THE UNIVERSITY OF ALBERTA

AERIAL SPRAYING: A Computer Simulation of Spray Movement in the Aircraft Wake

by

CRAIG STANLEY MERKL

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF **MASTER OF SCIENCE**

DEPARTMENT OF AGRICULTURAL ENGINEERING

EDMONTON, ALBERTA

SPRING OF 1989

ISBN 0-315-52985-7

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR **CRAIG STANLEY MERKL**

TITLE OF THESIS **AERIAL SPRAYING: A Computer Simulation of Spray Movement in the Aircraft Wake**

DEGREE FOR WHICH THESIS WAS PRESENTED **MASTER OF SCIENCE**

YEAR THIS DEGREE GRANTED **SPRING OF 1989**

(SIGNED) Craig Merkl

PERMANENT ADDRESS:

Box 396

Brooks, Alberta

T0J 0J0

DATED 03 April 1989

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled **AERIAL SPRAYING: A Computer Simulation of Spray Movement in the Aircraft Wake** submitted by **CRAIG STANLEY MERKL** in partial fulfilment of the requirements for the degree of **MASTER OF SCIENCE.**

....
Supervisor

....

....

Date....03APR89....................

Abstract

A computer program was developed in order to simulate spray droplet movement in the wake of an aircraft. The wake was modelled with two trailing vortices, a bound vortex, and a vortex to simulate the propeller swirl. Image vortices were included to satisfy the boundary conditions for a uniformly sloping ground plane. Crosswind effects were also included. Rigid spherical droplets were modelled, with effects of evaporation included. A Lagrangian method is used to track the path of individual spray droplets in order to obtain deposition patterns and to calculate uniformity of deposition for various distances between passes of an aircraft.

Next, the computer model was verified using actual flight test data. The use of such a simple model proved to yield a high degree of correlation with actual flight test data. The process of verification included the determination of appropriate values for the vortex core coefficient, the propeller swirl coefficient, the circulation distribution, and the initial separation of the assumed distinct trailing vortices.

Once the model was verified, application included parametric studies of some variables for a typical spray operation, such as aircraft height, weight, velocity, nozzle

placement, drop diameter, wet bulb depression and crosswind. The model provides significant flexibility in modelling the spray system, including various nozzle types, arrangements, orientations, and locations. By appropriate data input this flexibility allows modelling of any aircraft, spray system, and operating conditions.

Acknowledgements

The author would like to express his appreciation to Dr. D.J. Marsden for the training which he provided, as well as his guidance and patience throughout this research. Also to be thanked is Dr. J.J. Leonard for his guidance and continued patience during this work.

I would also like to take this opportunity to acknowledge and thank the Natural Sciences and Engineering Research Council and the Department of Agricultural Engineering for the financial assistance during the research.

Table of Contents

List of Tables

List of Figures

Figure		Page

Figure | Page

Figure | Page

List of Symbols

A	cross sectional area of drop, m²
a	total acceleration of drop, m/s²
a_x	x acceleration of drop, m/s²
a_y	y acceleration of drop, m/s²
a_z	z acceleration of drop, m/s²
A_R	aspect ratio of wing
b	wing span, m
b'_o	trailing vortex initial separation, m
b'	trailing vortex separation, m
C_D	drag coefficient of droplet
C_{DS}	drag coefficient of drop for Stoke's flow
C_L	lift coefficient
D	diameter of water droplet, m
D_o	initial diameter of water droplet, m
D_{Prop}	diameter of propeller, m
d_{rup}	distance for vortex to roll up, m
F	force on drop due to rate of change of momentum, N
F_D	resultant drag force on droplet, N
F_{Ds}	Stoke's drag on droplet, N
F_{Dx}	drag force on droplet in x direction, N
F_{Dy}	drag force on droplet in y direction, N
F_{Dz}	drag force on droplet in z direction, N
g	acceleration of gravity, 9.80665 m/s²
h	height of aircraft wing trailing edge, m
h_{mxwnd}	height at which crosswind is measured, m
h_{phys}	physical roughness height of surface, m

h_{surf}	apparent height of surface ($\frac{1}{30}\, h_{phys}$), m
l	lift of incremental spanwise section of wing, N/m
m	mass of drop, kg
n	number of collector locations used
P_n	gage pressure on nozzle, kPa
Pr	Prandtl number
r	radius from center of vortex, m
r_{model}	model vortex core radius, m
r'	radius of viscous core of vortex, m
Re	Reynolds number
t	time, s
t^*	vortex flow time, $t\Gamma_o/2\pi b'^2$
t_{rup}	time for vortex roll up, s
ΔT	wet bulb depression, °C
V_o	aircraft velocity, m/s
V_{dec}	velocity of descent of pair of line vortices in unconfined flow, m/s
V_{rel}	relative velocity of droplet w.r.t. air, m/s
V_s	initial velocity of spray emitted from nozzle, m/s
V_{tan}	tangential velocity in vortex, m/s
V_{mxwnd}	measured crosswind at height h_{mxwnd}, m/s
V_{xwnd}	crosswind at height, m/s
V_{xxwnd}	x component of crosswind at height, m/s
V_{zxwnd}	z component of crosswind at height, m/s
V_x	x velocity of droplet, m/s
V_y	y velocity of droplet, m/s
V_z	z velocity of droplet, m/s

V_{xind}	x induced velocity, m/s
V_{yind}	y induced velocity, m/s
V_{zind}	z induced velocity, m/s
V_{xrel}	x relative velocity of droplet, m/s
V_{yrel}	y relative velocity of droplet, m/s
V_{zrel}	z relative velocity of droplet, m/s
x	x coordinate of drop position, m
$\bar{X}$	arithmetic mean of readings
X_i	reading for one collector location for the combined swaths
x_m	x coordinate of vortex core m, m
x_n	x coordinate of vortex core n, m
y	y coordinate of drop position, m
y_m	y coordinate of vortex core m, m
y_n	y coordinate of vortex core n, m
z	z coordinate of drop position, m
z_m	z coordinate of vortex core m, m
z_n	z coordinate of vortex core n, m

GREEK SYMBOLS

α	angle of attack, degrees
β'	propeller swirl coefficient
β_m	angle to end of vortex m, degrees
η	coefficient of variation
γ	dihedral angle of each wing, degrees
Γ	circulation of vortex, m^2/s
Γ_o	total circulation of vortex, m^2/s
Γ_o^{ell}	circulation for elliptical loading, m^2/s

Γ_o^{rect}	circulation for rectangular loading, m^2/s
μ	absolute viscosity of air, kg/m s
ν	efficiency of atomizing process
ω	propeller rotational velocity, rad/s
ω'	slipstream rotational velocity, rad/s
ρ_a	density of air, kg/m^3
ρ_s	density of spray, kg/m^3
σ	standard deviation
τ	drop life, s

1. INTRODUCTION

The subject of this work pertains to the computer simulation of spray movement in the wake of an aircraft. The main focus of the work was the development of a computer model of the aircraft wake, simplified so as to avoid the need for a supercomputer and yet which would adequately predict the movement of spray in the wake. Such a model would be a useful tool for the evaluation of any particular spray aircraft and nozzle configuration under specified operating conditions. The basis for such an evaluation is the uniformity of spray deposition, considering the combined spray deposition from adjacent, overlapping passes of the spray aircraft. A computer model would facilitate the study of numerous configurations for a spray aircraft, while the actual flight testing of those configurations would be expensive to perform adequately.

The topic involves two diverse topics, those issues concerning droplets and those concerning the aircraft wake. The two will be examined separately, then merged for the development of the computer model.

The droplet problem itself involves several areas. The drag coefficient of the droplet and the force of drag on the droplet are the main concern when dealing with the droplet movement, and will be addressed first. The effect that evaporation has on the drag of a droplet will be dealt with next. Continuing from there, the topic of evaporation from a droplet, which is a subject unto itself, will finish off the chapter dealing with droplets.

Since the spray movement takes place in a flow field influenced mainly by the aircraft wake, this is the next main focus. However, since spraying generally takes place within close proximity to the ground plane, the implications of this are considered. The most significant component of the aircraft wake is a trailing vortex system. Hence, the aircraft wake description includes that of the simplified vortex system, along with the vortex core models. The method used for modeling the flow field induced by the propeller is included as well. The effect of a crosswind is considered in this chapter since the crosswind may make a significant contribution to the spray movement. Many of the terms used are standard aerodynamics terms, however, the reader unfamiliar with the subject of aerodynamics may find a reference, such as Houghton and Carruthers (1982), valuable for developing a better understanding of the aircraft wake.

Once the droplet theory and aircraft wake have been discussed, a description of the computer model is presented. The description focuses on the areas which are not covered in the theory of droplets and the wake, or are particular instances of the theory.

The main computer model developed was the program, WAKE77 (see Appendix I). This model can be used to examine any particular aircraft and nozzle configuration, operating under specified conditions, to determine the resulting spray deposition pattern and the uniformity of that deposition. This is accomplished by performing a numerical integration of spray droplet motion, starting at various locations along the wing span, in order to determine the resulting distribution. Also, a second program was written, which was for the main part a copy of WAKE77. This second program, NASA77 (see Appendix II), was configured specifically to facilitate analysis of flight test data. NASA77 was used for verification of the model design, and the numerical value of several parameters in this study. Both programs, along with instructions for use, are included in the appendices for further reference.

Once the model development was complete, verification was necessary. The chapter on model testing and verification of parameters deals with this. As mentioned, the program NASA77 was used for this purpose. The method of testing the computer model is discussed, along with the flight test data which was used for this purpose. Then, a parametric analysis of the model was considered to verify or determine the values for the necessary parameters. Each of the parameters which were uncertain in the model development were varied so as to determine the most appropriate values to be used for further studies.

Finally, application of the computer program, WAKE77, is discussed. The applications which are discussed are not exhaustive, but serve as examples of the types of information which can be obtained by reasonable simulation of aerial spraying. The applications provide deposition patterns and the measure of uniformity of spray deposition for varied distances between passes of the spray aircraft.

2. DROPLET THEORY

2.1 Drag Theory As Applied To Droplets

The theory for drag on droplets has been quite well developed in the past few decades. This has been the result of research for the application of sprays using nozzles, and as a result of the desire to improve understanding of the formation and development of clouds in the atmosphere.

For the purposes of this study certain assumptions are made about the droplets themselves. The majority stem from the fact that most droplets of concern in aerial spraying are less than 1000 microns in diameter. Others are more related to the movement of the mass of spray droplets upon emission from a nozzle.

2.1.1 Forces acting on a droplet

There are two forces on the droplet which predominate: gravity and drag. The forces on droplets due to static pressure gradients can be neglected for aerial spraying since, in general, the small diameter of the drops results in negligible static pressure differences across the droplet. Figure 2.1 shows the forces acting on a spray droplet along with the cartesian reference frame, the axes of which are coincident with those for the computer model described later. The resultant drag force, F_D, on the droplet is

$$F_D = \tfrac{1}{2} \rho_a V_{rel}^2 A C_D \qquad [2.1]$$

where V_{rel} is the resultant relative velocity of the droplet with respect to the air, ρ_a is the air density, A is the cross-sectional area of the droplet, and C_D is the droplet drag coefficient. This can be split into the component forces

$$F_{Dx} = \tfrac{1}{2} \rho_a V_{xrel}^2 A C_D \qquad [2.2]$$

$$F_{Dy} = \tfrac{1}{2} \rho_a V_{yrel}^2 A C_D \qquad [2.3]$$

$$F_{Dz} = \tfrac{1}{2} \rho_a V_{zrel}^2 A C_D \qquad [2.4]$$

where V_{xrel}, V_{yrel}, and V_{zrel} are the component velocities of the droplet relative to the air. As well, it should be noted that C_D is a function of the Reynolds number, Re, in which the velocity used is the resultant relative velocity of the droplet with respect to the air, V_{rel}.

$$C_D = f(Re) \qquad [2.5]$$

When gravity is considered, the resultant force on a drop of mass m may be broken into its three component forces, giving the components of the acceleration of the drop as follows.

$$a_x = \frac{F_{Dx}}{m} \qquad [2.6]$$

$$a_y = \frac{F_{Dy}}{m} \qquad [2.7]$$

$$a_z = \frac{F_{Dz} - mg}{m} \qquad [2.8]$$

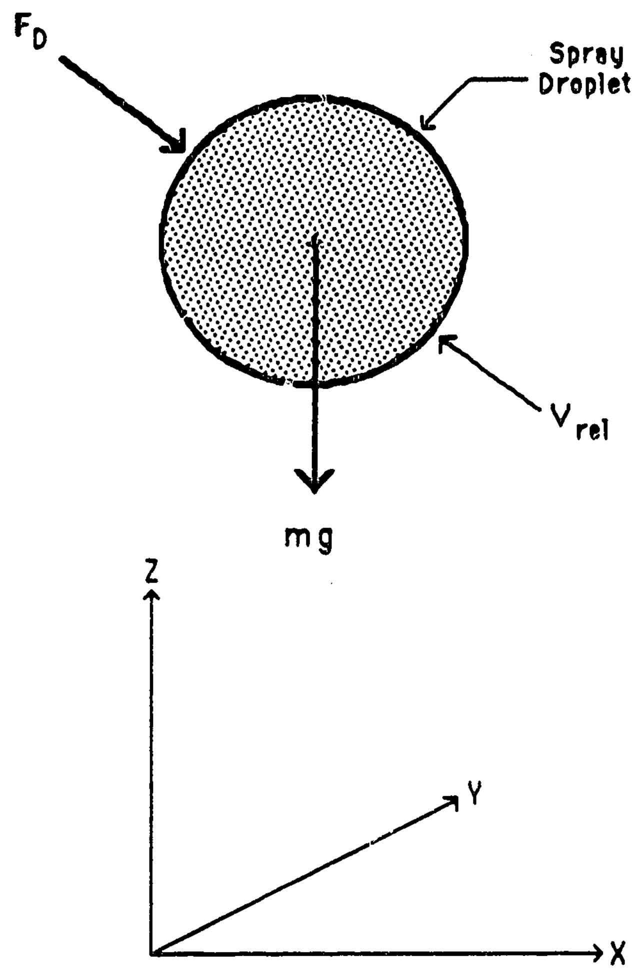


Figure 2.1 Forces Acting on a Droplet

2.1.2 Spherical drop shape assumption

The droplets which are being considered are generally less than 1000 microns and are assumed to remain perfectly spherical in shape and, consequently, behave as rigid spheres. In agreement with this, the assumption also is made that the fluid in the drop itself undergoes no significant circulation. This would be important with larger droplet sizes, as the motion of the fluid adjacent to the surface could alter the air movement around the droplet and hence the drag of the droplet. Very closely linked to the fluid circulation is the assumption that the droplets do not oscillate between oblate and prolate spheroids, as would be the case of larger droplets close to the point of release from the nozzle. According to Hughes and Gilliland (1952), this assumption is realistic for drops less than 1000 microns diameter.

2.1.3 Air entrainment with spray

Next, considering the spray emission from a nozzle of almost any type, there can be observed a certain amount of entrainment of the air mass with the movement of the spray itself. This entrainment could cause significantly lower drag effects on the droplet near the point of emission, where the spray jet is still relatively concentrated. Once the jet has proceeded further away from the nozzle there will be enough spreading of the spray jet to no longer cause any significant entrainment effects. Since this phenomena would be difficult to model accurately, and may be of negligible importance in this particular application, the assumption has been made that the entrainment effects are not significant at any point during droplet motion.

2.2 Drag Coefficients Of Droplets

The topic of drag coefficients of drops has been the subject of a significant amount of research. Some of the first major work was done by Eisner (1930). Due to the high quality of his work, his work has thereafter been referred to as the benchmark with which to compare most other studies. The shortfall of his work is that the results were purely graphical in form. Hence, much work has been done to verify Eisner's work and to develop mathematical formulations for the drag coefficients for given intervals of Reynolds numbers.

Perhaps the most detailed and carefully done work has been that of Beard and Pruppacher (1969). They used a "well designed and controlled wind tunnel" (Mason, 1971) to study droplet movement for the purposes of cloud studies. Their formulation and development resulted in equations, determined by the least squares method at the 95 percent confidence level (correlation coefficient = 0.95), as follows:

$$\frac{F_D}{F_{DS}} = \frac{C_D}{C_{DS}} \quad [2.9]$$

$$= \frac{C_D\, Re}{24} \quad [2.10]$$

$$= 1 + 0.010\, Re^{0.955}\ ,\ 0.2 \le Re \le 2.0 \quad [2.11]$$

$$= 1 + 0.115\, Re^{0.802}\ ,\ 2.0 \le Re \le 21 \quad [2.12]$$

$$= 1 + 0.189\, Re^{0.632}\ ,\ 21 \le Re \le 400 \quad [2.13]$$

where

F_{DS} = Stoke's drag, N

C_{DS} = Stoke's drag coefficient

$$C_{DS} = \frac{24}{Re} \quad [2.14]$$

In their work, Beard and Pruppacher (1969) showed that droplets do, in fact, remain spherical to $Re \le 200$, and to a very good approximation also to $Re \le 400$. This confirmed the assumption of spherical droplets which is made for drop modeling purposes.

For Reynolds numbers less than 50000, a formula developed by Langmuir and Blodgett (1946) commonly has been accepted,

$$\frac{F_D}{F_{DS}} = 1 + 0.197\, Re^{0.63} + 0.00026\, Re^{1.38}\ ,\quad 0 \le Re \le 50000 \quad [2.15]$$

Again, the droplets are assumed to be spherical and rigid even though many of the larger ones, for this range of Reynolds numbers, will actually be oblate spheroids according to Marshall (1954), and Hughes and Gilliland

(1952).

Actually, there is some difference between the values of the drag coefficient as obtained by Langmuir and Blodgett (1946), and by Beard and Pruppacher (1969) in the interval for the Reynolds number up to 400. The more recent studies by Beard and Pruppacher (1969) apparently give more accurate formulas relating drag coefficients to Reynolds numbers for Reynolds numbers up to 400. Their work dealing with terminal velocities and Reynolds numbers compares more favorably with the Eisner curve across the range of values of concern than that of Langmuir and Blodgett (1946). However, Langmuir and Blodgett (1946) presented the formula for higher Reynolds numbers as well, hence the formula of Langmuir and Blodgett is used for $Re \geq 400$. In the less certain region, $200 \leq Re \leq 400$, a linear interpolation between the two equations has been used in the computer model to avoid any discontinuity in the equation representing the drag coefficient in that interval.

For Reynolds numbers greater than 50000 the value

$$C_D = 0.500 \qquad [2.16]$$

is used, since an examination of the Eisner curve (Figure 2.2) readily reveals that the drag coefficient does not change much from this value for Reynolds numbers up to about 2×10^5. At higher Reynolds numbers the departure is greater, but Reynolds numbers in this range will be of little concern for the topic at hand.

Figure 2.2 shows the plot of the Eisner curve as well as the relation for C_D versus *Re* used in the model. The correlation can be seen to be very good in the range of Reynolds numbers of concern in aerial spraying. The departure at the higher Reynolds numbers is of little concern.

2.2.1 Acceleration affects on drag coefficients of drops

Lapple and Shepherd (1940) were amongst the first to estimate drop movement and the time for a drop to decelerate from high initial velocity (as from a spray nozzle for spray separation processes) to terminal velocity considering the instantaneous drag coefficient. However, they made no effort to explain the effects of deceleration on the drag coefficient.

As noted by Hughes and Gilliland (1952), the effects of acceleration on particles are most apparent at low Reynolds numbers. Although probably not a major concern here, it is worthwhile to note the rationale since there still may be some effect. As a spray particle accelerates (decelerates) the flow pattern around the drop must vary continuously. The Eisner curve, the accepted norm, gives C_D vs *Re* for steady state flow conditions. The flow around the drop at any point

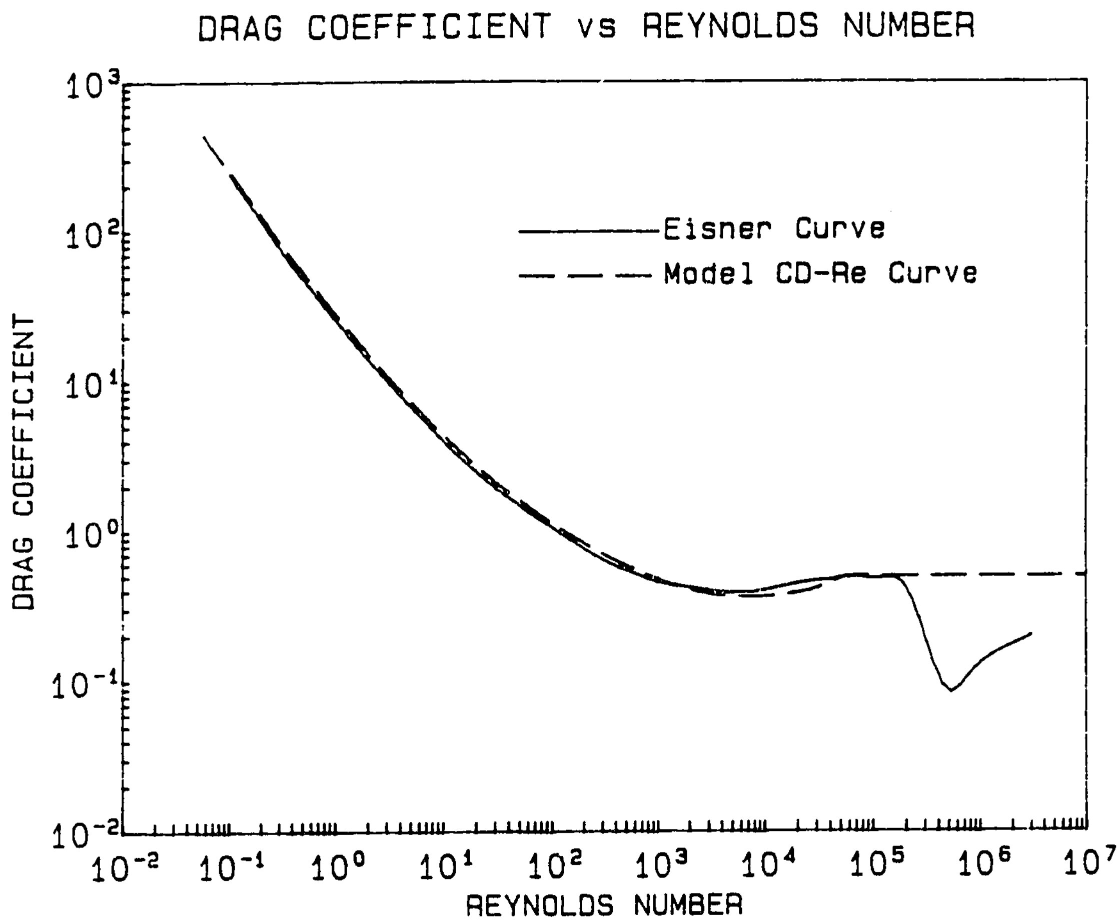


Figure 2.2 Eisner Curve and Model C_D vs *Re*

in time, especially as the drop is decelerating when first emitted from a nozzle, takes time to change. Hence, the flow is never actually in the steady state mode represented by the Eisner curve. Thus, the flow regime is always lagging behind what the steady state flow would be at the same instantaneous *Re*. As the drag coefficient increases with decreasing *Re*, it is apparent then that the drag coefficient at any time will be lower than that predicted by the instantaneous *Re*.

