMBER OF CURRENT CELL
 C       IBLOK = PRESENT BLOCK NUMBER
 C       NX1 = NUMBER OF CELLS IN THE X DIRECTION
 C       NY1 = NUMBER OF CELLS IN THE Y DIRECTION
 C       NZ1 = NUMBER OF CELLS IN THE Z DIRECTION
 C       XMIN = MINIMUM VALUE OF X (KM)
 C       XMAX = MAXIMUM VALUE OF X (KM)
 C       YMIN = MINIMUM VALUE OF Y (KM)
 C       YMAX = MAXIMUM VALUE OF Y (KM)
 C       ZMIN = MINIMUM VALUE OF Z (M)
 C       ZMAX = MAXIMUM VALUE OF Z (M)
 C       DELX = X GRID SPACING (M)
 C       DELY = Y GRID SPACING (M)
 C       DELZ = Z GRID SPACING (M)
 C       DELT = TIME STEP (SEC)
 C       IX = X INDEX OF CURRENT CELL
 C       IY = Y INDEX OF CURRENT CELL
 C       IZ = Z INDEX OF CURRENT CELL
 C       IXT = X TURBULENCE INDEX
 C       IYT = Y TURBULENCE INDEX
 C       IZT = Z TURBULENCE INDEX
 C       ITPT = NUMBER OF CURRENT TURBULENCE POINT
 C       XT = DIMENSIONLESS TURBULENCE X LOCATION
 C       YT = DIMENSIONLESS TURBULENCE Y LOCATION
 C       ZT = DIMENSIONLESS TURBULENCE Z LOCATION
 C       N1MAX = MAX NUMBER OF TURBULENCE POINTS IN X DIRECTION
 C       N2MAX = MAX NUMBER OF TURBULENCE POINTS IN Y DIRECTION
 C       N3MAX = MAX NUMBER OF TURBULENCE POINTS IN Z DIRECTION
 C       FS1 = DIMENSIONLESS SAMPLING FREQUENCY IN F1 DIRECTION
 C       FS2 = DIMENSIONLESS SAMPLING FREQUENCY IN F2 DIRECTION
 C       FS3 = DIMENSIONLESS SAMPLING FREQUENCY IN F3 DIRECTION
 C       TCON = TURBULENCE CONSTANT. LARGE VALUE ELIMINATES TURBULENCE.
 C       JCON = JAWS CONSTANT. 1. FOR WIND SHEAR + TURBULENCE
 C                             0. FOR TURBULENCE ONLY
 C       X = CURRENT X LOCATION (KM)
 C       Y = CURRENT Y LOCATION (KM)
 C       Z = CURRENT Z LOCATION (M)
 C       VX = EAST-WEST INERTIAL VELOCITY (POSITIVE EAST IN M SEC)
 C       VY = NORTH-SOUTH INERTIAL VELOCITY (POSITIVE NORTH IN M SEC)
 C       VZ = VERTICAL VELOCITY (POSITIVE UP IN M/SEC)
 C       X0 = INITIAL X VALUE (KM)
 C       Y0 = INITIAL Y VALUE (KM)
 C       Z0 = INITIAL Z VALUE (M)
 C       HDG = HEADING (DEGREES)
 C       VGRS = GROUND SPEED (M/SEC)
```