2.3 Apparent Acceleration Due To Evaporation

The concept of apparent acceleration due to evaporation from droplets is considered next. Considerations of the droplet momentum are used to show the apparent acceleration of the drop due to evaporation. The rate of change of momentum of the drop is as follows:

$$\frac{d}{dt}(m\ V_{rel}) = V_{rel}\frac{dm}{dt} + m\frac{d}{dt}V_{rel} \qquad [2.17]$$

where m is the droplet mass, V_{rel} is the velocity of the evaporating mass dm relative to the unevaporated droplet mass, and $\frac{d}{dt}$ is the first derivative with respect to time. Here the assumption is made that the evaporating mass of spray attains the velocity of the air. Using L'Ambert's principle of reversed forces, the force on the droplet F is equal to the rate of change of momentum.

$$F = m\ a \qquad [2.18]$$

$$= V_{rel}\frac{dm}{dt} + m\frac{d}{dt}V_{rel} \qquad [2.19]$$

The term $m \frac{d}{dt} V_{rel}$ is simply the acceleration of the drop related to a change in velocity, and is independent of evaporation. Since the term $V_{rel} \frac{dm}{dt}$ accounts for the effects of evaporation, this term should be considered to determine its significance. Also, for evaporation considerations $\frac{dm}{dt}$ will be a negative term, since the drop mass is decreasing. Couple this with the fact that V_{rel} is negative (where the direction of the drop velocity is considered positive), with respect to the force, and it is seen that evaporation provides a component of acceleration of the drop in the direction of the drop velocity.

The purpose of the following section is to show that it is reasonable to neglect the apparent acceleration effects for conditions in aerial spraying, since the forces on the droplet due to the drag of the drop in the air can be shown to be much greater. The equation for the drag on the droplet again is as follows:

$$F_D = \tfrac{1}{2} \rho_a V_{rel}^2 A C_D \qquad [2.1]$$

Now, with

$$Re = \frac{\rho_a V_{rel} D_o}{\mu} \qquad [2.20]$$

and

$$A = \frac{\pi D_o^2}{4}$$

where D_o is the initial drop diameter, one can derive the following:

$$F_D = \frac{\pi C_D \mu^2 Re^2}{8 \rho_a} \qquad [2.21]$$

Further, with $C_D = f(Re)$

$$F_D = \frac{\pi \mu^2 Re^2 f(Re)}{8 \rho_a}$$

and, with ρ_a = 1.2256 kg/m³ and μ = 1.78 × 10^{-5} kg/m s

$$F_D = 1.0152 \times 10^{-10} Re^2 f(Re) \qquad [2.22]$$

Thus, F_D is related only to Re.

Now, the apparent force on the drop is considered. The value of $\frac{dm}{dt}$ can based on the "half-life" of the drop as follows. The "half-life" is such time that the drop reduces to one half of its original diameter in the time $dt_{1/2}$. Given that the initial drop diameter is D_o, then the change in mass of the drop dm in its "half-life" is,

$$dm = \left(\rho_s \frac{\pi D_o^3}{6} \right) - \left(\rho_s \frac{\pi (D_o/2)^3}{6} \right)$$

$$= \frac{7}{8} \left(\rho_s \frac{\pi D_o^3}{6} \right)$$

Hence,

$$\frac{dm}{dt} = \frac{7 \pi \rho_s D_o^3}{48 dt_{1/2}}$$

Now, with

$$V_{rel} = \frac{Re \mu}{\rho_a D_o}$$

One gets

$$V_{rel} \frac{dm}{dt} = \left(\frac{Re \mu}{\rho_a D_o} \right) \left(\frac{7 \pi \rho_s D_o^3}{48 dt_{1/2}} \right)$$

which simplifies to

$$V_{rel} \frac{dm}{dt} = \left(\frac{Re D_o^2}{dt_{1/2}} \right) \left(\frac{7 \pi \mu \rho_s}{48 \rho_a} \right) \qquad [2.23]$$

Again, using ρ_a = 1.2256 kg/m³, μ = 1.78 × 10^{-5} kg/m s , and ρ_s = 1000 kg/m³.

$$V_{rel} \frac{dm}{dt} = 6.6539 \times 10^{-3} \left(\frac{Re D_o^2}{dt_{1/2}} \right) \qquad [2.24]$$

Now, dividing equation 2.22 by 2.24, the resulting expression is,

$$\frac{F_D}{V_{rel}\,\frac{dm}{dt}} = 1.5257 \times 10^{-8} \left(\frac{Re\, f(Re)\, dt_{1/2}}{D_o^2} \right) \qquad [2.25]$$

To illustrate the magnitude of the ratio, several values of D_o along with the Re and C_D corresponding to the terminal velocities, are included in Table 2.1. The "half-life" for each of these water drops, $dt_{1/2}$, was determined using the program, DROPTIME, given in Appendix IV. The program, DROPTIME, is based on the same formulations as those included in WAKE77, and can be used to determine, at any time, the evaporated size of a droplet moving at terminal velocity, for a specified wet bulb depression, ΔT.

As can be seen, the value of the ratio F_D to $V_{rel}\,\frac{dm}{dt}$ based on the "half-life" increases slightly for larger diameter droplets. Most importantly, the term $V_{rel}\,\frac{dm}{dt}$ can be shown to account for only about 0.3 percent of the total acceleration of the drop for a wet bulb depression of 10°C. For a wet bulb depression of 15°C the value is still only about 0.5 percent of the total acceleration. Therefore in the computer model this term, which represents the apparent acceleration due to evaporation, can be neglected justifiably.

Table 2.1 $F_D / (V_{rel} \frac{dm}{dt})$ for ΔT = 10°C and 15°C

			ΔT = 10°C		ΔT = 15°C	
D_o (µm)	C_D	Re	$dt_{1/2}$ (sec)	$\frac{F_D}{V_{rel} \frac{dm}{dt}}$	$dt_{1/2}$ (sec)	$\frac{F_D}{V_{rel} \frac{dm}{dt}}$
50	96.094	0.257	1.94	292.7	1.29	194.6
100	15.779	1.791	6.50	280.3	4.33	186.7
200	4.199	9.819	19.17	301.5	12.78	201.0
300	2.424	23.746	34.39	335.6	22.93	223.7
400	1.672	44.008	50.97	357.6	33.81	237.3
500	1.293	69.937	67.90	374.7	45.27	249.8
1000	0.691	270.441	162.88	464.4	108.43	309.2

2.4 Evaporation From Droplets

2.4.1 Drop life

The concern of this section is that of the prediction of the life of a droplet of pure water under given conditions of evaporation. The assumption of pure water is reasonable given the conditions as delineated in the following section dealing with the effects of oil concentrations in the droplet. Also, the drop life predictions are based on drops under free fall conditions at terminal velocity. This is reasonable since droplets in the size range of concern will be moving at only slightly greater velocities behind the aircraft than they would at terminal velocity in free fall conditions (Ormsbee and Bragg, 1978).

Trayford and Welch (1977) have given a method to calculate the drop life and the diameter of a droplet at any time after formation.

$$\tau = \frac{D_o^2}{\beta \ \Delta T} \qquad [2.26]$$

where,

τ = droplet life, s

β = 84.76 (1.0 + 0.3 $Pr^{1/3}$ $Re^{1/2}$) × 10^{-12}

ΔT = wet bulb depression, °C

D_o = initial diameter, m

Pr = Prandtl number = 0.72

Re = Reynolds number

The Prandtl number has been shown to remain nearly constant at 0.72 for air over the temperature range 250°K to 1000°K (Ranz and Marshall, 1952a and 1952b).

Here, the Reynolds number is that for the terminal velocity of the droplet at the initial diameter and density in a free fall state. Hence, this drop life is that for a drop which is continuing to fall at the terminal velocity corresponding to its instantaneous diameter.

Manning and Gauvin (1960) considered the problem of heat and mass transfer from decelerating, finely atomized sprays. They concluded that, if the feedwater temperature is not too different from the wet bulb temperature, the droplet will reach the wet bulb temperature within approximately 13 mm from the nozzle. Hence, it is reasonable to use the wet bulb depression of the air to give a measure of the mass transfer from a spray droplet. If the drop remained at a temperature different from the wet bulb temperature of the air, the droplife could not be based on the wet bulb depression.

Next, knowing the drop life, the drop diameter at any time after formation can be determined (Trayford and Welch, 1977).

$$\frac{D}{D_0} = \left(1 - \frac{t}{\tau}\right)^{1/2} \qquad [2.27]$$

where

t = time since formation of drop, s

D = diameter of drop at time t, m

Walton and Walker (1970) have given a treatment of the drop diameter at any time, similar to that of Trayford and Welch (1977) but in a slightly different form. These can be shown to be equivalent (see Appendix V).

For purposes of the computer model, the evaporation of water from spray droplets stops at the point when $\frac{D}{D_o} = 0.5$, leaving only 1/8 the initial volume. As the concentration of the nonevaporating component of the drop changes, the rate of evaporation will decrease and, hence, this should be a reasonable assumption. As Figure 2.3 shows, the time to reach $\frac{D}{D_o} = 0.5$ for most water drops is sufficiently large so as to be of no concern in aerial spraying. Results from NASA77 showed that, for normal spray conditions conducted at an aircraft height of near one half wingspan, most drops with D_o near to, or greater than, about 200 μm would hit the ground within approximately 2 seconds.

2.4.2 Acceleration effects on drop life

The model used to determine the terminal velocity, and, hence, the drop life, τ, produced terminal velocities and Reynolds numbers slightly higher than that given by Mason

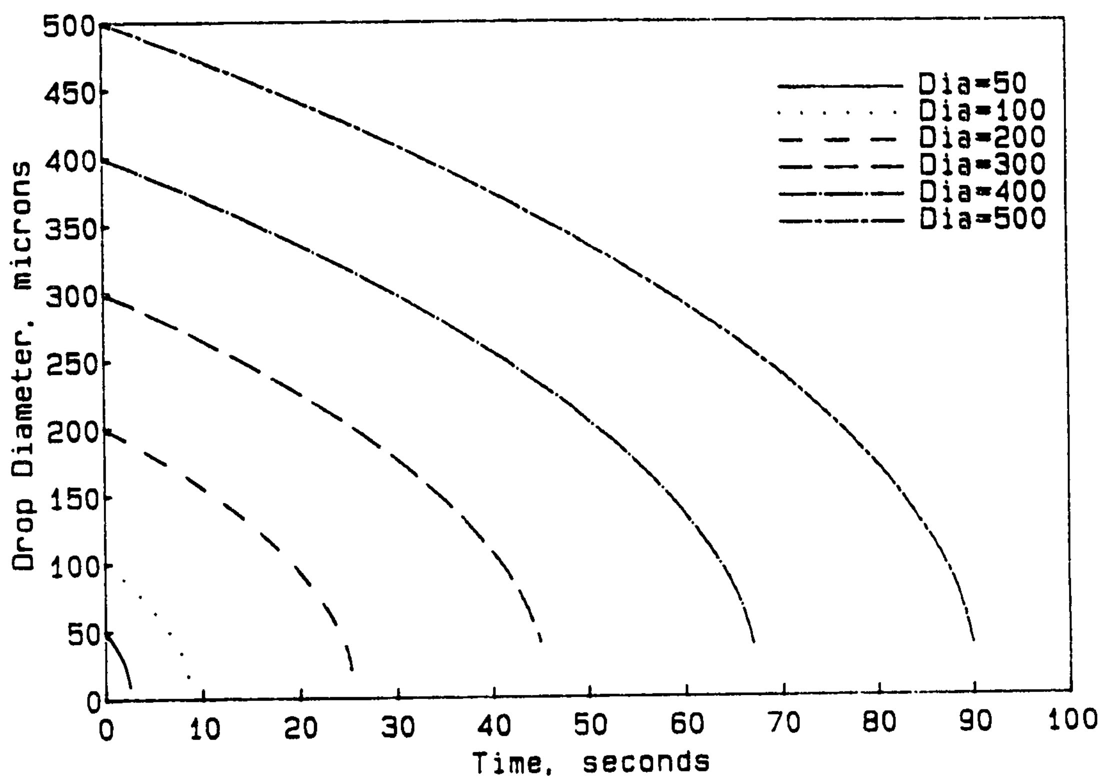


Figure 2.3 DROP DIAMETER versus TIME at ΔT = 10°C

(1971). For example, for a drop diameter of 100 microns at the terminal velocity Mason gave Re = 1.69, while the computer model gave Re = 1.79. As a result there appears to be a slightly shorter drop life due to the higher terminal velocity. For a 100 micron droplet and wet bulb depression of 10°C, Trayford and Welch (1977) gave $\tau \approx 9.6$ seconds. The present computer model gave, for the same conditions, $\tau \approx$ 8.66 seconds. For a 50 micron drop, Trayford and Welch gave $\tau \approx 2.6$ seconds, while the model gave $\tau \approx 2.6$ seconds.

Possibly, the deceleration of the drop contributes to the discrepency. Hughes and Gilliland (1952) have shown how acceleration affects can be important. Acceleration causes there to be an increase in the drag coefficient C_D, and deceleration causes a decrease in C_D as compared to the C_D based on the instantaneous velocity in steady state conditions.

For the case of spray movement, the drop decelerates from the initial velocity. The instantaneous drag coefficient will always be somewhat lower than that corresponding to steady state conditions. Thus, the model predicts a terminal velocity for the decelerating drop that is slightly higher than the real case. This will result in slightly greater evaporation rates and, hence, reduced drop life.

2.4.3 Effects of oil content on evaporation rates

A considerable uncertainty in the evaporation rate of a water based spray drop may develop due to the content of oil or other such nonevaporating liquids, henceforth referred to as "oil", in a spray. The problem exists because of the various possible distributions of the oil throughout the droplet. This may range from the oil moving to the surface, being evenly distributed throughout the droplet, or being confined inside and away from the droplet surface.

The highest evaporation rates generally will occur when the surface of the droplet is entirely water, since most other spray components are less volatile than water. Then, the rate will be that given by Trayford and Welch (1977). This assumes that, during evaporation, the surface remains water, with the other materials continually concentrating near the center or, at least, away from the surface. Obviously, this leads to the point when the water is depleted, leaving a drop consisting solely of oil. For aerial spraying, as has been shown, the time frames which are relevant will normally exclude the latter possiblity.

The other extreme is that the oil content of a drop will concentrate on its surface. If this occurs quickly after droplet formation, the droplet will essentially remain at the initial size since oil has an evaporation rate of nearly zero compared to water. This may occur even if just enough of the oil, not necessarily all of it, moves to the surface and disperses so as to cover the droplet. Thus, a very thin film just several molecules deep could considerably reduce evaporation rates.

The most likely condition will, however, be such that the oil content is dispersed about the droplet in some irregular fashion. Then, depending on the precise nature of the oil distribution and shape of the droplet, varying amounts of oil and water will appear on the surface. A greater percentage covered with an oil film would result in a lower evaporation, and *vice versa*. This concept is another complete area of potential research.

Another factor to be considered is the changing concentration of different insoluble materials in the droplet as evaporation proceeds. This will cause evaporation rates to decrease as solids accumulate at the surface, restricting water movement to the surface. Hence, in the model, the simulation is stopped once the droplet reaches one half of its initial diameter.

Due to the complexity of the problem of modeling the evaporation rate of a droplet, the effects of varying evaporation rates have been ignored. In order to model a less volatile droplet at a given wet bulb depression, the wet bulb depression can be reduced according to the approximated reduction in volatility. Likewise, to model a more volatile droplet, the wet bulb depression may be increased by an appropriate amount.

3. THE AIRCRAFT WAKE

3.1 The Wake Model

The spray operation considered is such that the aircraft is operating in steady level flight with no flap deflection. Hence, as will be explained later, the wing sheds vorticity from the wingtips, which rolls up into two discrete vortices. Basically, a vortex is a line or chain of fluid particles spinning on a common axis and carrying around with them a swirl of fluid particles which flow around in circles (Houghton and Carruthers, 1982).

A lifting wing generates lift by producing circulation, Γ, such that the lift per unit span, l, is given by:

$$l = \rho_a V_o \Gamma \qquad [3.1]$$

where, ρ_a is the air density, V_o is the aircraft velocity. The total circulation generated by the wing, Γ_o, is the local value for the circulation, which is a function of spanwise location, integrated across the span, b. Circulation, Γ, is defined as the line integral of the tangential velocity component around any closed circuit in a fluid (Houghton and Carruthers, 1982). If the closed circuit is taken around the center of a vortex, then the circulation is called the strength of the vortex.

The wake model considered is a simple horseshoe vortex system as shown in Figure 3.1. The circulation contained in each trailing vortex is equal to the total circulation generated by the wing. The circulation of the wing is considered to be produced by the bound vortex, which is a vortex located at 25 percent chord, as shown in Figure 3.1. The bound vortex so located can be considered to give adequate results for the induced velocities beyond a fraction of a chord from the trailing edge of the wing (Spreiter and Sacks, 1951). The spanwise extent of the bound vortex is determined by the initial spanwise separation of the two trailing vortices. The two completely rolled up trailing vortices are assumed to originate at the spanwise location where the bound vortex ends. These trailing vortices are attached to the wing at 25 percent chord so as be continuous with the bound vortex. For the total vorticity of the generated vortex system to be zero, the starting vortex, which is that vortex generated by the wing when just beginning to produce lift at the time of takeoff, must be considered. However, the starting vortex is not deemed to be relevant since it is left behind upon takeoff, keeping its influence far removed.

3.2 The Circulation Generated by a Lifting Wing

Again, the total circulation generated by the wing, Γ_o, is the local value for the circulation integrated across the span, b. The circulation shed at any spanwise location can be determined by the spanwise rate of change of circulation of the wing ($\frac{d\Gamma}{dy}$). The total value, Γ_o, corresponds to the circulation of the wing in the plane of symmetry (Spreiter and Sacks, 1951). Figure 3.2 shows the relation for Γ versus y for an elliptically loaded wing, and for a rectangularly loaded wing, where y is the spanwise position being measured from the centerline of the aircraft. The distribution of the load on a wing is equivalent to the circulation distribution since the circulation generates the lift, which is equivalent to the load at any spanwise location. Note that an elliptical loading does not imply an elliptical planform, nor does a rectangular loading imply a rectangular planform, hence, the reader should be careful to note which is stated. Thus, when a rectangular wing is referred to, the wing is meant to be rectangular in planform, not load distribution.

In accordance with Spreiter and Sacks (1951) the value for the total circulation of an elliptically loaded wing is Γ_o^{ell}, where

$$\Gamma_o^{ell} = \frac{2\ b\ V_o\ C_L}{\pi\ A_R} \qquad [3.2]$$

where

b = wing span, m

V_o = aircraft velocity, m/s

C_L = lift coefficient

A_R = wing aspect ratio, (span/chord)

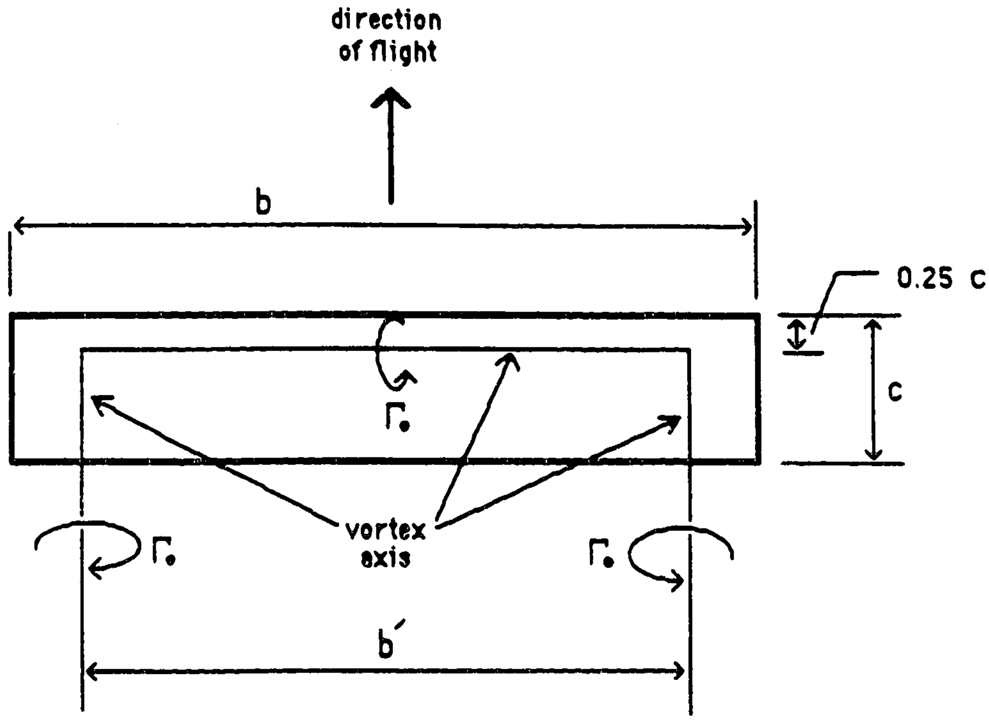


Figure 3.1 Simple Horseshoe Vortex

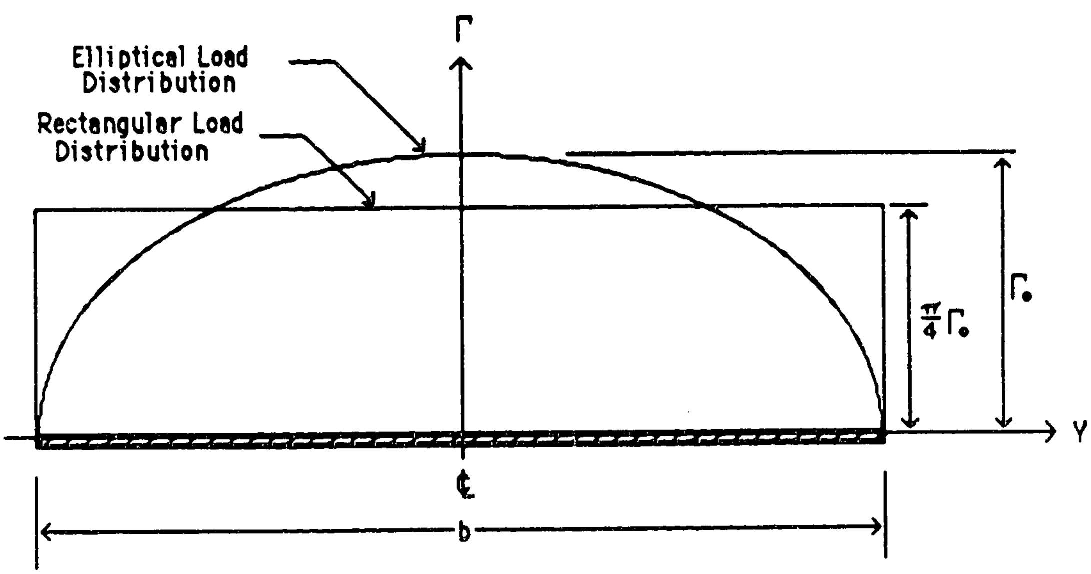


Figure 3.2 Γ versus *y* for elliptically loaded wing and for rectangularly loaded wing

In accordance with Houghton and Carruthers (1982) the total circulation for a rectangularly loaded wing is Γ_o^{rect} , where

$$\Gamma_o^{rect} = \frac{b\, V_o\, C_L}{2\, A_R} \qquad [3.3]$$

Now, dividing Eqn. 3.2 by Eqn. 3.3 gives:

$$\frac{\Gamma_o^{ell}}{\Gamma_o^{rect}} = \frac{4}{\pi} \qquad [3.4]$$

Thus, the value of Γ_o for a rectangularly loaded wing is $\pi/4$ times the value of Γ_o for an elliptically loaded wing.

The actual value of Γ_o is calculable for a finite wing with plain rectangular planform (ie. in level flight with no flap deflection), but the propeller, fuselage, and tail empennage will cause sufficient interference that the value cannot be determined, precisely, for the whole aircraft. McCormick (1954) and George (1985) showed the circulation for a rectangular wing of finite aspect ratio to be between that of rectangularly and elliptically loaded wings. For a wing which has an aspect ratio of about 5.5 (e.g. as will the aircraft used for flight tests) the value for the total circulation may be taken as the average of the values for the rectangular and elliptical loading.

3.3 Vortex Roll Up

Vorticity is first shed from the trailing edge of the wing as a vortex sheet, as proposed by Prandtl (1921). The vortex distribution shed from the trailing edge of the wing is determined largely by the planform and airfoil shape along the span (geometry of the wing) as shown by Spreiter and Sacks (1951), and others. For conditions of a smooth planform and no flap deflection, the vortex sheet rolls up, starting at the wing tips, into two discrete vortices trailing the wing. Figure 3.3 shows the scheme of roll up of the vortex sheet beginning at the wing tip.

As Betz (1933) showed, the center of vorticity, which is similar in nature to the center of gravity, will remain at a constant spanwise location during the roll up process, given that the aircraft is free of nearby boundaries. The actual pattern of roll up will be infuenced by the spanwise rate of change of loading of the wing ($\frac{d\Gamma}{dy}$). This topic is covered adequately by Spreiter and Sacks (1951), Brown (1973), Donaldson *et al*. (1974), Bilanin and Donaldson (1975), Roberts (1983), Yates (1974), and others. Yates (1974) gives an especially good discussion of vortex roll up considering the problem of multiple vortices, as would be the case from flap deflection.

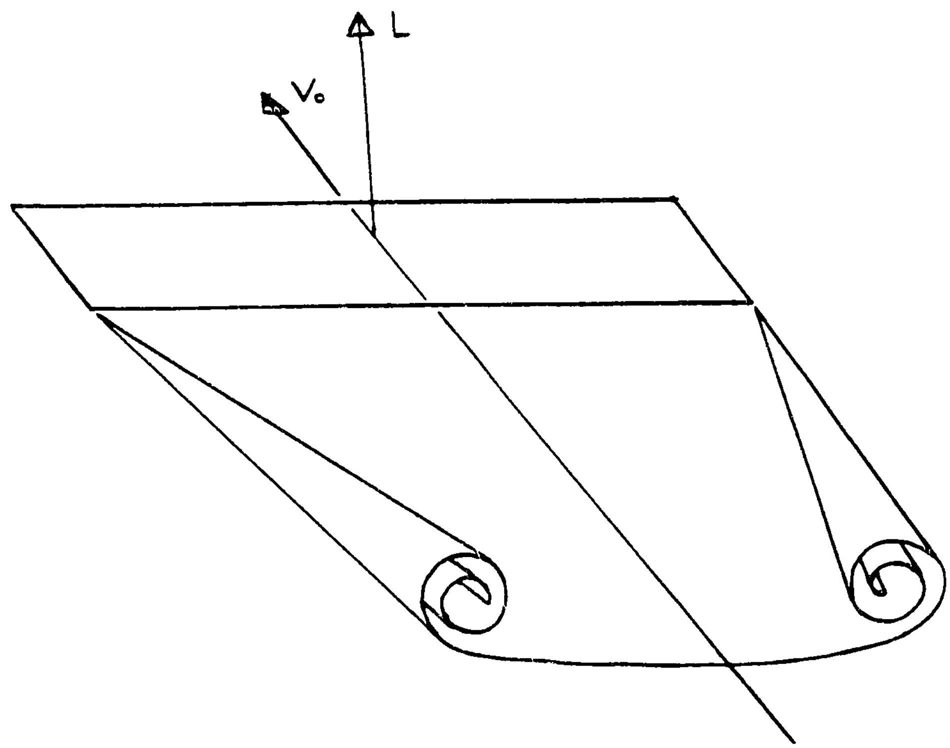


Figure 3.3 Vortex Roll Up Behind a Lifting Wing

"Roll up of a sheet asymptotically approaches the completely rolled up condition as distance to the wing approaches infinity," Spreiter and Sacks (1951). The actual time for the vortex sheet to roll up into discrete vortices incorporating essentially all of the voriticiy, has been shown by Sprieter and Sacks (1951) to be dependent only on the aspect ratio, vortex spacing, lift coefficient, and aircraft velocity. The calculations of Spreiter and Sacks (1951) were based on an ellipitical wing loading and gave the following value for time to roll up:

$$t_{rup} = \frac{0.36\ A_R\ b'}{C_L\ V_o} \qquad [3.5]$$

where

t_{rup} = time to roll up, s

A_R = aspect ratio of wing

b' = vortex spacing, m

C_L = lift coefficient

V_o = aircraft velocity, m/s

The distance to roll up, d_{rup}, then can be found:

$$d_{rup} = t_{rup}\ V_o$$

$$= \frac{0.36\ A_R\ b'}{C_L} \qquad [3.6]$$

For example, with typical values for spray aircraft A_R = 5.5, C_L = 0.6, V_o = 45 m/s and b' = 0.82 b (where the span, b = 12 m), d_{rup} is determined to be 32.5 m and t_{rup} = 0.72 seconds.

The vortices will roll up even faster when more of the load (than for elliptical loading) is towards the wing tips as with subsonic rectangular wings (Spreiter and Sacks, 1951.)

Thus, the time for vortex roll up is significant, but with regard to the average time for droplet movement it may be neglected. Since spray movement actually often takes longer than this, the error should not be significant if the model of the two fully rolled up trailing vortices is used.

3.3.1 Trailing vortex lateral spacing

McCormick (1954) showed that the separation of the trailing vortices could be calculated exactly, for aircraft of various loadings, in unconfined flow conditions, i.e. far from the ground.

For an elliptically loaded wing

$$\frac{b'}{b} = \frac{\pi}{4} \cong 0.7868 \qquad [3.7]$$

For wings of rectangular planform with $A_R = 6$

$$\frac{b'}{b} \cong 0.82 \qquad [3.8]$$

Hence, the value of 0.82 can be used to closely represent the spanwise location for the initial separation of the trailing vortices from a rectangular wing with A_R = 5.5, as applies to the aircraft used for the flight tests.

Trayford and Welch (1977) have, in fact, used discrete trailing vortices such as this. These authors used 90% span initial separation but provided no justification for choosing this particular value.

3.3.2 Induced velocity field

When the induced velocity field near the wing is considered, the vortex sheet model is necessary to give accurate results, especially for high aspect ratio aircraft. Further behind the wing, the vortex sheet may be considered to be rolled up into the two discrete vortices. Spreiter and Sacks (1951) suggest that the induced velocity field due to the horseshoe vortex system will give adequate results at distances of one chord length or more from the wing. Spreiter and Sacks have found that the vortices may be considered to be rolled up completely at a distance of about six span lengths for an aspect ratio of six, and less for lower aspect ratios.

The topic of the induced velocity field will be dealt with further in the chapter on the model assumptions.

3.4 Ground Plane Effects

In an unconfined region it should be noticed that the movement of the fully rolled up vortices would only be downwards in the absence of crosswinds. For two infinite line vortices at the same height, and of equal but opposite strength Γ, their motion is simply vertically downwards. Their velocity of descent, V_{dec}, being

$$V_{dec} = \frac{\Gamma}{2 \pi b'} \qquad [3.9]$$

where b' is the lateral separation between the vortices (Lamb, 1932).

The presence of a ground plane imposes a boundary condition such that there can be no airflow component perpendicular to the ground surface. This condition is adequately satisfied by the three image vortices as shown in Figure 3.4. The bound vortex and each of the two trailing vortices are shown above the ground plane. The ground plane acts as a reflection plane such that the image vortices appear as reflections of the real vortices.

As a result of the images, with the wing close to the ground plane, there will be considerable outward lateral movement of the vortices during the roll up process. The effect of the ground plane on the motion of the trailing vortices will be more dramatic the nearer to the ground the vortices originate or descend.

Actual trailing vortices, once rolled up, will be curved semi-infinite line vortices because they stay attached to the wing as it moves away and, at any point,

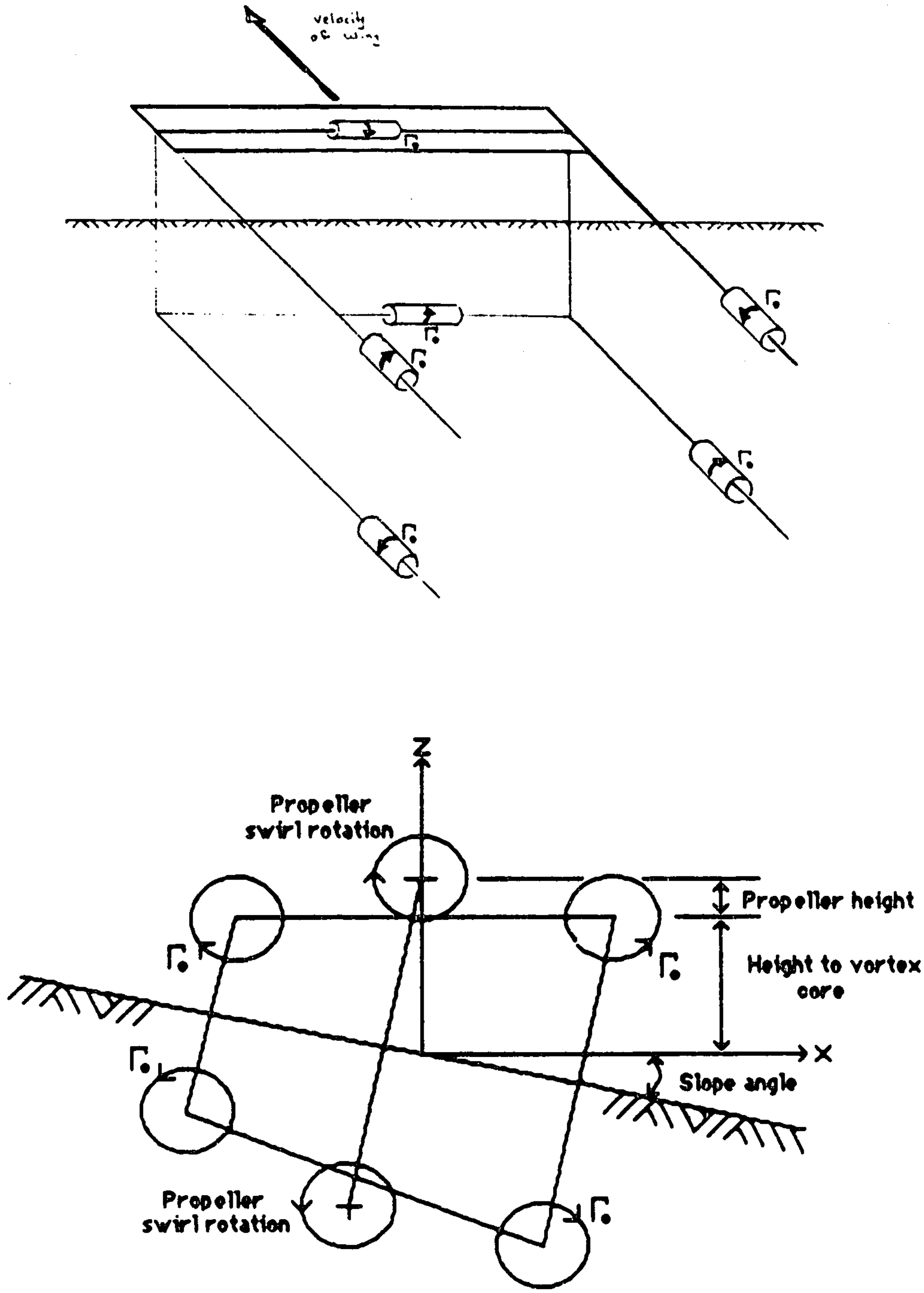


Figure 3.4 Ground Plane and Image Vortex System

they move down and outwards laterally, depending on the wind and ground plane proximity.

These effects are all modeled in the equations for vortex core movement near a uniformly sloping ground plane, given in the section on the model development.

The ground plane is a significant influence when considering both the vortex roll up and the path taken by the trailing vortices after formation. Since the spray drop will initially move under the influence of the incompletely rolled up vortex sheet, some consideration should be given to this effect.

3.5 Vortex Roll Up In Ground Effect

The influence of the ground plane will undoubtedly have an affect on the roll up process and time. This is one matter which does not seem to have received much attention in research as yet and, apparently, is not completely understood. In fact, it is likely that the vortex roll up would not proceed in the same fashion near the ground as would be the case in an unconfined region.

The simple horseshoe vortex system normally does not show the lateral placement of the trailing vortices to be dependent on the ground proximity. Hence, a scheme to modify the initial vortex spanwise location, depending on the height of the aircraft, was developed to give a more reasonable spanwise position at which to attach the trailing vortex pair. At distances far from the ground plane, the vortices will roll up such that their lateral spacing conserves the center of vorticity, as has been explained by Betz (1933).

This may not necessarily be the case with a wing which is in close proximity to a ground plane. With a rectangular wing, as is commonly found on agricultural aircraft, the load distribution is such that more vorticity is shed towards the wingtip than would be the case of an elliptically loaded wing. When the wing is closer to the ground this vorticity will not migrate inwards as in unbounded conditions. Thus, near the ground, more of the vorticity will roll up and remain at a greater spanwise location than in unbounded conditions.

3.6 Ground Effect And Secondary Vortices

The formation of a secondary vortex at the ground level has been documented by Bilanin *et al*. (1978) and Harvey and Perry (1971). The trailing vortex descends near to the ground plane after some time and, at the same time, moves laterally outwards. The reason for the formation of this secondary vortex is linked with the viscosity of air and the scrubbing effects of the trailing vortex adjacent to the ground plane. Morris (1978) has suggested that the boundary layer below the descending trailing vortex grows and separates, resulting in the secondary vortex. The secondary vortex is of opposite sign to that of the descending trailing vortex. As the secondary vortex grows it will induce both upward and outward motion on the descending vortex, hence is termed 'vortex bounce'. Any spray droplets, in the aircraft wake, laterally inwards from the secondary vortex, will be affected by the same upwards and outwards induced flow.

Harvey and Perry (1971), using a wind tunnel with a moving floor, were amongst the first to show that vortex bounce does occur, and to explain the phenomenon in terms of the separation of the boundary layer. They suggested the formation of a 'bubble' of vorticity which separates from the boundary layer flow forming the secondary vortex. Bilanin *et al*. (1978) showed isopleths of vorticity for a vortex pair descending toward a ground plane, and the vortex bounce caused by the secondary vortex, as shown in Figure 3.5. Figure 3.6 shows the effect of the secondary vortex on the motion of the one of the trailing vortices, as compared to the movement under inviscid conditions. Dee and Nicholas (1969) were the first to actually measure the occurrence of vortex bounce using actual flight tests and commented on it. However, they made no effort to explain the phenomenon.

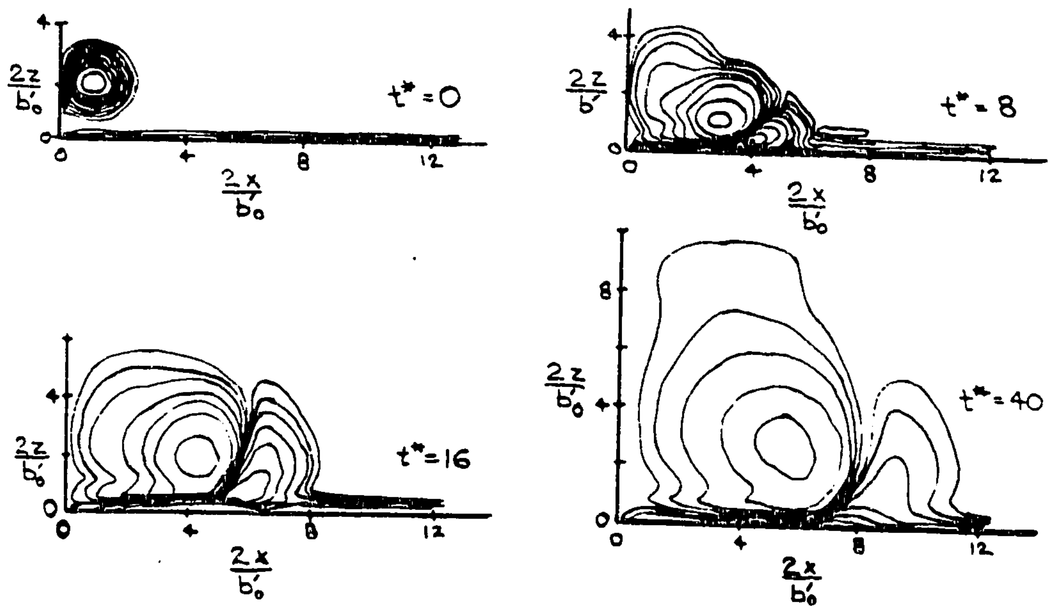


Figure 3.5 Isopleths of Vorticity for a Vortex Pair Descending near Ground Plane (adapted from Bilanin et al., 1978)

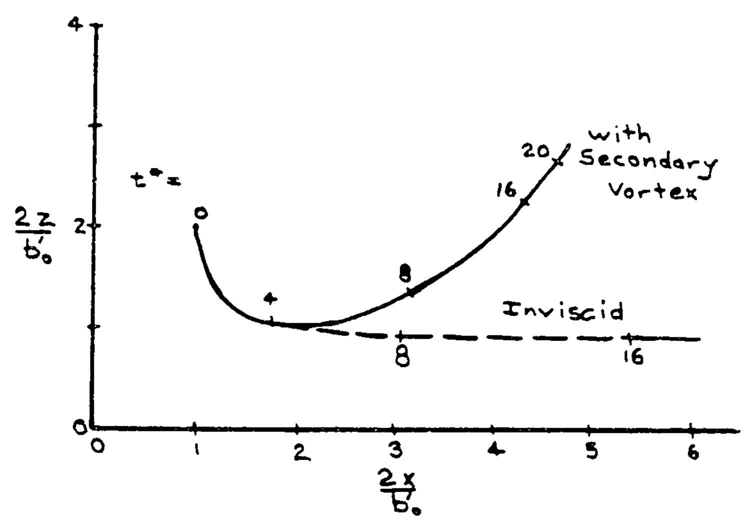


Figure 3.6 Effect of Secondary Vortex on Trailing Vortex Motion (adapted from Bilanin *et al.*, 1978)

As shown in Figure 3.5, the formation of this secondary vortex will take a finite period of time. This is indicated by the vortex flow time, t^*, a parameter which is a dimensionless measure of time in a vortex flow. Bilanin *et al.* (1978) showed that

$$t^* = \frac{t\,\Gamma_o}{2\,\pi\,b'^2} \qquad [3.10]$$

An idea of the magnitude of t which corresponds to the particular values of t^* for standard spray aircraft can be determined by substitution of the relevant parameters. Using equations 3.3 and 3.8 with $b' = 0.82\,b = 0.82\times12.625$ m, $V_o = 50$ m/s, $C_L = 0.5$, and $A_R = 5.5$, gives $\Gamma_o^{rect} \cong 28.69$ m²/s. Now, subsitituting into equation 3.10, one gets $t^* = 0.043\,t$. Hence, for the value of $t^* = 8$, when the secondary vortex is starting to develop, $t = 188$ seconds. Since most spray movement for medium to large droplets takes only a few seconds, the secondary vortex will not be sufficiently developed to have any significant influence on these droplets. For finer sprays, the secondary vortex may have some affect, however, the exact magnitude is unknown at this time. Although some researchers are convinced of the importance of the secondary vortex, the model does not include the development of the secondary vortex in any case.