```
C       VAIRS = AIRSPEED (M/SEC)
C       UWIND = EAST-WEST WIND SPEED (POSITIVE EAST IN M/SEC)
C       VWIND = NORTH-SOUTH WIND SPEED (POSITIVE NORTH IN M/SEC)
C       WWIND = VERTICAL WIND SPEED (POSITIVE UP IN MMSEC)
C       MODEF = FLIGHT MODE
C                   1 = HORIZONTAL FLIGHT
C                   2 = ILS APPROACH (3 DEGREE GLIDE SLOPE)
C                   3 = TAKEOFF (3 DEGREE GLIDE SLOPE)
C       CELL = ARRAY CONTAINING CORNER POINTS FOR PRESENT FIELD IN
C              PRESENT CELL
C       ICELL = CURRENT CELL NUMBER
C       ICWBL = CURRENT CELL NUMBER WITHIN BLOCK (1-16)
C       ICB = BEGINING INDEX OF CELL WITHIN BLOCK (1,9,17,ETC.)
C       ICE = ENDING VALUE OF CELL WITHIN BLOCK = ICB+7
C       XC = X VALUE WITHIN CELL (0-DELX)
C       YC = Y VALUE WITHIN CELL (0-DELY)
C       ZC = Z VALUE WITHIN CELL (0-DELZ)
C       T = TIME
C       NPTS = MAXIMUM NUMBER OF POINTS TO BE CALCULATED. THIS IS A
C              MEANS OF TERMINATING THE PROGRAM EARLY FOR DEBUGGING
C              PURPOSES.
C       IPT = NUMBER OF THE PRESENT POINT BEING CALCULATED
C       SIGMX = MAXIMUM VALUE OF GUST INTENSITY
C-----------------------------------------------------------------------
      DIMENSION LBUFF(128),IA(128),NAME(10),LABEL(40),CELL(8),ITT(128)
      COMMON DELX,DELY,DELZ,D3,SIGMX
      REAL JCON
      INTEGER*4 ICELL,ITPT,N1MAX,N2MAX,N3MAX
      CALL LGBUF(LBUFF,128)
      PI = 4.*ATAN(1.)
      WRITE(1,9999)
9999  FORMAT("ENTER FILE TO BE OPENED$")
      READ(1,9998) (NAME(I),I=1,10)
9998  FORMAT(40A2)
      WRITE(1,9997)
9997  FORMAT("ENTER FLIGHT MODE$"/
     *       "   1 = HORIZONTAL FLIGHT"/
     *       "   2 = ILS APPROACH (3 DEGREE GLIDE SLOPE)"/
     *       "   3 = TAKEOFF (3 DEGREE GLIDE SLOPE)")
      READ(1,*) MODEF
      WRITE(1,9996)
9996  FORMAT("ENTER CONSTANT GROUND SPEED$")
      READ(1,*) VGRS
      WRITE(1,9995)
9995  FORMAT("ENTER TRUE HEADING IN DEGREES$")
      READ(1,*) HDG
      WRITE(1,9994)
9994  FORMAT("ENTER INITIAL POSITION (X0,Y0,Z0) IN (KM,KM,M)$")
      READ(1,*) X0,Y0,Z0
      WRITE(1,9988)
9988  FORMAT("ENTER DELTA T IN SECONDS$")
      READ(1,*) DELT
      WRITE(1,9993)
9993  FORMAT("ENTER TAPE HEADER (40A2)$")
      READ(1,9998) (LABEL(I),I=1,40)
      WRITE(1,9979)
9979  FORMAT("ENTER MAXIMUM NUMBER OF POINTS$")
      READ(1,*) NPTS
      WRITE(1,8999)
8999  FORMAT("ENTER JCON,TCON$")
```

```
      READ(1,*) JCON,TCON
      WRITE(1,8998)
8998  FORMAT("ENTER N1MAX,N2MAX,N3MAXΔ")
      READ(1,*) N1MAX,N2MAX,N3MAX
      WRITE(1,8997)
8997  FORMAT("ENTER FS1,FS2,FS3Δ")
      READ(1,*) FS1,FS2,FS3
      XTMAX = N1MAX/FS1
      YTMAX = N2MAX/FS2
      ZTMAX = N3MAX/FS3
      WRITE(1,8995)
8995  FORMAT("ENTER XT0,YT0,ZT0Δ")
      READ(1,*) XT0,YT0,ZT0
      WRITE(1,7777)
7777  FORMAT("ENTER SIGMXΔ")
      READ(1,*) SIGMX
      WRITE(8,9998) (LABEL(I),I=1,40)
      WRITE(8,9992) MODEF
9992  FORMAT(" FLIGHT MODE = "I2)
      WRITE(8,9991) HDG
9991  FORMAT(" TRUE HEADING = "F6.1" DEGREES")
      WRITE(8,9990) VGRS
9990  FORMAT(" INERTIAL HORIZONTAL VELOCITY="F8.0" M/SEC")
      WRITE(8,9989) X0,Y0,Z0
9989  FORMAT(" INITIAL POSITION: X0="F8.2" KM Y0="F8.2" KM  Z0="
     *   F6.0" M")
      WRITE(8,8994) XT0,YT0,ZT0
8994  FORMAT(" XT0="F8.3"  YT0="F8.3"  ZT0="F8.3)
      WRITE(8,9987) DELT,TCON
9987  FORMAT(" TIME INCREMENT = "F8.3" SECONDS  TCON="E12.4)
      OPEN(34,FILE=NAME,STATUS='OLD',IOSTAT=IOS,ERR=99
     * ,ACCESS='DIR',RECL=256,MAXREC=7201)
      WRITE(1,8996)
8996  FORMAT("ENTER TURBULENCE FILE NAMEΔ")
      READ(1,9998) (NAME(I),I=1,10)
      OPEN(37,FILE=NAME,STATUS='OLD',IOSTAT=IOS,ERR=99
     * ,ACCESS='DIR',RECL=256,MAXREC=2048)
      RFAD(34,REC=1,IOSTAT=IOS,ERR=99) (IA(I),I=1,16)
      NX1 = IA(1) - 1
      NY1 = IA(2) - 1
      NZ1 = IA(3) - 1
      XMIN = IA(7)/100.
      XMAX = IA(8)/100.
      DELX = IA(9)
      YMIN = IA(10)/100.
      YMAX = IA(11)/100.
      DELY = IA(12)
      ZMIN = IA(13)
      ZMAX = IA(14)
      DELZ = IA(15)
      D3 = DELX*DELY*DELZ
      WRITE(8,9986) XMIN,XMAX
9986  FORMAT(" XMIN="F8.2" KM   XMAX="F8.2" KM")
      WRITE(8,9985) YMIN,YMAX
9985  FORMAT(" YMIN="F8.2" KM   YMAX="F8.2" KM")
      WRITE(8,9984) ZMIN,ZMAX
9984  FORMAT(" ZMIN="F8.2" KM   ZMAX="F8.2" KM")
      WRITE(8,9983) DELX,DELY,DELZ
9983  FORMAT(" DELX="F6.0" M   DELY="F6.0" M   DELZ="F6.0" M")
      WRITE(8,9960) NPTS
```

```
9960  FORMAT(" NPTS="I5)
      WRITE(8,9982)
9982  FORMAT(8X"T"9X"X"9X"Y"9X"Z"7X"UWIND"5X"VWIND"5X"WWIND"5X
     *"VAIRS")

C********** CALCULATE # OF BLKS PER LINE AND # BLKS PER FIELD *******
      IF(MOD(NX1,16) .EQ. 0) THEN
        NBLPL = NX1/16
      ELSE
        NBLPL = NX1/16 + 1
      ENDIF
      NBLPF = NBLPL*NY1*NZ1

C******** CALCULATE VX,VY,VZ *********
      THETA = PI*HDG/180.
      VX = VGRS*SIN(THETA)
      VY = VGRS*COS(THETA)
      IF(MODEF .EQ. 1) THEN
        VZ = 0.
      ELSE
        IF(MODEF .EQ. 2) THEN
          VZ = -VGRS*TAN(3.*PI/180.)
        ELSE
          VZ = VGRS*TAN(3.*PI/180.)
        ENDIF
      ENDIF

C****** INITIALIZE POSITION AND TIME ********
      X = X0
      Y = Y0
      Z = Z0
      T = 0.
      IPT = 1
      XT = XT0
      YT = YT0
      ZT = ZT0

C****** CALCULATE CURRENT CELL INDICES, IX, IY, IZ, IXT, ETC. *****
50    IX = 1000.*(X - XMIN)/DELX + 1
      IY = 1000.*(Y - YMIN)/DELY + 1
      IZ = (Z - ZMIN)/DELZ + 1
      IF(XT .GT. XTMAX) XT = XT - XTMAX
      IF(YT .GT. YTMAX) YT = YT - YTMAX
      IF(ZT .GT. ZTMAX) ZT = ZT - ZTMAX
      IF(XT .LT. 0.) XT = XT + XTMAX
      IF(YT .LT. 0.) YT = YT + YTMAX
      IF(ZT .LT. 0.) ZT = ZT + ZTMAX
      IXT = IFIX(XT*FS1) + 1
      IYT = IFIX(YT*FS2) + 1
      IZT = IFIX(ZT*FS3) + 1
D     WRITE(6,5000) IXT,IYT,IZT
5000  FORMAT(" IXT="I5"  IYT="I5"  IZT="I5)
D     WRITE(6,5001) XT,YT,ZT
5001  FORMAT(" XT="F8.3"  YT="F8.3"  ZT="F8.3)
      ITPT = IXT + (IYT-1)*N1MAX + (IZT-1)*N2MAX*N3MAX
      IF(MOD(ITPT,128) .EQ. 0) THEN
        ITBLK = ITPT/128
      ELSE
        ITBLK = ITPT/128 + 1
      ENDIF
```