3.7 Tail Plane Effects

The effects of the tail plane of a conventional aircraft, on the wake of the aircraft, are usually neglected since contribution is small compared to that of the main wing (in the order of 10 percent).

The tail produces a negative lift force as a consequence of longitudinal stability requirements. As a result of the negative lift, a horseshoe vortex system will develop, similar in nature to that of the main plane, but of opposite sign. The wing must produce an additional amount of lift to counter the negative lift of the tail. Hence, the lift of the wing can be considered to be equal to the all up weight of the aircraft, and the increased circulation of the wing will be cancelled by the circulation of the tail. The effect of the center of gravity position would be important to consider if this were not the case. Within the margin of safety, the center of gravity can move fore and aft a small amount. With a more forward center of gravity, the tail must produce greater negative lift. Hence, the wing produces greater lift, increasing the circulation about the wing. However, as noted above, this effect will be insignificant for this application.

3.8 The Propeller And Its Vortex System

A rather complex program using a helical vortex pair originating at the tips of the propeller as well as an axial vortex was developed and used by Bragg (1977). In fact, this helical vortex pair combined with the axial vortex does model the real propeller accurately, however, Bragg gave no consideration to the effect of the aircraft fuselage and tail empennage on the resultant induced velocity. The question remains as to whether such a complex model is warranted without due consideration given to the whole aircraft, or if a simplified model would suffice.

The main effect of a propeller is the axial velocity increase in the slipstream. This increase, in the velocity of the influenced air mass, is necessary to provide the momentum change of that air mass to produce thrust. However, in producing thrust with a propeller, invariably, there is a significant swirl imparted to the slipstream.

The propeller swirl coefficient is defined as

$$\beta' = \frac{\omega'}{\omega} \qquad [3.11]$$

where,

β' = propeller swirl coefficient

ω = propeller rotational velocity, rad/s

ω' = slipstream rotational velocity, rad/s

For a propeller alone in an airstream, propeller design theory can be used to determine the correct value for a swirl coefficient. However, to do this, the power output, efficiency, and the operating characteristics of the propeller must be known. The complete aircraft (the fuselage, wings, radial engine, and tail) causes significant interference, and renders even the most complex propeller design models only approximations.

As already mentioned, axial flow would occur in actual conditions, but for considerations of aerial spraying, would only displace the spray along the direction of the flight path. Hence, sufficient for this work, the propeller can be modeled as a simple line vortex with a Rankine core and no axial or radial flow.

Houghton and Carruthers (1982), in an example problem concerning propeller design, show the swirl coefficient to be about 0.015. This agrees with a value calculated by Marsden (1988). This value may be accurate for the propeller alone, but for the entire aircraft the value is reduced considerably. As will be seen later, upon examining the test results for verification of the computer model (NASA77), swirl coefficient values as high as 0.0075 significantly reduce the degree of correlation of the computer model predictions with the flight test data.

3.9 Vortex Core

The core of a vortex is determined largely from the roll up process, and hence can be shown to be significant to the induced velocities caused by the vortex.

A vortex in inviscid incompressible fluid would demonstrate a tangential velocity, V_{tan}, versus radius, r, profile as shown in Figure 3.7 , with

$$V_{tan} \propto \frac{1}{r} \qquad [3.12]$$

For a vortex with total circulation, Γ,

$$\Gamma = 2 \pi r V_{tan} \qquad [3.13]$$

or,

$$V_{tan} = \frac{\Gamma}{2 \pi r} \qquad [3.14]$$

Beyond a certain distance from the axis, a vortex in a viscous fluid will behave as a vortex in inviscid fluid. However, the inner core is significantly affected by viscosity. The vortex in inviscid fluid has a singularity as $r \to 0$, which implies an infinite tangential velocity at the very center. This obviously cannot be a realistic condition, and experimental measurements have confirmed this. The tangential velocity has been found to vary linearly with the radius for the inner region of the viscous core, then to round off to meet the curve of the potential vortex velocity profile, as shown in Figure 3.9. The exact extent of this viscous core, however, is not agreed upon entirely.

Spreiter and Sacks (1951) were amongst the first to derive a value for the vortex core radius, r'. These authors derived a value for the vortex core radius based on equating the total kinetic energy (both inside and outside the core) per unit length of the vortex pair to the induced drag. They assumed a Rankine style vortex, as shown in Figure 3.8, such that the core rotated as a solid body. Their calculations for an elliptically loaded wing gave $r' = 0.0775\ b$. Bilanin and Donaldson (1975) also used the $r' = 0.0775\ b$. Corsiglia, *et al*. (1973) used hot-wire anemometer surveys of the wing tip vortex to show that radius of the laminar core region is between 0.010 and 0.015 times the span.

Roberts (1983) has shown that the laminar subcore radius, r', can be expressed as

$$r' = 0.175\ [\ \frac{c}{2}\ log(\frac{1}{c})\ ]^{1/2}\ b \qquad [3.15]$$

where $c = \frac{40000}{Re}$ and, for a lifting wing, $Re = \frac{\rho\ V_o\ b}{\mu\ A_R}$. Based on a wing with elliptical spanwise loading and $Re = 10^7$, Roberts showed the laminar core radius to be $0.018\ b$. This radius was the radial position of the maximum tangential velocity. Roberts (1983) then showed there to be a turbulent core region extending to $r = 0.175\ b$, which corresponded to the position at which the tangential velocity of the core region attained the same V_{tan} of the potential vortex. Roberts' view on the velocity profile of the core region is shown in Figure 3.9, and corresponds to the approximate currently accepted form for the velocity profile in the vortex core in viscous flow.

The exact extent of the viscous region is still a matter of debate. As well, there is uncertainty as to whether the core is mostly laminar or turbulent, and where V_{tan} for turbulent flow will match V_{tan} for potential flow. Also in question is the exact relationship between the tangential velocity and the radius of the core region. Roberts (1983), Donaldson, *et al*. (1974), Corsiglia *et al*. (1973), Brown (1973), and McCormick, *et al*. (1968), all give good reviews and discussion of the velocity profile in the vortex core. Basically, all of these authors adhere to the idea that the core region velocity profile is similar to that already discribed, with the flow outside the core conforming to the potential flow field. The work of Brown (1973), however, differs from the rest in disputing the relationship of $V_{tan} = \frac{1}{r}$. Instead, Brown (1973) proposed that the relationship should be $V_{tan} = \frac{1}{r^{1/2}}$. This view apparently has not been supported elsewhere and, hence, is not pursued further at this time.

From the figures given by Roberts (1983) and Corsiglia *et al*. (1973), the maximum value of V_{tan} is seen to occur where $r' \cong 0.018\ b$. Also, from their figures, the value of the maximum V_{tan} of the viscous core is seen to be about the same as the value of V_{tan} for the potential vortex, provided that V_{tan} of the potential vortex is taken at $r \cong r'$, that

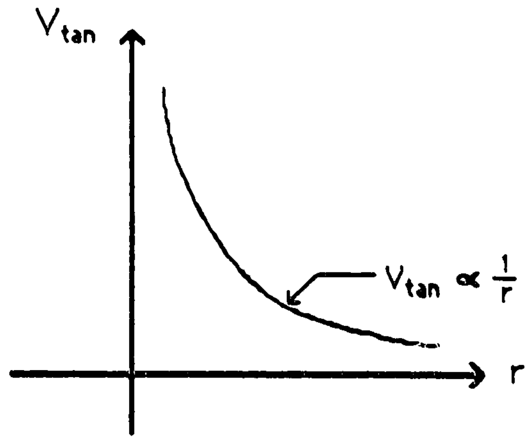


Figure 3.7 Vortex in Inviscid Incompressible Flow

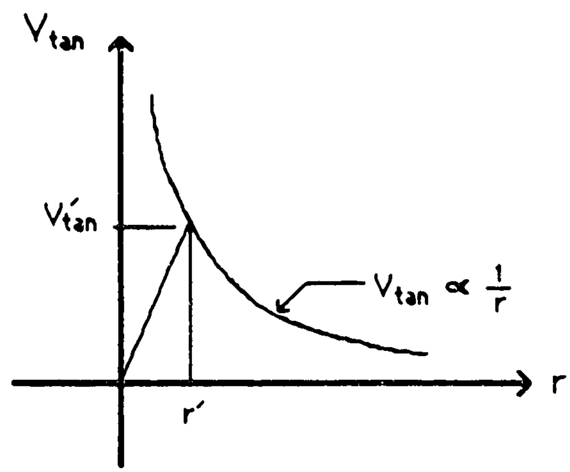


Figure 3.8 Vortex with Rankine Core

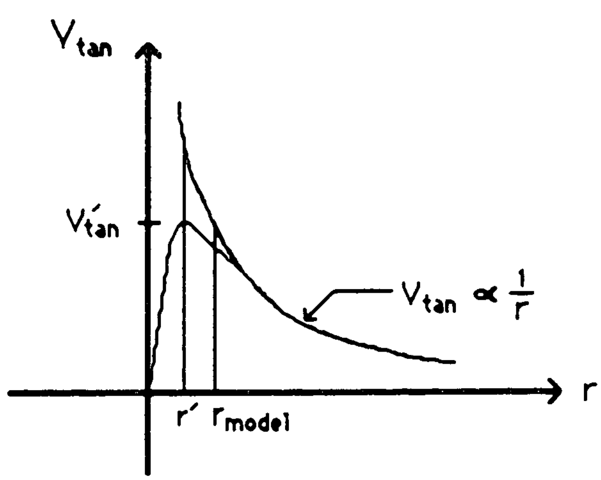


Figure 3.9 Vortex with Viscous Core Region

is, the radius where the maximum V_{tan} occurs in a viscous core. The value $r_{model} = 2r'$ is one of the radii of the Rankine core (as shown in Figure 3.9) compared using the model, as described in the next chapter.

3.10 Axial And Radial Velocity In Vortex Cores

The understanding of the axial velocity component, particularly its direction along the axis of the trailing vortex still needs some research. Batchelor (1964) was one of the first to consider,in a purely theoretical sense, the problem of the axial flow in vortices but did not provide conclusive evidence to show what was really happening. Roberts (1983) provided a very good discussion on the axial and radial flow in trailing vortices, and related that to the persistence and decay of the vortices. Corsiglia *et al.* (1973) showed the axial velocity at the center of the core to be in the order of 0.2 V_o, which is substantial from the view of the vortex life and energy considerations. However, for the concerns of aerial spraying, the simple displacement of spray in the axial direction of the vortex will not really affect the spray distribution. Possibly, there are effects to be considered which would involve the radial velocity near the core, but at this time there is insufficient knowledge of the phenomenon of radial velocity to incorporate into a vortex model.

4. COMPUTER MODEL ASSUMPTIONS

This chapter considers those assumptions used to develop the computer model to be used for spray movement prediction. The main emphasis is on those areas which differ appreciably from the theory as laid out in the preceding sections. Also considered is the additional information necessary for the purpose of model development. As will be seen, the model is a inviscid flow model, except within the vortex cores.

4.1 Coordinate System For Model

The coordinate system used for the model is a right handed Cartesian system. The origin is at the ground level directly below the quarter chord point of the wing and at the aircraft centerline. The x direction is along the right wing, the y direction along the direction of flight, and the z direction is vertically upwards. Figure 4.1 shows the axis arrangement.

4.2 Aircraft Wake Model

The circulation of the vortices can be determined using the relevant aircraft statistics and operating conditions. Most aircraft used for aerial spraying are of rectangular planform, with $A_R \approx 6.0$. Hence, the total circulation can

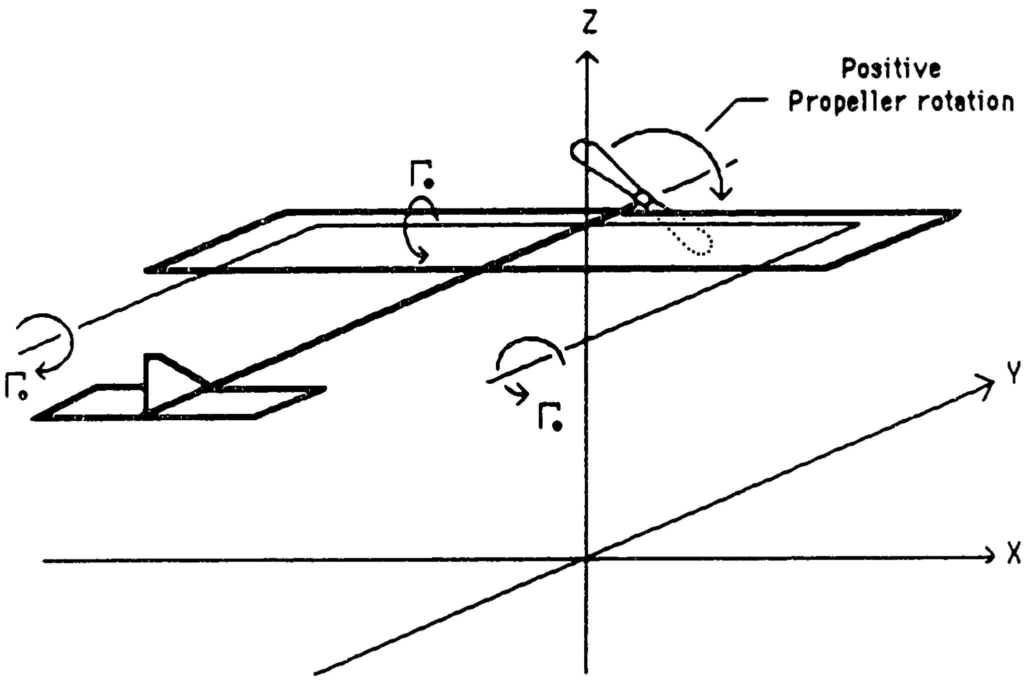


Figure 4.1 Coordinate System Arrangement for Model

then be taken to be the average of the values for the rectangular and elliptical loading. Since the aircraft for the flight tests, described in the next chapter, has $A_R \cong 5.5$, this is a reasonable value for the total circulation using those flight tests to verify the model.

The actual wake behind the aircraft is initially a vortex sheet, as was proposed by Prandtl (1921). This vortex sheet then rolls up to form two discrete vortices. In order to completely model the vortex sheet roll up and the spray movement would involve prohibitively large amounts of computer time. The use of a simplified model to predict spray movement was expected to use much less computer time to run the model, allowing more extensive tests.

4.2.1 The horseshoe vortex system

As already discussed, the model considered is a conventional simple horseshoe vortex system with straight trailing vortices as shown in Figure 3.1. The bound vortex location at 25% chord is considered to give adequate results for the induced velocities beyond a fraction of a chord from the trailing edge of the wing (Spreiter and Sacks, 1951). The spanwise extent of the bound vortex is determined by the initial spanwise separation of the two trailing vortices. The two completely rolled up trailing vortices are assumed to originate at the spanwise location where the bound vortex ends, the trailing vortices being attached to the wing at 25 percent chord so as to form a continuous vortex tube with the bound vortex. For the total vorticity of the generated vortex system to be zero the starting vortex must be considered. However, the starting vortex is not deemed to be relevant since it is left behind upon takeoff, keeping its influence far removed.

The trailing vortices are considered to be rolled up the entire time. This does not actually represent the true state, but for the conditions of droplet movement this has been determined to be adequate. Initial spray movement under the wing would be influenced by a flow which is basically induced by the shed vortex sheet which is dependent on the circulation distribution of the wing and the resulting shed circulation.

As discussed in the previous chapter, the initial separation of the trailing vortices in close proximity to the ground, may not be the same as that in unconfined flow. The models, WAKE77 and NASA77, treat the percent spanwise initial vortex separation of the two hypothetical rolled up vortices as an input variable. Then, a scheme is used to adjust the initial vortex separation, b_o', such that if $h \geq b$, then $b_o' = 0.82b$, and when $h = 0$ then the initial separation equals the input value. When the aircraft is at a height such that $h < b$, a linear interpolation, based on the height of the aircraft, is used to determine the percent span initial vortex separation, INISEP. The scheme adopted has been

$$\text{INISEP} = \text{PSVSEP} - (\text{PSVSEP} - 82) \times \frac{h}{b}$$

where

$$\text{INISEP} = \left(\frac{b'}{b} \right) \times 100$$

$$\text{PSVSEP} = \left(\frac{b'_o}{b} \right) \times 100$$

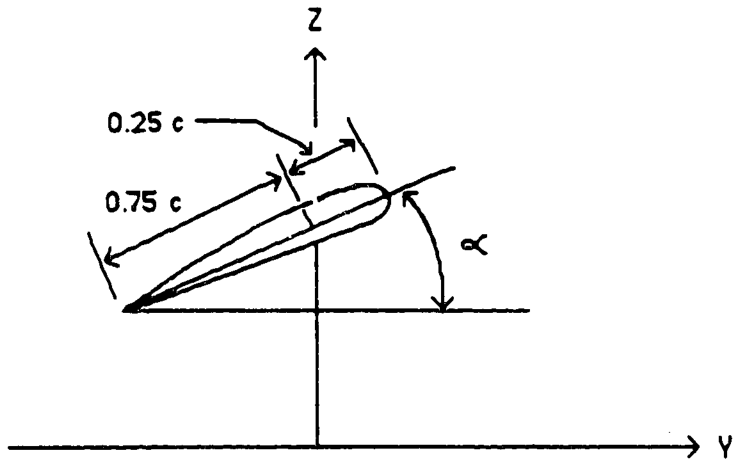
Given that, PSVSEP is the input value for the initial percent spanwise separation of the trailing vortices, b'_o is the initial separation of the vortices, b' is the vortex separation at any time and height, and b is the wingspan. The value of 82 is given to represent the 82% spanwise location for the initial separation of the trailing vortices from a rectangular wing with $A_R \approx 6$.

The initial height of the vortices above the ground is determined in the following manner. The model considers the height of the aircraft to be measured to the trailing edge of the wing at the centerline. Hence, using the relevant aircraft statistics and operating conditions, the angle of attack is computed. Since the bound vortex and the trailing vortices originate at the 25% chord point, their height from the ground, will be dependent on the angle of attack of the wing, as shown in Figure 4.2. The dihedral of the wing, also, will affect the initial height of the trailing vortices, since the further out towards the tip the vortices are considered to be attached, the higher they will be with respect to the wing root, as shown in Figure 4.3.

4.2.2 Vortex core size

The vortex core has been modeled using a Rankine vortex core, since the simple Rankine core provides some means to show the viscous effects in the core region where viscosity plays a dominant role. The Rankine core gives the linear relation between V_{tan} and r as shown in Figure 3.8, with the maximum tangential velocity at radius r'. This value of r', also, is considered to be the edge of the viscous core of the Rankine vortex. At that location the tangential velocity of the potential vortex is the same as that of the Rankine core. For the model, the value of $r' = 0.0775\ b$ is used, after the manner of Spreiter and Sacks (1951), because when using a Rankine style vortex the smaller r', as given by Roberts (1983), would result in a maximum V_{tan} greater than the actual case. This is shown in Figure 3.9. Thus, if a drop passes through the core region of a vortex in the model, it will experience more realistic induced air velocities near the edge of the core, even though the actual core edge will be displaced somewhat.

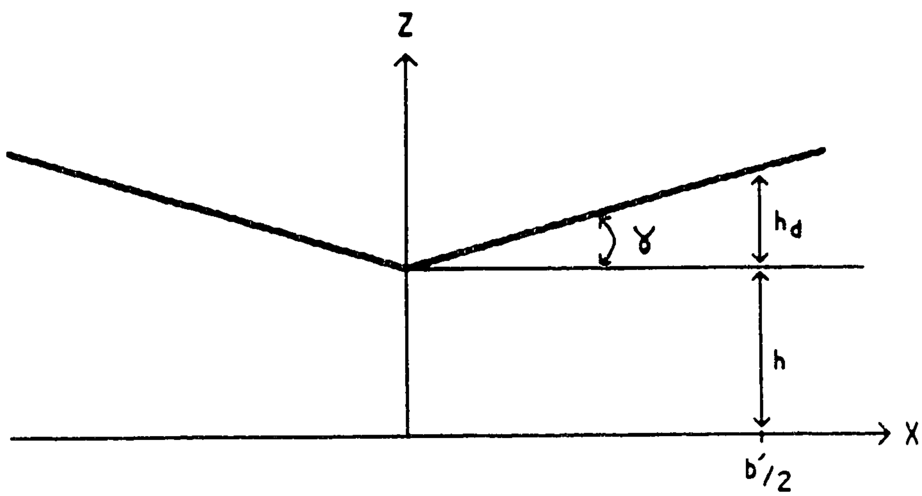
For model testing, the results of which will be discussed later, several values of r' were tried in order to determine the effect of using different vortex core sizes.



0.25 c = Bound Vortex Chordwise location

α = Angle of Attack

Figure 4.2 Angle of Attack of Wing



Vortex height $= h + h_d$

$= h + b'/2 * \tan(\gamma)$

b' = Lateral Position of trailing vortex

γ = Dihedral Angle for Each Wing

Figure 4.3 Dihedral Angle for Wing

4.2.3 Propeller model

A simplified propeller model was used, since the interference caused by the aircraft body was not considered. As mentioned in the last chapter, the propeller could be modeled with a complex program using a helical vortex pair originating at the propeller tips, as well as an axial vortex, such as that developed and used by Bragg (1977). However, if the whole aircraft is not considered, such a complex model does not seem warranted.

The propeller was modeled as a simple line vortex with a Rankine core and no axial or radial flow. In actual conditions, the axial flow would only displace the spray along the direction of the flight path, and the radial flow is not significant in comparison with the tangential flow. As with the trailing vortices, the maximum V_{tan} will occur at some radius, r'. For the propeller, this will correspond to the diameter of the propeller, D_{Prop}. Thus, for the model, the propeller vortex rotates as a solid body within D_{Prop} at the rotational velocity given by the value ω'. Outside this radius the propeller vortex behaves as a potential vortex.

The value for a swirl coefficient is a variable in the program, however, the correct value to use was in question. For a propeller alone in an airstream, propeller design theory can be used to determine the correct value. To do this, the power output, efficiency and the operating characteristics of the propeller must be known. For the flight test conditions, as given in the next chapter, insufficient information was available to do this. As well, the complete aircraft (the fuselage, wings, radial engine and tail) causes significant interference.

For modeling purposes, the value of the swirl coefficient was varied in the model between 0 and 0.010 in order to determine the most suitable value. Upon examining the test results it was readily apparent that a high value of 0.0075 for the swirl coefficient significantly reduced the degree of correlation of the model with the flight test data. According to the results of the model, the use of a much lower swirl coefficient appears to lead to very good correlation of the model results with the flight test data. As seen later, a value of approximately 0.0035 to 0.0040 for the swirl coefficient seemed most appropriate for the test aircraft.

4.3 Motion Of Trailing Vortices

The motion of the trailing vortices after formation and roll up is a result of the total air velocity at the core of each. This can be seen to be the sum of the induced velocities from the other vortices, as well as the contribution from the propeller and crosswind. Since the motion is mostly within a short distance from the ground, the vortex images, which are used in order to satisfy the condition of no normal flow at the ground plane, must also be considered.

Clearly, the bound vortex and its image do not contribute to the lateral movement of the trailing vorticies, since their axes lie perpendicular to the axes of the trailing vortices. The bound vortex and its image, however, do contribute to the downward movement of the trailing vortices and propeller vortex when considered in close proximity to the wing. This effect is only for a short period of time, and rapidly diminishes as the distance from the wing is increased. Hence, the effect of the bound vortex and its image on the movement of the trailing vortices and propeller vortex is neglected entirely for the computer model.

Actual trailing vortices, once rolled up, will be curved semi-infinite line vortices because they stay attached to the wing as it moves away, and at any point they move down and outwards laterally, depending on the wind and ground plane proximity. To greatly simplify mathematics, the vortices are initially placed at a particular given percentage spanwise location, thenceforth they move as straight line vortices. The error introduced here due to the actual curvature will be negligible. As well, the trailing vortices will no longer connect with the bound vortex, which is still considered to be attached to the wing at the 25% chord position and to extend to the spanwise location where the trailing vortices were originally attached. Since the rate of lateral movement, of the trailing vortices, is low compared with the aircraft speed (at least in conditions of low wind as in spraying conditions) the vortices will only make a very small angle with the flight path.

The ground plane acts to increase the separation between the vortices with time and, thus, will minimize the error involved in the precise placement for initial location.

4.3.1 Induced velocities at vortex cores

The induced velocities at core n due to another vortex m is determined as follows. The numbering of the vortex cores and the coordinate system is shown in Figure 4.4. For the trailing vortices and their images, (n, m = 1 to 4),

$$G = \frac{\Gamma_o}{2\pi} \qquad [4.1]$$

For the propeller and propeller image, (n, m = 7, 8),

$$G = \omega' D_{Prop} \qquad [4.2]$$

Then the induced velocities at n are

$$V_{xind} = \frac{G (z_n - z_m)}{(x_n - x_m)^2 + (z_n - z_m)^2} \qquad [4.3]$$

$$V_{zind} = \frac{G (x_n - x_m)}{(x_n - x_m)^2 + (z_n - z_m)^2} \qquad [4.4]$$

where n = 1 to 8 and n ≠ 5,6

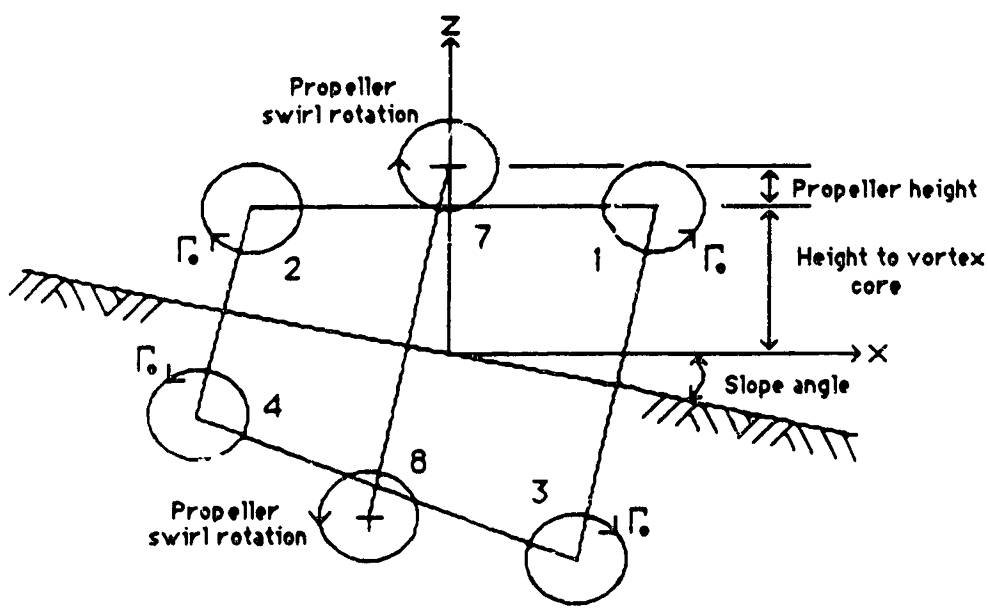


Figure 4.4 Numbering of Vortex Cores

and for each n, m = 1 to 8, and m ≠ n, and m ≠ 5,6.

Now the total x and z component velocities at the location of core n are given by

$$V_{xind}^{total} = V_{xxwnd} + \Sigma_{m=1}^{8} V_{xind} \quad [4.5]$$

$$V_{zind}^{total} = V_{zxwnd} + \Sigma_{m=1}^{8} V_{zind} \quad [4.6]$$

where V_{xxwnd} and V_{zxwnd} are the x and z components of the crosswind, as will be explained in the next section. Note that, for both summations, m ≠ n, and m ≠ 5,6. This is because a vortex cannot induce a velocity on its own core; and the bound vortex and its image are numbered m = 5 and 6, respectively, and are not considered to produce any effect on the motion of the trailing vortices.

4.4 Crosswind Model

The ability to model the wind is essential for spray simulation since the wind is the major atmospheric influence. The main concern is the crosswind component since any headwind or tailwind merely affects displacement of the spray along the flight path. As a result only the crosswind component of the wind is included in the model. The crosswind is modeled as flowing parallel to the ground plane, which is assumed to be sloping uniformly across the area of concern. The direction of flow is such that a crosswind from the left to the right is considered positive, consistent with the coordinate system.

In modeling the crosswind component, it is necessary to consider the proximity of the ground plane and the boundary layer effect on the variation of the crosswind with height. At any particular height above ground level the crosswind will be considered to be a steady flow. The variation with height follows a scheme used by Morris *et al.* (1984), such that at any height z the crosswind component is V_{xwnd}.

$$V_{xwnd} = V_{mxwnd} \frac{\ln(h_{mxwnd}/h_{surf})}{\ln(z/h_{surf})} \qquad [4.7]$$

Where h_{surf} is $\frac{1}{30}$ the physical surface roughness height h_{phys}. V_{mxwnd} is the measured crosswind component at height h_{mxwnd}.

Figure 4.5 shows a plot of $\frac{V_{xwnd}}{V_{mxwnd}}$ versus $\frac{\ln(h_{mxwnd}/h_{surf})}{\ln(z/h_{surf})}$. The figure is based on h_{mxwnd} = 3.048 m and h_{phys} = 0.3048 m.

4.5 Ground Plane Boundary Conditions

The ground plane boundary condition for the flow is such that there can be no component of flow normal to the ground plane itself. To satisfy this condition, the crosswind is assumed to be parallel to the uniformly sloping

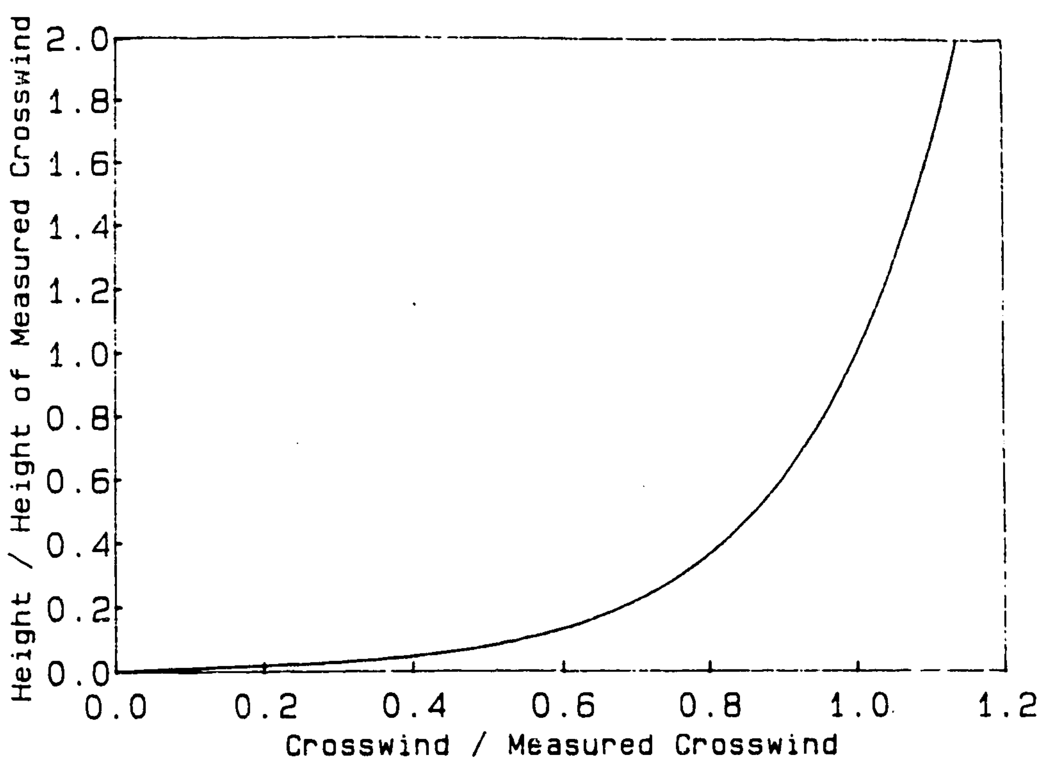


Figure 4.5 Crosswind Model based on h_{mxwnd} = 3.048 m and h_{phys} = 0.3048 m

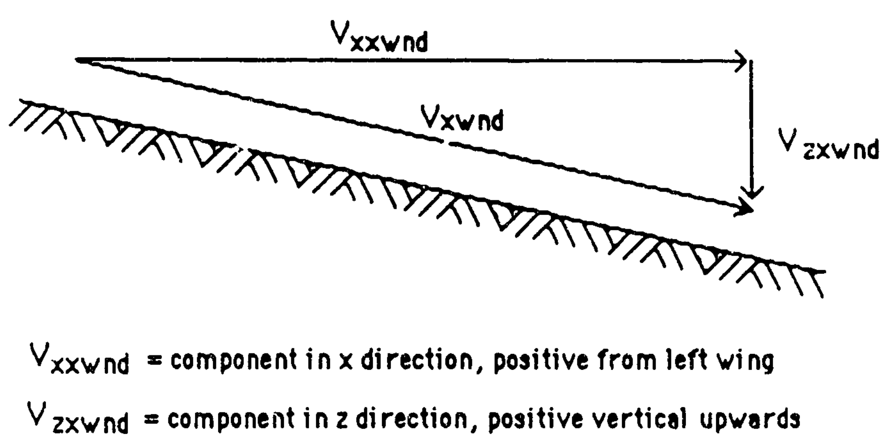


Figure 4.6 Crosswind Components on Sloping Ground

ground surface. Figure 4.6 shows the crosswind, V_{xwnd}, and the x and z components thereof, V_{xxwnd} and V_{zxwnd}.

As well, there are image vortices of the bound vortex, the trailing vortices, and of the vortex used to simulate the propeller. Figure 4.4 showed the arrangement of the image vortices.

4.6 Evaporation and Drop Life

For the computer model, the droplife, τ, is calculated using the terminal velocity of the drop, as explained in chapter 2. As was noted, the model predicts a slightly shorter drop life than would be the case for a drop continuing to fall at terminal velocity. However, the drop will have to decelerate from its initial velocity, upon emission from the nozzle, to its terminal velocity. During that time, the rate of evaporation will be higher than the value while moving at terminal velocity. Sjenitzer (1952) estimated the fractional evaporation from drops to be from 30% to 10% of total evaporation for drops 0.1 to 1.0 mm respectively during deceleration. Thus, he suggested that consideration be given to the higher velocity during deceleration to determine the amount of evaporation. Considering Sjenitzer's comments, the lower drop life as determined by the model should be sufficiently accurate to produce reasonable results.

4.6.1 Evaporation effects on spray distribution

The problem of evaporation rate of a droplet was simplified by considering the evaporation rate to be independent of spray composition, as discussed in chapter 2. As was suggested, the user of the computer program could simply adjust the wet bulb depression according to the suspected change in volatility of the drop for compositions other than just water.

The model can show the effects of evaporation on the spray distribution for the first 20 seconds. The model stops at that time, since either the drop would be so small as to be insignificant on a volume basis and is considered lost, or the particle would be drifted out of the spray target area, and again considered lost. Thus, the rate of evaporation can be tested to check its effect on the spray distribution.

4.7 Drop Movement

To determine the droplet movement in the aircraft wake, a Lagrangian formulation was used, meaning that the path of each droplet was followed from beginning to end. Generally, the induced velocities were calculated at any position considering the bound and trailing vortices, the propeller vortex, and the image vortices, as well as the crosswind. The drag force on the droplet was calculated, then the acceleration of the droplet. Using the instantaneous location, velocity, and acceleration of the drop, its position can be determined after some small time increment dt. This process was repeated until the drop reached the predetermined height for the simulated crop canopy or the ground plane.

4.7.1 Induced velocities at droplet location

The induced velocities at location (x,y,z) due to the vortices are determined in a manner similar to that for the trailing vortex core movement. The equations describing these velocities are much like those influencing the trailing vortex movement, but with a few extra conditions involving the position of the droplet.

For the trailing vortices and their images, m = 1 to 4 , let

$$G = \frac{\Gamma_o}{4\pi} \qquad [4.8]$$

For the bound vortex and its image, m = 5 and 6, let

$$G = \frac{\Gamma_o}{4\pi} \qquad [4.9]$$

and for the propeller vortex and propeller vortex image, m = 7 and 8, let

$$G = \omega' D_{Prop} \qquad [4.10]$$

The induced velocities at (x,y,z) are then as follows:

For m = 1 to 4, 7 and 8

$$V_{xind} = \frac{G\,(z - z_m)}{(x - x_m)^2 + (z - z_m)^2} \qquad [4.11]$$

$$V_{yind} = 0 \qquad [4.12]$$

$$V_{zind} = \frac{G\,(x - x_m)}{(x - x_m)^2 + (z - z_m)^2} \qquad [4.13]$$

and for m = 5, 6

$$V_{xind} = 0 \qquad [4.14]$$

$$V_{yind} = \frac{G\,(z - z_m)}{(y - y_m)^2 + (z - z_m)^2} \qquad [4.15]$$

$$V_{zind} = \frac{G\,(y - y_m)}{(y - y_m)^2 + (z - z_m)^2} \qquad [4.16]$$

Now, the total x, y, z component velocities or the air at the location of the droplet are given by

$$V_{xind}^{total} = V_{xxwnd} + \Sigma_{m=1}^{8} V_{xind} \qquad [4.17]$$

$$V_{yind}^{total} = \Sigma_{m=1}^{8} V_{yind} \qquad [4.18]$$

$$V_{zind}^{total} = V_{zxwnd} + \Sigma_{m=1}^{8} V_{zind} \qquad [4.19]$$

Again, V_{xxwnd} and V_{zxwnd} are the x and z components of the crosswind.

Also to be considered is that the bound vortex is a finite vortex and so the angles from the two ends must be considered when m = 5, 6. The value for the induced velocity will then include the factor ($\cos(\beta_{ma}) + \cos(\beta_{mb})$) , with m = 5, 6. Here, β_{ma} is the angle to one end of the finite vortex and β_{mb} is the angle to the other end, as shown in Figure 4.7.

For the trailing vortices and images, m = 1 to 4, a similar condition arises since they are semi-infinite vortices. Then, the value for the induced velocity will include the factor ($1 + \cos(\beta_m)$) , where β_m is the angle to the end of the semi-infinite vortex, as shown in Figure 4.7.

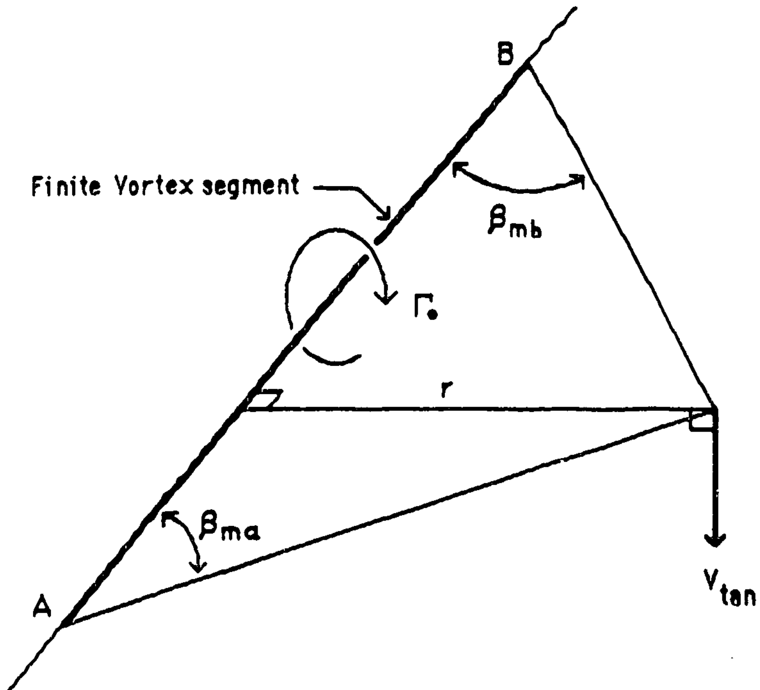
If the drop is outside of the core, as will usually be the case, the formulas will hold as they stand. However, when the drops pass within the core of the vortices the induced velocities will be reduced linearly with the radius from the center of the core, since the vortices are modeled as Rankine vortices. As a result, for any position and vortex, if the radius, r, to the drop is less than r', the edge of the Rankine core, the induced velocity includes the factor $\left(\frac{r}{r'} \right)^2$.

For the trailing vortices, the propeller vortex, and their images,

$$r = [(x - x_m)^2 + (z - z_m)^2]^{1/2} \qquad [4.20]$$

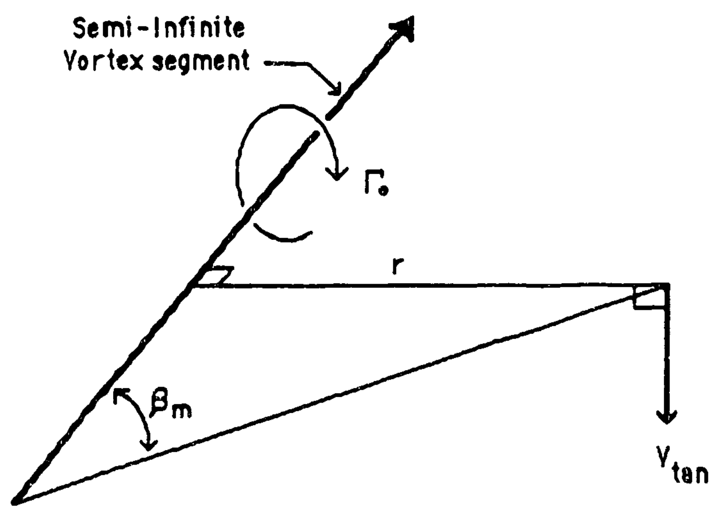
While for the bound vortex and its image,

$$r = [(y - y_m)^2 + (z - z_m)^2]^{1/2} \qquad [4.21]$$



$$V_{tan} = \frac{\Gamma_o}{4\pi r} \left(\cos(\beta_{ma}) + \cos(\beta_{mb}) \right)$$

Finite Vortex Segment



$$V_{tan} = \frac{\Gamma_o}{4\pi r} \left(1 + \cos(\beta_m) \right)$$

Semi-Infinite Vortex Segment

Figure 4.7 Finite and Semi-Infinite Vortices showing angles to ends

The value of r' for the trailing vortices is given as r' = CORCOEF × b. The value of the core coefficient, CORCOEFF, was normally taken as 0.0775, but this was varied in the model tests, also. For the propeller vortex, r' = D_{Prop} as already discussed.