```
      IPWB = ITPT - (ITBLK-1)*128
D     WRITE(6,5005) ITPT,ITBLK
5005  FORMAT(" ITPT="I8"  ITBLK="I5)

C*****  CALCULATE UWIND ****
      ICELL = IX + NX1*(IY-1) + (IZ-1)*NX1*NY1
      ILINE = IY + (IZ-1)*NY1
      IF(MOD(IX,16) .EQ. 0) THEN
        ICWB = 16
        IBLOK = (ILINE-1)*NBLPL + IX/16 + 1
      ELSE
        ICWB = MOD(IX,16)
        IBLOK = (ILINE-1)*NBLPL + IX/16 + 2
      ENDIF
      READ(34,REC=IBLOK,IOSTAT=IOS,ERR=99) (IA(I),I=1,128)
      ICB = (ICWB-1)*8 + 1
      ICE = ICB + 7
      IC = 1
      DO 100 IB=ICB,ICE
        CELL(IC) = IA(IB)/100.
        IC = IC + 1
100   CONTINUE
      READ(37,REC=ITBLK,IOSTAT=IOS,ERR=99) (ITT(I),I=1,128)
      IF(JCON .LT. 0.001) THEN
        UTRB = ITT(IPWB)/(10000.*TCON)
      ELSE
        UTRB = ITT(IPWB)*SIGX(X,Y,Z)/(10000.*TCON)
      ENDIF
      XC = 1000.*(X-XMIN) - (IX-1)*DELX
      YC = 1000.*(Y-YMIN) - (IY-1)*DELY
      ZC = Z - ZMIN - (IZ-1)*DELZ
      CALL NTERP(CELL,XC,YC,ZC,UWIND)
      UWIND = JCON*UWIND + UTRB
D     WRITE(6,5003) SIGX(X,Y,Z),UTRB
5003  FORMAT(" SIGX="F8.3"  UTRB="F8.3)

C*****  CALCULATE VWIND *****
      IBLOK = IBLOK + NBLPF
      READ(34,REC=IBLOK,IOSTAT=IOS,ERR=99) (IA(I),I=1,128)
      IC = 1
      DO 125 IB=ICB,ICE
        CELL(IC) = IA(IB)/100.
        IC = IC + 1
125   CONTINUE
      CALL NTERP(CELL,XC,YC,ZC,VWIND)

C*****  CALCULATE W  *****
      IBLOK = IBLOK + NBLPF
      READ(34,REC=IBLOK,IOSTAT=IOS,ERR=99) (IA(I),I=1,128)
      IC = 1
      DO 150 IB=ICB,ICE
        CELL(IC) = IA(IB)/100.
        IC = IC + 1
150   CONTINUE
      CALL NTERP(CELL,XC,YC,ZC,WWIND)

C***** CALCULATE AIRSPEED *****
      VAIRS = VGRS - UWIND*SIN(THETA) - VWIND*COS(THETA)

C***** WRITE RESULTS *****
```

```
      WRITE(8,9981) T,X,Y,Z,UWIND,VWIND,WWIND,VAIRS
9981  FORMAT(5X,F6.2,7(2X,F8.2))

C*****  INCREMENT TIME AND POSITION  *****
      T = T + DELT
      X = X + VX*DELT/1000.
      Y = Y + VY*DELT/1000.
      Z = Z + VZ*DELT
      XT = XT + VX*DELT/TLS(X,Y,Z)
      YT = YT + VY*DELT/TLS(X,Y,Z)
      ZT = ZT + VZ*DELT/TLS(X,Y,Z)

      IF(IPT .LT. NPTS) THEN
        IPT = IPT + 1
        GO TO 250
      ELSE
        ENDFILE 8
        CLOSE(34,STATUS='KEEP',IOSTAT=IOS,ERR=99)
        CLOSE(37,STATUS='KEEP',IOSTAT=IOS,ERR=99)
        STOP
      ENDIF