It should be noted that with the bound vortex located at 25 percent chord and a nozzle positon behind the wing, the drop is considered to be unable to enter within the bound vortex core. Neither is it possible for any drop to enter within the core of any vortex image.

The propeller vortex and its image have been modeled as infinite line vortices since they are merely approximations to the actual case, with the best fit swirl coefficients used.

To determine the correct directions for the induced velocities, the ability to determine the position that the drop takes relative to the center of each vortex core is included in the model.

4.7.2 Drag force on the droplets

Once the induced velocities are determined for any position, the force on the drop due to drag may be calculated. The forces are calculated for the component directions as was given in section 2.1.1. The relative velocity components in each direction are used to determine the component forces and, hence, the accelerations. The formulas are repeated here for convenience.

$$F_{Dx} = \frac{1}{2} \rho_a V_{xrel}^2 A C_D$$

$$F_{Dy} = \frac{1}{2} \rho_a V_{yrel}^2 A C_D$$

$$F_{Dz} = \frac{1}{2} \rho_a V_{zrel}^2 A C_D$$

$$a_x = \frac{F_{Dx}}{m}$$

$$a_y = \frac{F_{Dy}}{m}$$

$$a_z = \frac{F_{Dz} - mg}{m}$$

The component accelerations of the drop simplify to the form used for the model.

$$a_x = \left(\frac{3 \rho_a C_D}{4 \rho_s D} \right) V_{xrel}^2 \qquad [4.22]$$

$$a_y = \left(\frac{3 \rho_a C_D}{4 \rho_s D} \right) V_{yrel}^2 \qquad [4.23]$$

$$a_z = \left(\frac{3 \rho_a C_D}{4 \rho_s D} \right) V_{zrel}^2 - g \qquad [4.24]$$

where V_{xrel}, V_{yrel}, and V_{zrel} are the component velocities of the droplet relative to the air.

4.7.3 Runge-Kutta integration scheme

The droplet movement was modeled using a Runge-Kutta algorithm for second order ordinary differential equations, in two dimensions as given by Dodes (1978). Using this scheme, only the droplet position is needed at any time, in order to predict the next position, given the velocity and acceleration of the droplet at the current location.

4.7.4 Time increment for drop movement simulation

The time increment chosen for simulating the movement of the drop affects the numerical stability of the simulation. The initial time increment is set to 0.000625s, 0.0000625s, or 0.00000625s for the initial droplet diameters of $D_o \geq 100\ \mu m$, $100\ \mu m > D_o \geq 10\ \mu m$, and $D_o < 10\ \mu m$ respectively. These increments were determined to be adequate for numerical stability of the model simply by monitoring the Reynolds number for the droplet during trial runs of the model. The crucial period is when a droplet is first released from the nozzle. That period produces the greatest deceleration forces and hence greatest possibility of instability. Larger initial time increments were tried without success as the Reynolds numbers developed instabilities.

The time increment was continually modified during the remaining droplet movement, the maximum time increment being 0.04s. Also, the Reynolds number was continually monitored to ensure numerical stability of the simulation. This procedure proved quite satisfactory for consistent results and provided the means to improve the computer model efficiency. The use of an integration scheme other than the Runge-Kutta would necessitate a change in the initial time increment and the prcedure to modify the time increment.

4.8 Spray Nozzle Models

The ability to model several different types of spray nozzles was seen to be a major requirement for any spray model. Hence, the model includes the ability to model three types of nozzles. These include hollow cone nozzles, flat fan nozzles, and rotary nozzles. Figures 4.8, 4.9, and 4.10 show the nozzle models used and the relevant angles.

The model provides the means to simulate the emission of any number of drops from each nozzle for the specified configuration. The number of droplets is specified, then, for whatever nozzle type being used, the droplets are distributed evenly throughout the spray pattern of the nozzle. For the hollow cone nozzle, the drops are spread evenly around the cone surface. For the flat fan, the drops are distributed evenly across the fan angle. For the rotary nozzle, the drops are released at uniformly spaced radials in a plane.

Also included in the model is a single drop nozzle, which allows the user to specify for a single drop the exact velocity and direction of emission. This is useful to determine the exact path a particular drop takes. The program, WAKE77, is set up so that the user may store the data for the path of each spray droplet, then use the data to plot the trajectories later.

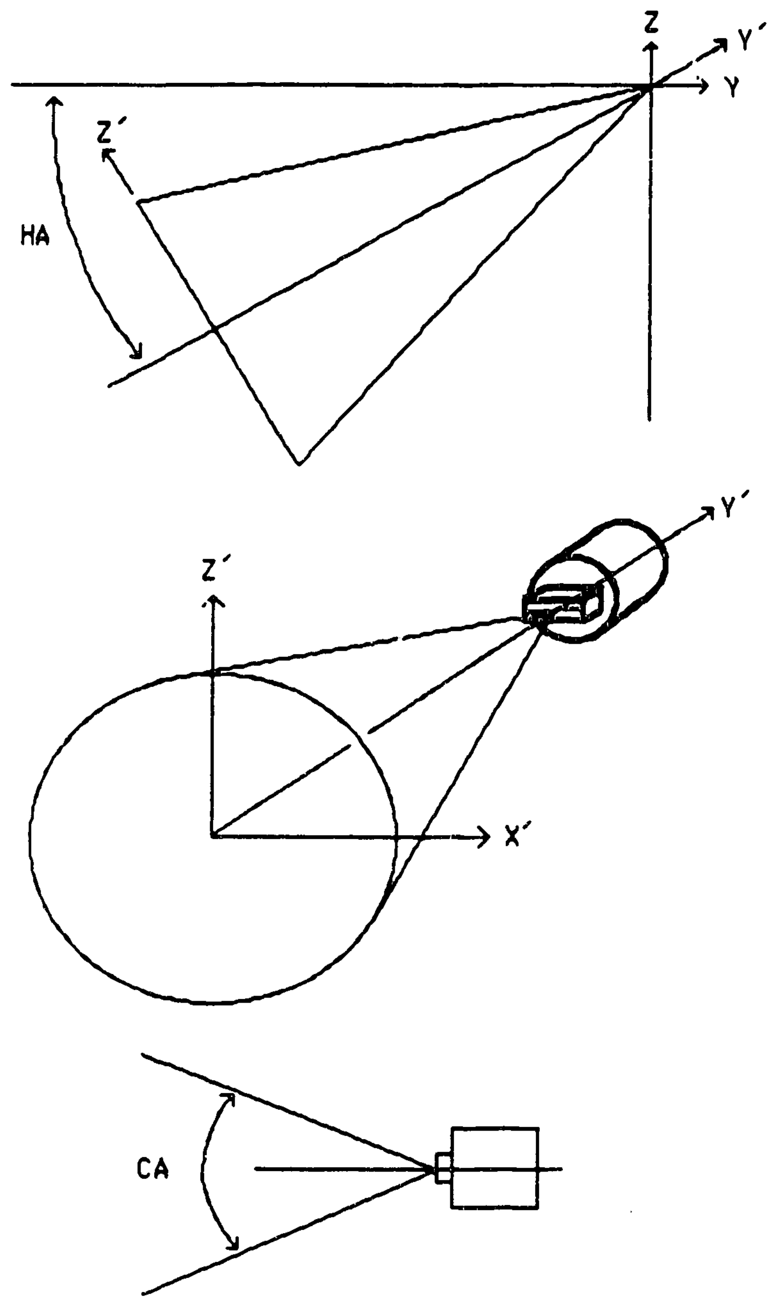
Another feature required for specification of the nozzle configuration is the angle that the axis of the nozzle makes with the horizontal. This is labeled the horizontal angle in the nozzle figures. Again, this is one of the parameters allowed in the computer model. It should be mentioned that there is no flexibility allowed in the yaw direction of the nozzles. That means, the nozzles can only be aimed in a plane which is parallel to the plane specified by the y and z axes.

4.8.1 Hollow cone nozzle

The hollow cone nozzle is so named since it produces a spray pattern which is cone shaped, with the apex at the nozzle itself and the emanating spray forming the cone shape with no spray within the cone surface. The cone angle specifies the exact shape of hollow cone spray pattern. Figure 4.8 shows the model of the hollow cone spray pattern, as well as the relevant angles and directions.

4.8.2 Flat fan nozzle

The flat fan nozzle is so named since it produces a spray pattern which is fan shaped with the apex at the nozzle itself and the emanating spray forming a flat fan shape with virtually no spray outside of the fan itself. Figure 4.9 shows the model of the flat fan spray pattern, as



HA = Angle from Horizontal
CA = Cone Angle

Figure 4.8 Hollow Cone Spray Nozzle Model well as the relevant angles and directions.

4.8.3 Rotary nozzle

The rotary nozzle modeled is one for which the spray moves along a rotating disk radially outwards to the edge of the disk at which point it is flung free of the disk. The resulting velocity of the drops is tangential to the circumference of the disk, and is a function of the speed of rotation of the disk. Notice that diminishing the diameter of the rotary nozzle to a point results in the same velocity directions as if a point source were emitting fluid in a radial direction in a plane. As a result, the model simply neglects the diameter of the disk and uses the point source approximation. This could be achieved, also, by using a degenerate form of a hollow cone nozzle, i.e. one with a cone angle of 180°. Figure 4.10 shows the model of the rotary spray nozzle pattern, as well as the relevant angles and directions.

4.8.4 Initial velocity of spray emitted from nozzles

Consideration of the conversion of spray pressure in the nozzle to the velocity of the spray drop basically follows Bernoulli's theorem. In accordance with the work of Goering, *et al*. (1972) the initial velocity of the spray emitted from the nozzle, v_s, can be found as follows:

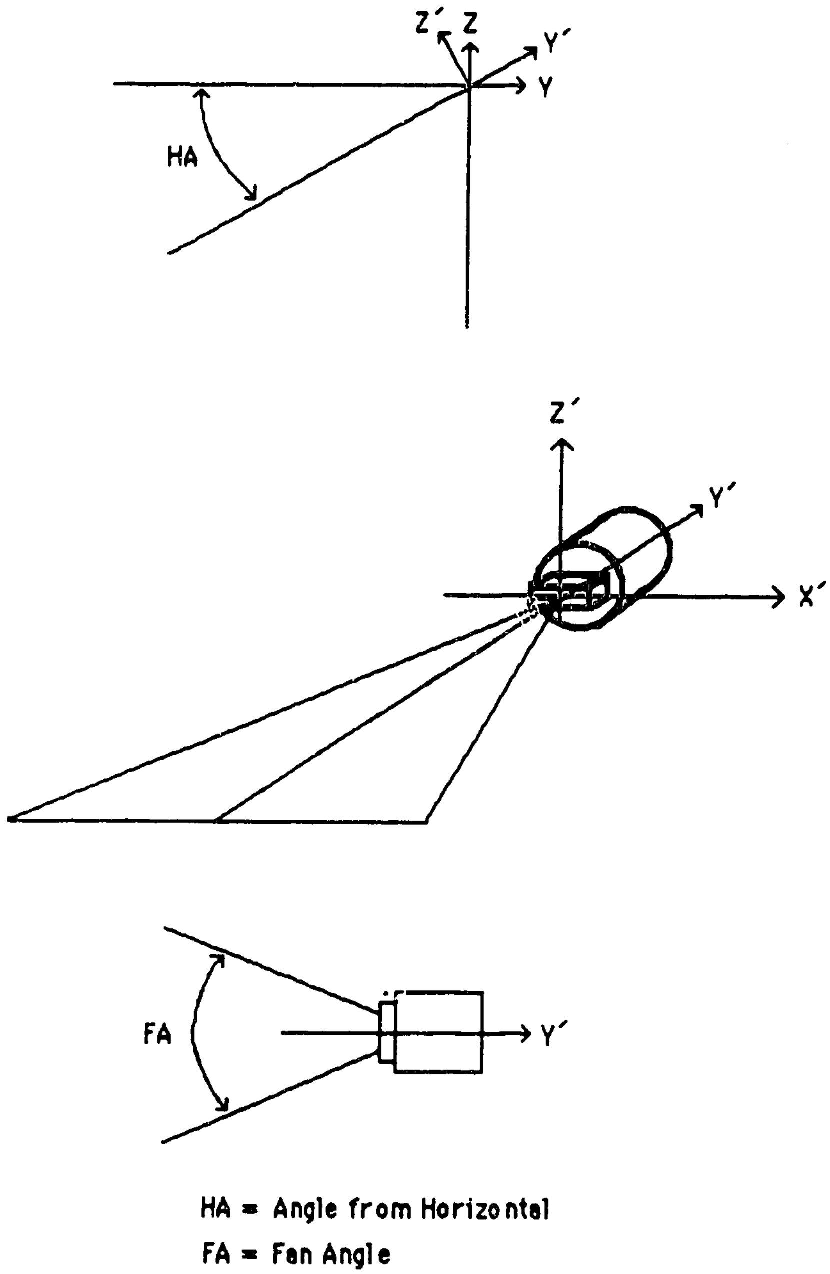


Figure 4.9 Flat Fan Spray Nozzle Model

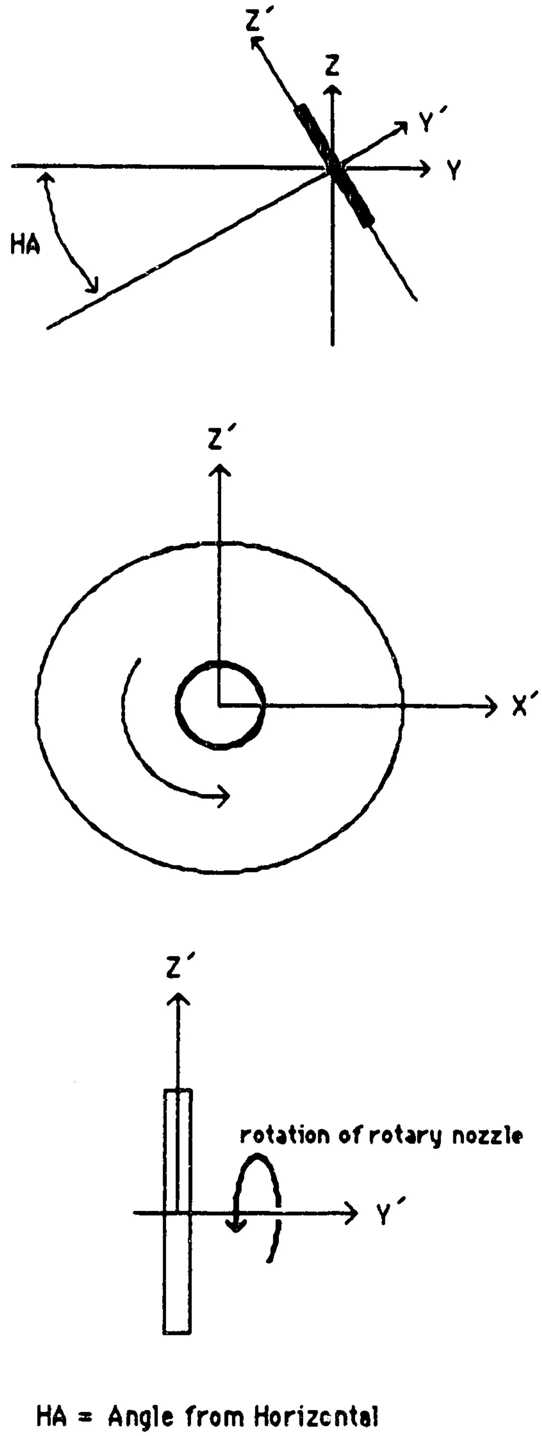


Figure 4.10 Rotary Spray Nozzle Model

$$V_s = \left(\frac{2\ \nu\ P_n}{\rho_s} \right)^{1/2} \qquad [4.25]$$

where

P_n = gauge pressure on nozzle, Pa

ρ_s = density of spray, kg/m³

ν = efficiency of atomizing process

The value ν shows the energy losses involved in converting the pressure to spray velocity. Amberg and Butler (1969) considered this value to be about 0.80, with which Goering concurred and, hence, used in his work. As a result, for a nozzle pressure of 276 kPa the initial velocity of water as a spray will be about 21 m/s. For the case of rotary nozzles, the rotational velocity and nozzle diameter dictate the velocity of emission. Hence, in the computer model, WAKE77, the release velocity of the drop for the rotary nozzle is required input.

For all of the nozzles, the initial velocity of a droplet is determined by its exact angle of release from the nozzle and the angle the nozzle axis makes with the horizontal. The program subroutines NOZZLE and NZVELS provide the means to determine the emission velocity of each droplet with respect to the nozzle. The velocity of the aircraft is then added to the y velocity component to determine the total velocity of each droplet.

4.8.5 Nozzle placement along the span

The placement of spray nozzles along the span is also a necessary consideration to allow flexibility in modeling a particular aircraft spray system. Hence, the model development includes this feature as one of the subroutines for data input, NOZLOC. The choice is given for even spacing of the nozzles along the span between two specified spanwise locations, or uneven spacing of the nozzles requiring the location for each nozzle to be input. The two alternatives provide the ability to readily specify the configuration of any number of nozzles on the aircraft.

4.8.6 Position of the nozzle from the wing trailing edge

The release point of the droplet with respect to the trailing edge of the wing is considered to be POSIY and POSIZ. These are the displacements from the trailing edge in the y and z directions of the model,respectively. These are shown in Figure 4.11.

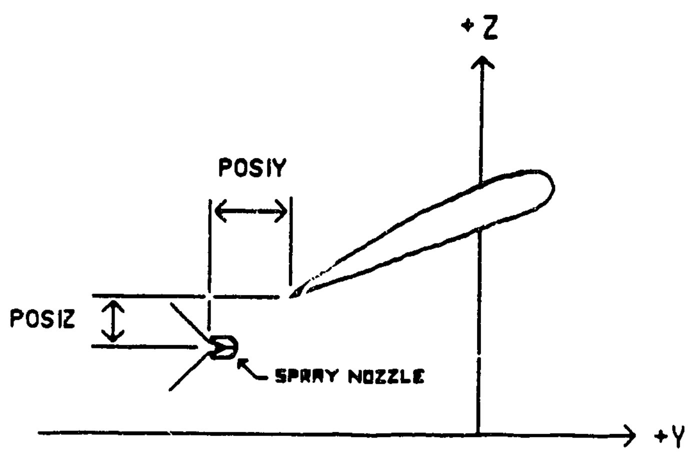


Figure 4.11 Position of a spray nozzle from the wing trailing edge

4.9 Combination of Overlapping Spray Passes

A scheme was developed for the combination of overlapping spray passes or swaths. The distance between the centers of the flight path on adjacent passes is considered to be the swath width. If any spray moves laterally outside of the swath, then the adjacent swaths are combined in the regions of overlap. For adjacent passes in the same direction, this was done by shifting the deposit from the swath to the left and right by one swath width, then combining with the center swath deposit to give the final spray deposition amount. If spray drifts more than one swath width, then a second swath, away from the centerline, is combined with the first, in a similar fashion. If the adjacent passes are made in opposite directions, then the spray distribution for the adjacent swaths must be recalculated. This is because the direction of propeller swirl will be reversed for the adjacent swaths. This causes the modeling of spraying with adjacent runs in opposite directions, to take basically twice the computer time that it takes for modeling of adjacent runs in the same direction.

4.9.1 Spray uniformity calculations

The uniformity of the spray swath is calculated in accordance with the procedure layed out in the ASAE

Standards (1985). The width of the spray collection strip is considered to be the central section of the swath which effectively provides complete swath overlap. The procedure which is repeated here, results in the calculation of the standard deviation of the spray deposition across the collection strip.

$$\sigma = \left(\frac{\Sigma(X_i - \bar{X})^2}{n - 1} \right)^{1/2} \qquad [4.26]$$

$$= \left(\frac{\Sigma X_i^2 - (\Sigma X_i)^2/n}{n - 1} \right)^{1/2} \qquad [4.27]$$

and

$$\eta = \frac{\sigma \times 100}{\bar{X}} \qquad [4.28]$$

where,

$\bar{X}$ = arithmetic mean of readings

X_i = reading for one collector location for the combined swaths

n = number of collector locations used

σ = standard deviation

η = coefficient of variation

The value of the coefficient of variation, η, gives the measure of the uniformity of the spray distribution over the collection path considering the spray from overlapping swaths. As can be seen by the formulas, increased uniformity is seen by lower values of η.

In the model, the coefficient of uniformity is determined both on the basis of volume and the number of droplets deposited at a collector location.

Since the model only simulates one drop size for each run, the spray distribution from various drop sizes should be combined in order to give a measure of uniformity for combined drop sizes.

4.10 Computer Model Input

The computer model, WAKE77, can be run in an interactive mode such that the user is clearly prompted for all needed information for the particular simulation. The WAKE77 program, also, is organized so as to accept the flight test data input from two data files.

To test and verify the simulation model accuracy, the model was revamped to accept flight test data input from files providing the needed individual run and drop information. That program is called NASA77, since the test data was provided by NASA (the United States National Aeronautics and Space Administration). The conditions for each flight are given in the tables and appendices of Morris *et al*. (1984). Further explanation of the model testing and verification is given in the next chapter.

The WAKE77 model and NASA77 model are included in Appendix I and II, respectively. Also found there is the required information needed to run the programs and to understand the output.

5. MODEL TESTING AND VERIFICATION OF PARAMETERS

This chapter considers the verification of the simulation model which was developed. The intent was to determine how well such a simplified model of the aircraft wake could model spray movement in the wake. Hence, the model verification was a significant step in the work.

The verification was done by comparing results produced by the computer program, NASA77, to actual flight test data. For the comparison, the computer program predicted the spanwise location at which a drop would impact a collector array, henceforth referred to as the 'predicted' location. These predictions were based on the flight test conditions for each pass of the aircraft, as detailed by Morris *et al.* (1984). From the work of Morris *et al.* (1984), the final spanwise location of a drop was determined, for the given flight test conditions, as explained in the next section. The final location determined from the work of Morris *et al.* is henceforth referred to as the `actual' location.

5.1 Flight Test Data And Facilities

To verify the model accuracy, it was necessary to compare the model data with reliable flight test data. The data from Morris *et al.* (1984) was chosen because of the apparent accuracy of their data. Also, a significant consideration was the completeness of their report. Basically, all of the required information was included regarding the actual conditions for the flight tests.

To simulate spray particles, commercially available, spherical, polystyrene beads, with specific gravity of 0.65, were used to simulate the spray droplets of predetermined diameters. The modeling scheme used followed that of Ormsbee and Bragg (1978). Two sizes of simulated spray droplets were used to represent spray sizes (150 and 300 microns diameter) used in typical insecticide and herbicide applications. The actual polystyrene beads used to simulate these two diameters were spheres 300 to 355 microns diameter and spheres 600 to 700 microns diameter. However, in the prediction model the spheres were all assumed to be 327 and 650 microns diameter, respectively.

Morris *et al.* (1984) included the particle deposition plots for 68 passes of an aircraft over three collector arrays, which collected the polystyrene beads deposited each pass. Since the raw data was not available, these graphs were reduced to determine the mean locations for particle deposits for each pass. The information is shown in Appendix III, along with the corresponding flight conditions for each run. The aircraft was equipped with one dispersal device mounted on each wing at equal spanwise locations. Hence, for each pass of the aircraft there was deposition on each side of the aircraft corresponding to the material released from the dispenser on that side. These dispensers were used to release the polystyrene beads. The release point of the beads, with respect to the trailing edge of the wing, was considered to be at coordinates POSIY and POSIZ, in accordance with the definitions of the model. For each pass of the aircraft over the collector arrays, the spanwise location of the dispenser was given.

The height of the aircraft, an Ayres Thrush Commander-800, was given as being the height of the spray boom over the collector array. Since the model requires as input the height of the trailing edge of the wing at the centerline, the distance from the spray boom to the wing trailing edge was added to the height recorded in Morris *et al.* (1984). As well, the spanwise location at which the height of the boom was measured was not given, so this was assumed to be the centerline of the aircraft.

Each pass of the aircraft was made over a set of three collector arrays, which collected the simulated droplets across the span of the aircraft. Then, for each pass of the aircraft, the mean location for deposited particles for each wing was determined for each collector array. For each particular pass, the aircraft height was not always the same over each of the three collector arrays, but in all cases, the height was the same over at least two of the arrays.

Therefore, only considering the cases with the same height of pass, the mean location of the particle deposits was determined by averaging the mean deposition location for each of those passes done at the same height.

The computer model incorporates the ability to simulate a ground surface that is uniformly sloping. The flight test facility used in the work of Morris *et al*. (1984) featured ground slope conditions such that the ground slope was about 2 percent upwards from the left to the right side of the aircraft. The droplet collector array, however, was in a horizontal plane (i.e. slope = 0). As a result the collector array virtually touched the ground at the end off the right side of the aircraft, and was roughly 1.3 meters from the ground at the end off the left side of the aircraft.

5.2 Method Of Testing Model Integrity

The program NASA77 was run for the range of initial spanwise vortex separations, b'_o , from 0.82 b to 1.00 b. The percent span initial vortex separation, as shown in the figures in this chapter, is given by ($\frac{b'_o}{b} \times 100$). Using the flight test conditions, as given in Appendix III, for each of the the 68 runs for each wing, giving 136 test data points, the final droplet location on the collector array was calculated and considered to be the 'predicted' values. Also, of these 136 test data points only 120 were useful due to significant uncertainty regarding the accuracy of 16 of them. The observation was made that these all corresponded to dispenser locations near the centerline. Hence, few of the remaining runs corresponded to dispenser locations near the centerline. Thus, unfortunately, the propeller influence would not show up as much as if more initial drop locations near the centerline were used. Again, note that the average location was taken from the three collector rows, as was mentioned earlier.

The flight test data were run through the model numerous times while varying a single input parameter each time. Then the predicted values were plotted against the actual values to determine the model accuracy. Next, a linear regression comparing the predicted with the actual results was done. From this the slope, intercept, and correlation coefficient for each data set were found corresponding to differing initial spanwise vortex separations.

5.2.1 Significance of slope, intercept, and correlation coefficient for plots of predicted vs actual drop locations

Determining the meaning of the results from the linear regression was the next concern. A linear regression produces the slope, intercept, and correlation coefficient of the best fit line through the data. For this work, the calculated slope and intercept show the correlation of the model predictions with the actual data. Perfect prediction of the actual drop locations would give a slope equal to 1.00 and an intercept equal to 0.0. Then, the value of the correlation coefficient, for the calculated slope and intercept, shows the goodness of fit of the data to that regression line. A correlation coefficient of 1.00 would indicate perfect correlation, for the calculated regression line. Hence, increasing scatter in the data would show up as a lower correlation coefficient.

5.2.1.1 the slope

The value of the slope of the best fit line indicates whether the model accurately predicts the lateral movement of the spray away from the centerline of deposition. The main parameter which should affect the slope of the line is the value of the total circulation. If the value of Γ_o is too high, then the droplets from both wings would experience excessive spanwise outwards movement, due to the over estimated induced air velocities. Likewise, if Γ_o is too low, the droplets would not move spanwise outwards as much. Hence, values of the slope greater than 1.00 should be indicators that Γ_o is over-predicted.

5.2.1.2 the intercept

The intercept of the line should give an idea of how the model predicts the shift of the entire spray deposition laterally to the left or right of the centerline. The propeller swirl and the crosswind are really the only two components which should cause a consistent shift of the droplets, independent of the side from which the droplets originate.

If the propeller swirl is modeled as being too great, the droplets should all tend to be shifted towards one side, dependent on the direction of propeller rotation. For the test aircraft, the propeller rotation is clockwise as seen by the pilot. Hence, the spray should be shifted towards the left if the propeller induced swirl is over estimated. A value of the swirl coefficient which is too large should then show up as a shift of the line of predicted versus actual drop locations giving an increasingly negative intercept for increasing swirl coefficients.

The crosswind model, if incorrect, could produce a shift of the droplets as well. As will be noted later, most of the crosswinds during flight testing were from the right side. If the crosswind is modeled as being too strong at positions below the height at which it was measured, then, for any particular pass of the aircraft, the drops will be shifted downwind, regardless from which side of the aircraft they were emitted. Thus, an over estimation of the crosswind at lower levels would be seen as a shift of all of the drops to the left, producing an intercept which is less than zero. Likewise, if the model under estimates the magnitude of the crosswind at lower levels, the intercept would be greater than zero.

5.2.1.3 the correlation coefficient

The correlation coefficient gives a measure of scatter in the data. For the calculated slope and intercept, the correlation coefficient gives the goodness of fit of the data to that line, regardless of whether the line shows accurate predictions.

5.2.2 Irregularities in the plots of correlation coefficient versus percent span initial vortex separation

Examination of Figures 5.1, 5.4, and 5.5 giving the plots of correlation coefficient versus percent span initial vortex separation reveals that the curves have local high and low regions, over the range shown. This problem of irregularity in the plots of correlation coefficient versus percent span initial vortex separation is most likely just a function of the data available. The flight tests were done with the dispenser for the polystyrene beads at discrete spanwise locations, those being 15, 25, 40, 50, 60, 70, 75, 80, 85, 90, 95 percent span. The model is only approximate in that it does not attempt to model vortex roll-up. The rsults of the model will be least accurate for drops released near the assumed initial vortex core positions. For the percent span initial vortex locations near the dispenser, the simulated drops would be nearer the core than when the percent span initial vortex locations was such that the dispenser was not exactly at the same percent span. As a result the core influence would be most significant for a few beads in the former case, possibly causing the irregularities. Likely, if the dispenser had been placed at a greater number of different spanwise locations, these irregularities would be smoothed out.

5.3 Parameter Testing And Model Verification

This section shows the results of evaluation of several of the parameters which were uncertain and, thus, could be varied in the model. They include the value of the total circulation generated by the wing, the value of the swirl coefficient of the propeller, and the value for the core coefficient of the trailing vortices. While varying each of these, the initial separation of trailing vortices was varied between 82 and 100 percent span, so as to determine the most reasonable initial separation. By this method, the most appropriate value for each parameter could be determined. Also, as already mentioned, flight test conditions for each pass of the aircraft are given in Appendix III.

The choice of the best value of each parameter was based on the correlation coefficient, slope and intercept of the best fit line from the linear regression. The correlation coefficient alone was not sufficient to choose the best value for each parameter. As will be shown, over a range of values for each parameter being varied, the correlation coefficient did not change substantially. Hence, the best value, for the parameter being varied, was chosen as that which gave a slope nearest to 1.00 and an intercept nearest to 0.00, while also giving a high correlation coefficient (near 0.88).

5.3.1 Value of total circulation

The value for the total circulation, Γ_o, generated by the wing was varied in the test model. Five different values of Γ_o were used ranging from 0.75*((Γ_o^{ell} + Γ_o^{rect})/2) to 1.25*((Γ_o^{ell} + Γ_o^{rect})/2). This corresponds to a range of values for circulation which brackets the values for elliptical loading, rectangular loading, and the average of those two. The wing on the test aircraft had an aspect ratio, A_R, of 5.5, hence, the value for the total circulation should be approximately the average of the values for the rectangular and elliptical loading, as was explained in Chapter 3. The values of $\Gamma_o = \Gamma_o^{ell}$ and Γ_o^{rect} effectively bracket the average value, (Γ_o^{ell} + Γ_o^{rect})/2, since they are 10.7% higher and lower values, respectively, hence an even greater margin is covered in this test.

These tests were done with the swirl coefficient held constant at 0.0040 and the vortex core coefficient constant at 0.0775. These were reasonable values, as seen by later tests.

Figure 5.1 shows the plot of the correlation coefficient versus the percent span initial vortex separation. From that plot the low values of circulation are seen to give better correlation coefficients at lower values of the initial vortex separation, whereas for high values of circulation the best correlation coefficients appear at greater initial vortex separation. As well, it can be seen in Figure 5.1 that the actual best correlation coefficient is not significantly different with differing vortex strengths.

The fact that a good correlation was possible with lower values for Γ_o than predicted by the "free flight" value of (Γ_o^{ell} + Γ_o^{rect})/2, suggests the possibility of incomplete roll up of the vortices during the time for spray movement. However, it should be remembered that even though the vortex sheet takes a finite time to roll up, the total vorticity shed by the wing must remain constant during that time. The phenomenon of vortex roll up is not entirely understood and will not be dealt with any further in this study.

Another possible explanation for why the lower Γ_o values give better correlation at closer spacing of the trailing vortices is as follows. With the trailing vortices closer together, more drops are emitted from locations spanwise outwards of the trailing vortices. This results in more drops being subject to the upward induced velocity field due to the vortices, hence, giving more time for drop movement laterally outwards.

Considering Figures 5.1 to 5.3 together shows that the value of Γ_o = (Γ_o^{ell} + Γ_o^{rect})/2 provides a very reasonable value for the correlation coefficient (about 0.88) at an initial spanwise separation of the trailing vortices of about 94% (actually through the range 92 to 97%). This is significant since that initial vortex spacing also gives a slope very near 1.0, which would indicate accurate correlation of the predicted drop locations with those from the flight tests. Figure 5.3 shows the correlation of the intercept with the vortex separation, and that for the above value of Γ_o the intercept is about 0.4. This means that the average predicted values are about 0.4m greater than the flight test values.

Although the value for Γ_o cannot be determined precisely according to these tests, it is readily apparent that the value of Γ_o = (Γ_o^{ell} + Γ_o^{rect})/2 does provide very reasonable results. Hence, it is this value that is used for the next chapter concerning the application of the model to hypothetical spray configurations.

The comments in the section regarding the swirl coefficient effects should be noted as well, since they lend more credibility to the total circulation being the average of the value for rectangular and elliptical loading.

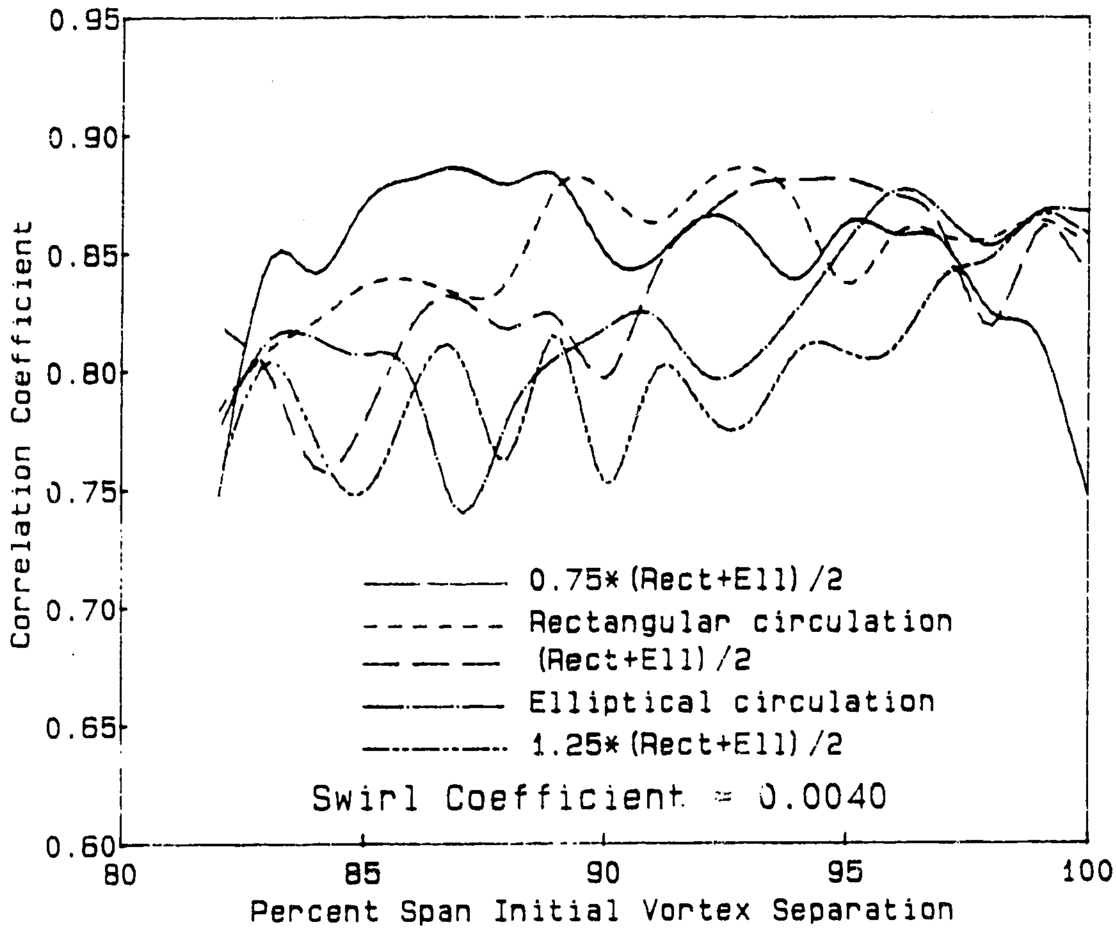


Figure 5.1 Correlation Coefficient with Different Values of $\Gamma_{\circ}$

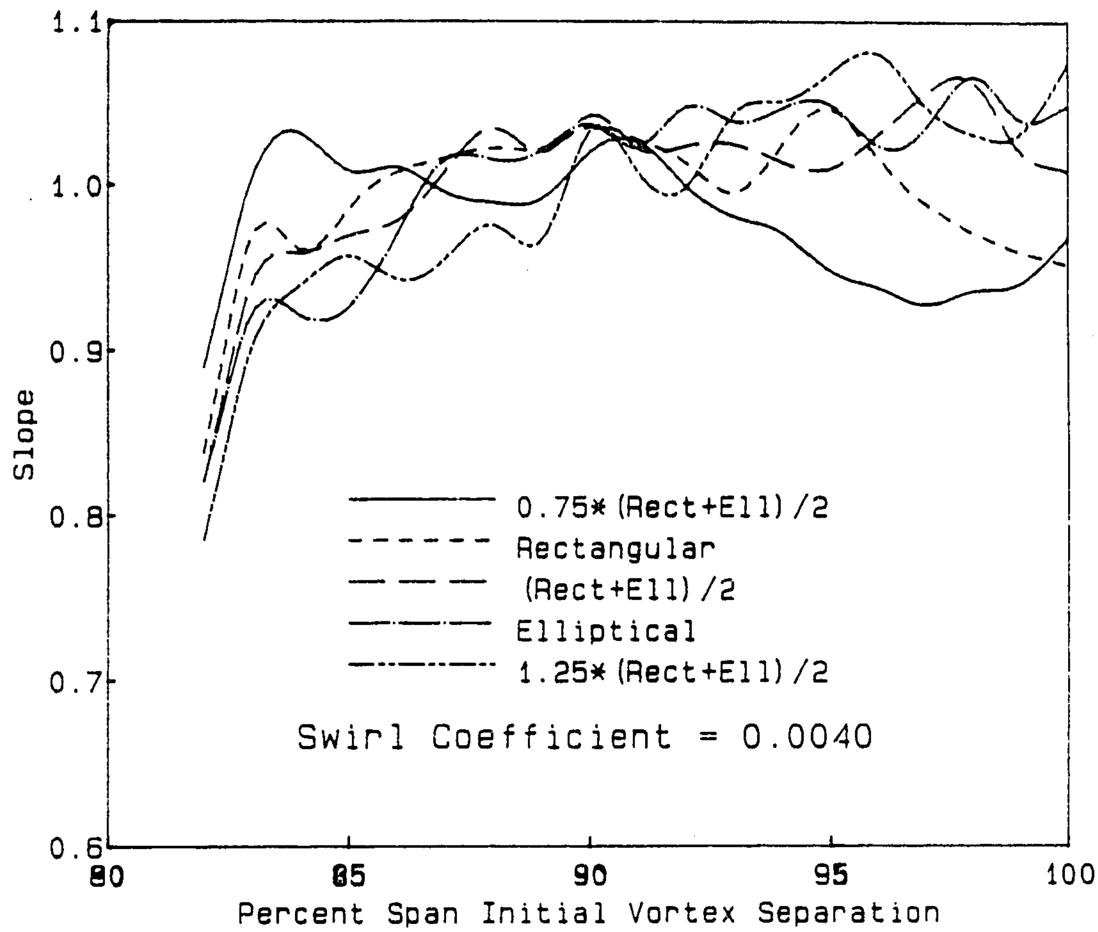


Figure 5.2 Slope with Different Values of Γ_o

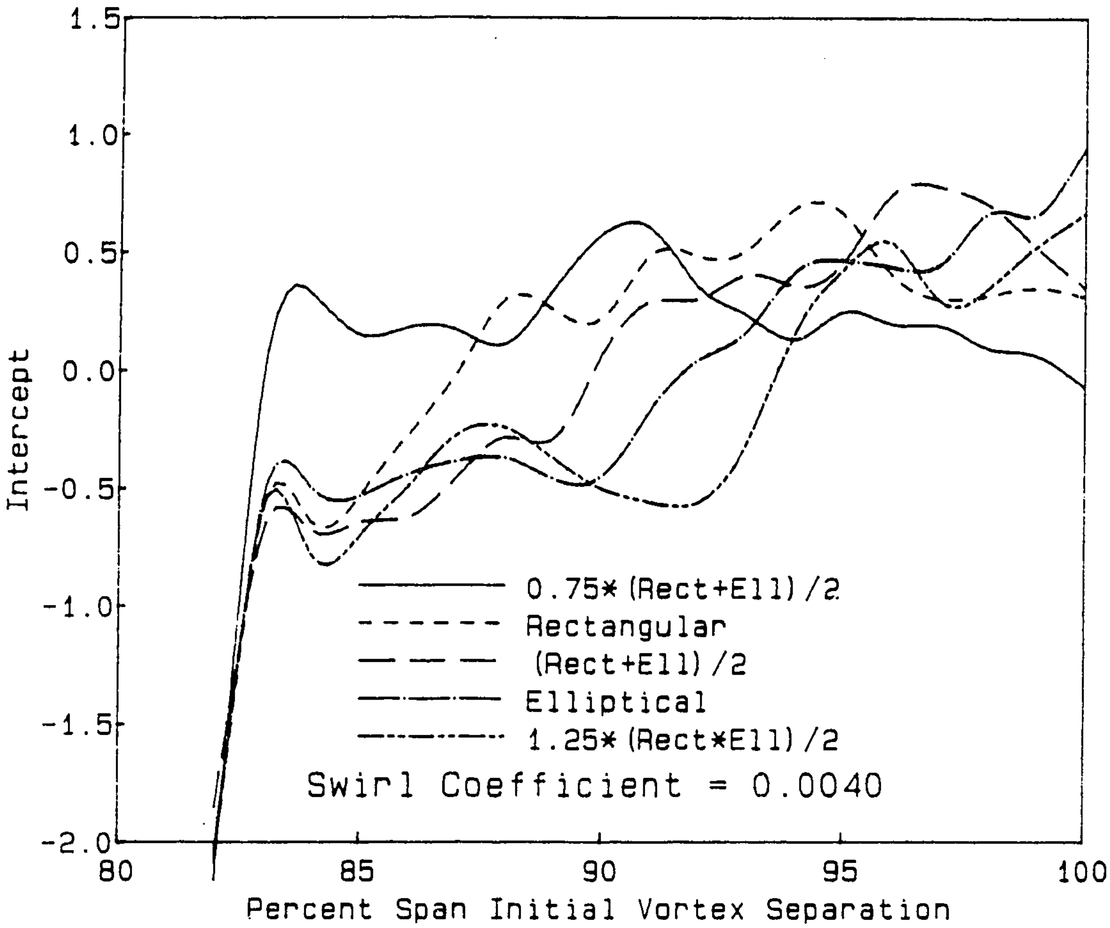


Figure 5.3 Intercept with Different Values of Γ_o

5.3.2 Effects of viscous vortex core diameter

Three different vortex core coefficients (0.018, 0.036, and 0.0775) were used in the tests to determine the most reasonable core size. The need for this was due to the uncertainty as to the exact value which should be used, as has been discussed at length in section 3.9. The swirl coefficient used for these tests was 0.0040, which, as will be shown later, is a reasonable value. Also, the value of the circulation is chosen as the average of the elliptical and rectangular loading values, which has been shown to be reasonable. The effect of changing the vortex core coefficient is shown in Figure 5.4.