250   IF((X.GT.XMIN .AND. X.LT.XMAX) .AND.
     *   (Y.GT.YMIN .AND. Y.LT.YMAX) .AND.
     *   (Z.GT.ZMIN .AND. Z.LT.ZMAX)) THEN
            GO TO 50
      ELSE
            ENDFILE 8
            CLOSE(34,STATUS='KEEP',IOSTAT=IOS,ERR=99)
            CLOSE(37,STATUS='KEEP',IOSTAT=IOS,ERR=99)
      ENDIF
      STOP
99    WRITE(1,9980) IOS
9980  FORMAT("IOSTAT ERROR #"I4)
      END

      FUNCTION TLS(X,Y,Z)
C-----------------------------------------------------------------------
C FUNCTION TLS: TURBULENT LENGTH SCALE FUNCTION. FUNCTIONAL VALUE
C IS FROM ED41 TERRESTRIAL ENVIRONMENT DOCUMENT.
C-----------------------------------------------------------------------
      TLS = 31.5*(Z/18.3)**0.64
      RETURN
      END

      FUNCTION SIGX(X,Y,Z)
C-------------------------------------------------------------
C FUNCTION SIGX: THIS ROUTINE CALCULATES THE U TURBULENCE
C STANDARD DEVIATION FOR THE JAWS JULY 14,1982 CASE.
C-------------------------------------------------------------
      COMMON DELX,DELY,DELZ,D3,SIGMX

C*************************************************************
C (XCTR,YCTR) ARE THE COORDINATES OF THE MICROBURST CENTER
C IN KILOMETERS.
C*************************************************************
      XCTR = 14.2
      YCTR = -1.5
      R = SQRT((X-XCTR)**2 + (Y-YCTR)**2)
```

```
C*****************************************************************
C CALCULATE THE XY FACTOR. THIS IS BASED ON B-57B MEASUREMNTS
C FROM THE SAME DATE. THE 57 ENCOUNTERED THE UPPER PART OF A
C MICROBURST AND EXPERIENCED A DECREASE IN TURBULENCE AS IT PASSED
C THROUGH THE DOWNDRAFT. FOR THE SAME RUN THE AVERAGE OF SIGX,SIGY,
C AND SIGZ WAS ABOUT 6 M/SEC. FXY CAUSES A DECREASE IN SIGX OF
C ABOUT 6 AWAY FROM THE OUTFLOW CENTER, AND 3 IN THE CENTER. ANOTHER
C TERM IN FXY CAUSES SIGX TO DECREASE TO EXP(-.5) AT A DISTANCE OF
C 5 KM. THIS IS IN KEEPING WITH A DECREASE IN TURBULENCE AWAY FROM
C WIND SHEAR.
C*****************************************************************
      FXY = (SIGMX - .5*SIGMX*EXP(-R*R/(.75**2*2.)))*EXP(-R*R/50.)
      IF(Z .GT. 300.) THEN
        FZ = EXP((Z-300.)/200.)
      ELSE
        FZ = 1.
      ENDIF
      SIGX = FXY*FZ
      RETURN
      END

      SUBROUTINE NTERP(CELL,XC,YC,ZC,V)
C-------------------------------------------------------
C INTERPOLATION ROUTINE
C-------------------------------------------------------
      DIMENSION CELL(1)
      COMMON DELX,DELY,DELZ,D3,SIGMX
      PHI1 = (DELX-XC)*(DELY-YC)*(DELZ-ZC)/D3
      PHI2 = XC*(DELY-YC)*(DELZ-ZC)/D3
      PHI3 = XC*YC*(DELZ-ZC)/D3
      PHI4 = (DELX-XC)*YC*(DELZ-ZC)/D3
      PHI5 = (DELX-XC)*(DELY-YC)*ZC/D3
      PHI6 = XC*(DELY-YC)*ZC/D3
      PHI7 = XC*YC*ZC/D3
      PHI8 = (DELX-XC)*YC*ZC/D3
      V = PHI1*CELL(1) + PHI2*CELL(2) + PHI3*CELL(3) + PHI4*CELL(4)
     *  + PHI5*CELL(5) + PHI6*CELL(6) + PHI7*CELL(7) + PHI8*CELL(8)
      RETURN
      END
      END$
```