The results indicate that there are no drops which pass closer than 0.036 *b* from the vortex axis. This arises from the observation that vortex core coefficients of 0.018 and 0.036 produce identical results. Figure 5.4 displays this fact by effectively showing only two lines. The small variation in the correlation coefficient with differing core coefficient values, indicates that the core coefficient value is not as crucial as expected. The core coefficient value of 0.0775 is seen to give just slightly better results than the other values. This could be because the larger core radius brings the maximum tangential velocity more in line with that predicted by Roberts (1983) even though the actual radius at which it occurs is over-predicted.

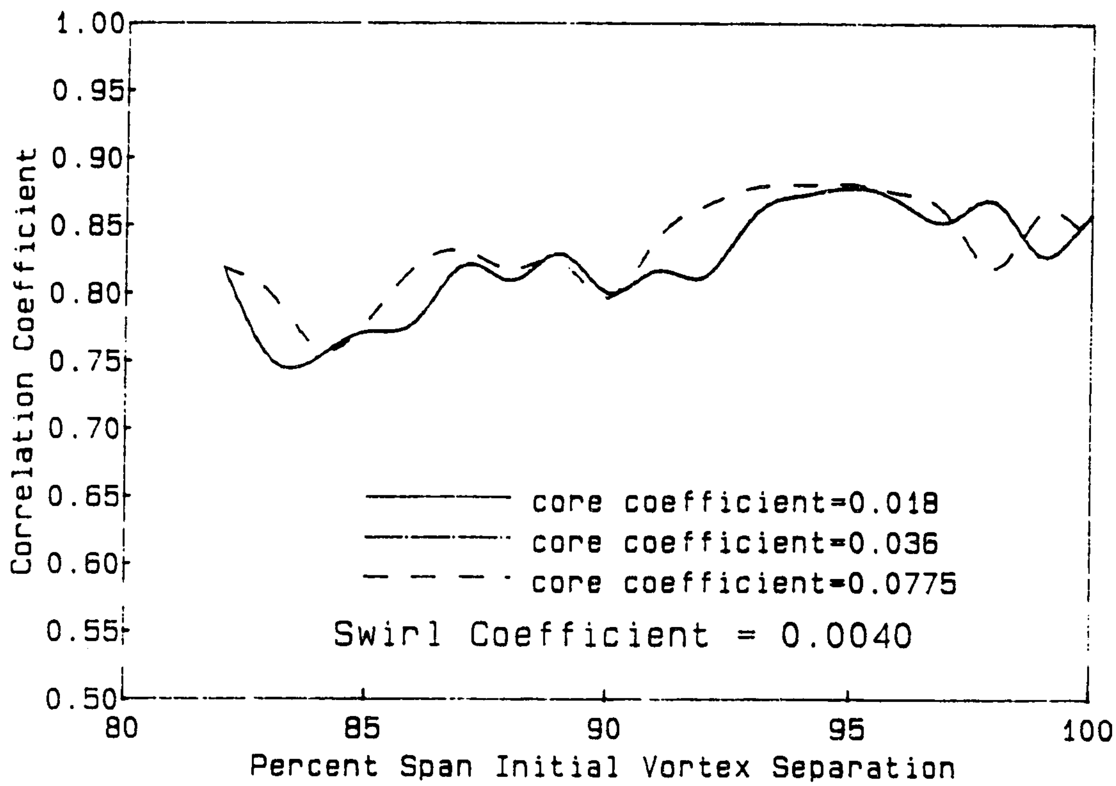


Figure 5.4 Effect of Viscous Vortex Core Diameter on Correlation Coefficient

Also, the trend should be noted that the best correlation for any core size occurs for the range of initial vortex separation from about 92 to 97% span.

5.3.3 Effects of swirl coefficient

The value for a swirl coefficient is a variable in the program. The correct value to use was in question since, for the flight test conditions, insufficient information was available to determine this. The value of the swirl coefficient was varied in the model between 0 and 0.010 in order to determine the most suitable value. For these tests the core coefficient was maintained equal to 0.0775, and the circulation was, again, the average of the elliptical and rectangular loading values.

Upon examining the test results (Figures 5.5 to 5.7) it is readily apparent that higher values for the swirl coefficient (such as 0.0075) significantly reduced the degree of correlation of the model with the flight test data. A plot for a swirl coefficient of 0.010 is not shown since the correlation coefficient for this value was significantly lower over the entire range. According to the results of the model, the use of a swirl coefficient much lower than expected leads to very good correlation of the model results with the flight test data.

Examination of Figure 5.5 shows that the best value of the correlation coefficient does not change appreciably with changes in swirl coefficient, at least at the lower values.

The value of 0.0035 for a swirl coefficient seems most appropriate when examining Figures 5.6 and 5.7. At that value the slope of the plot of Predicted vs Actual value is very nearly 1.00. The value of the intercept is above, but near, zero as should be the case. Figure 5.5 shows that, for higher swirl coefficients, there is an increase in the slope, while the lower swirl coefficients show the opposite trend.

The coincidence of the best correlation coefficient at the same percentage span as the most accurate fit, in terms of slope and intercept, lends more credence to the model. Thus, a reasonable swirl coefficient which is about 0.0035 to 0.0040 gives the best results. For this value the slope is about 1.01 and intercept is about 0.4 with a correlation coefficient of 0.88. It should be understood that the precise value for the swirl coefficient would be greatly dependent on the exact operation conditions of the aircraft for each flight.

The low values of the swirl coefficients which are most reasonable lend credence to the suspicion that the aircraft fuselage, wings, and tail section all contribute

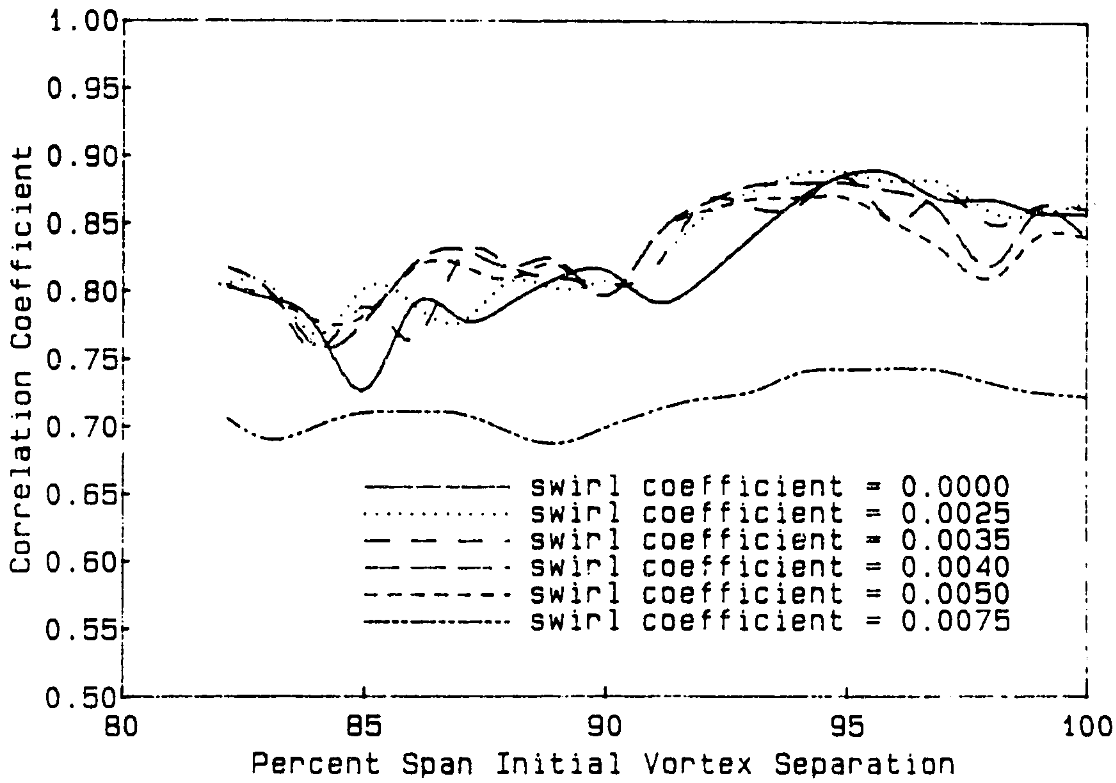


Figure 5.5 Effect of Swirl Coefficient on Correlation Coefficient

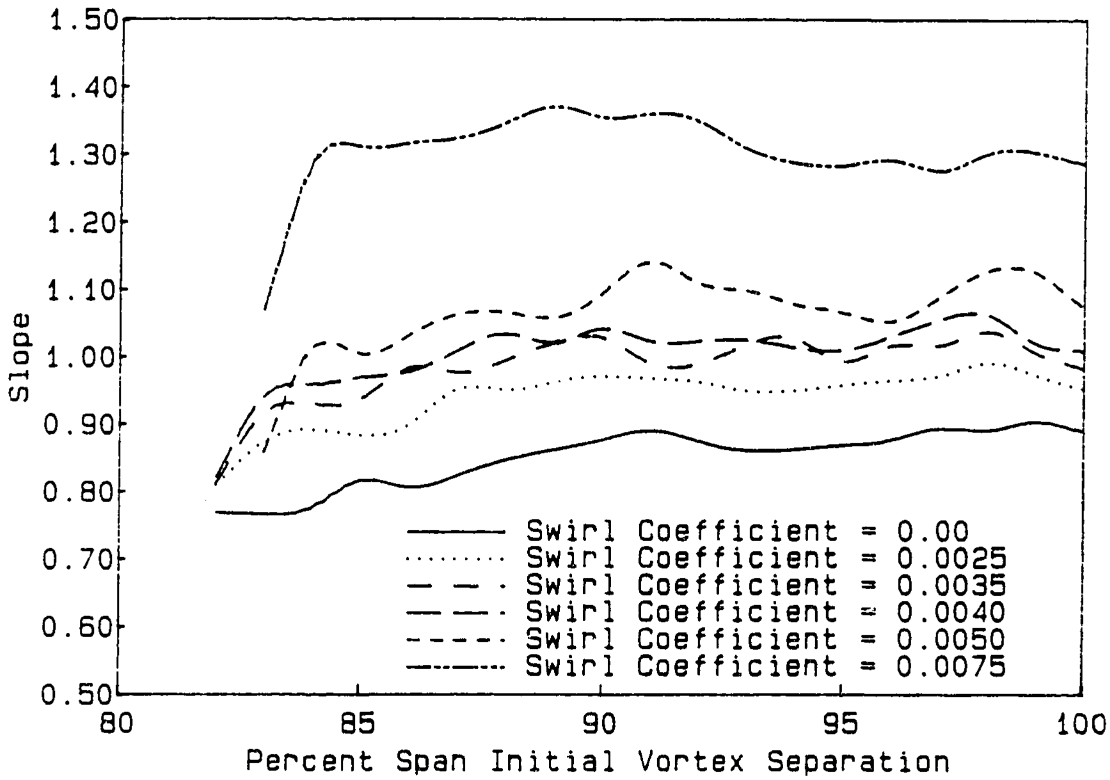


Figure 5.6 Effect of Swirl Coefficient on Slope

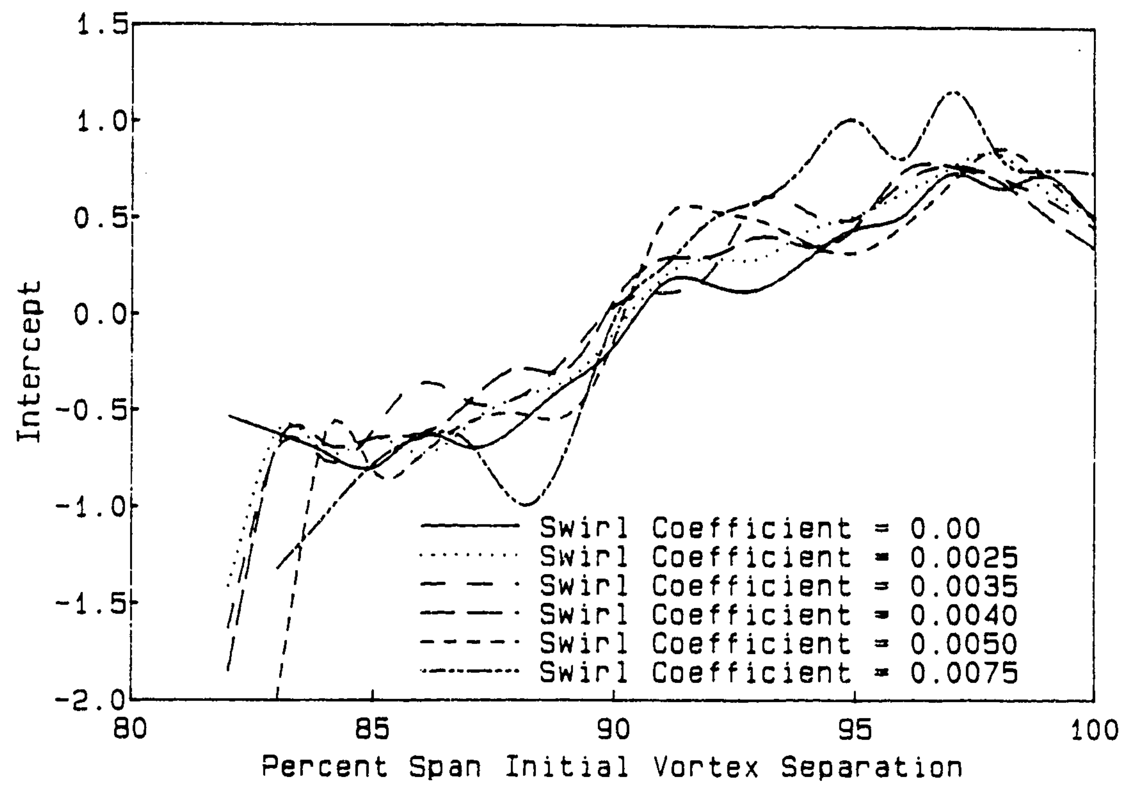


Figure 5.7 Effect of Swirl Coefficient on Intercept

significantly to the reduction of the swirl induced by the propeller. Thus, for a complete aircraft, the propeller swirl coefficient is far different from that predicted by propeller theory for an unconfined propeller.

Again, the trend should be noted that the best correlation for any swirl coefficient occurs for the range of initial vortex separation from about 92 to 97% span.

Also, notice should be given to the quality of the correlation for swirl coefficients equal 0.0035 to 0.0040, since for this test the value of circulation was set equal to the average of the elliptical and rectangular loadings. The high degree of correlation shown, lends greater credibility to the reasonableness of this value for the circulation.

5.3.4 Initial spanwise vortex separation

The proper spanwise location at which to attach the trailing vortices was unknown. Consequently the model was run with the vortices initially attached at one percent increments along the span, between 82 and 100 percent spanwise separation. Afterwards the best-fitting spanwise location was used as input to the WAKE77 model.

A scheme was used to adjust the initial vortex separation b_o' such that at height $h = b$ the initial vortex separation was 82% b. When $h = 0$ the initial separation is that which is the input value. Between these two values, a linear interpolation was used to determine the initial separation based on the height of the aircraft. Since the model gave acceptable results with this scheme, there is reason to expect that it is quite reasonable. The value of 82% initial separation enters because the aircraft had a rectangular wing with $A_R \approx 5.5$, as explained in Chapter 3.

The correlation coefficient versus the initial spanwise vortex separations for different values of circulation and core coefficient are plotted in Figures 5.1 and 5.4, respectively. From these plots the value for initial spanwise vortex separation can be seen to be most reasonable in the range 92 to 97%. Also, from Figure 5.5, it is apparent that, for different values of the swirl coefficient, the most reasonable initial spanwise vortex separation is in the same range.

Since the initial spanwise vortex separation which is input is that for the aircraft at h = 0, the value at most operating heights is between the input value and 82%. From the flight test data, it can be seen that most passes were at about 4 m height (one third span). As a result, those predictions would have been done with initial vortex separations of about 90 to 91%. This result now confirms that the constant value, of 90% span initial vortex separation, used by Trayford and Welch (1977) was a reasonable value to use.

5.3.5 Crosswind effects

The crosswind model used is that used by Morris *et al.* (1984), as has been noted. Closer examination of the results shows a consistent positive intercept for the regression analysis of the predicted versus actual drop locations. This seems to indicate that there may be an error in the crosswind model.

Examination of the flight test data, indicates that the crosswind was most often from the right. In fact, 112 of 120 (94%) of the data points which are used to do the regression, have crosswinds from the right. Hence, if the wind were stronger at heights lower than the height at which the crosswind was measured, the intercept would shift left, towards zero, the desired value.

5.3.6 Secondary vortex formation

The results show no support for the formation of a secondary vortex at ground level in the time for spray movement during the flight tests, after the manner described by Bilanin *et al.* (1978).

As noted earlier, the model used does not incorporate the viscosity of the air at the ground plane. Thus, the development of the secondary vortex has not been simulated. With the formation of a secondary vortex, which induces vortex bounce, there would be a change in the air velocity so as to increase the upward and spanwise outward components. The times which are required for drop movement in spray applications, using all but very fine sprays, are relatively short in comparison to that for the formation of the secondary vortex. The model adequately predicts the drop movement without considering this secondary vortex effect. Hence, this supports the contention that such a viscous effect is only relevant when concerned with a longer time frame, such as that required to model the aging process of the vortex system.

5.4 Computer Time Used For Model Runs

The computer simulation was done on one of the University of Alberta mainframe central processing units, an Amdahl 5870 mainframe.

In model verification about 10 minutes of CPU time were used to run a typical data set containing 136 drops from various spanwise locations, while varying the initial spanwise vortex separation from 82 to 100 percent. Thus, 2584 drop trajectories were simulated, 43% of which were 650 micron diameter and 57% of which were 327 micron diameter, representing 300 and 150 micron diameter droplets respectively. The smaller droplet sizes used more computer time than the large droplets, since the time increments were smaller and, generally, the droplets remained airborne for longer time periods

The model had been written to run on an IBM PC, using Compiled Basic, as well. The original intent was that the program could be run on a PC, hence, making it accessible to the commercial spray applicator. That idea was dispelled, once the time to run a data set was determined to be near 1 week.

6. MODEL APPLICATION

6.1 Method Of Applying The Model

The application of the computer program is really the expected end of the development process. Hence, the following section gives some samples of the results which the WAKE77 program can be expected to provide.

For all tests, the results show the coefficient of variation for a range of swath widths from 10m to 50m. This range adequately collects the spray deposition. The maximum width between passes of the aircraft, also referred to as swaths, which can be utilized without significant increase in the coefficient of variation is an important result. This follows since wider swaths require fewer passes of the aircraft to cover a given area. Hence, both the magnitude of the coefficient of variation and the width of swath for the given magnitude are significant results.

6.1.1 Test conditions for model application

The tests were all run using the same basic aircraft as that used for the testing and verification of the computer model. The basic values used for the aircraft and operating conditions are those given in the example files, AIRCRAFT and DROPINFO, shown in Appendix I. The parameters which are varied in each particular test configuration are given in the figures accompanying each test. Those which are not explicitly mentioned have been set to the default values as in the example files.

The conditions, delineated as follows, applied for each of the sample cases. All of the tests were run with adjacent passes done in the same direction. The crosswinds were assumed to be zero, and the wet bulb depression was set to zero. Only 100°, flat fan nozzles were used and these were oriented straight down (i.e., 90° from the horizontal). Also, for all tests, except one for which droplet diameter was the variable, the droplet diameter was maintained at 300μm.

From the results outlined in sections 5.3.1 and 5.3.4, the following conditions were found to be reasonable. The value of 94% *b* for the initial vortex separation was found to yield satisfactory results when used in conjuncion with the scheme to modify the value with height, and so is used for these tests. Also, as was seen from the tests, the value of the total circulation,
$\Gamma_o = ((\Gamma_o^{ell} + \Gamma_o^{rect})/2)$ provided satisfactory results and, hence, is continued for these applications.

6.1.2 Number Of Drops Per Nozzle Used For The Tests

For the following sample applications, the same number of droplets were used per test, so as to have a consistent basis with which to work. For each pass of the aircraft, 960 drops were emitted, corresponding to 40 drops per nozzle for each of the 12 nozzles on each side of the aircraft. The number was chosen so as to provide consistent results while using moderate amounts of computer running time. The number of drops needed to give consistent results was not known exactly, so additional runs were made using both 20 and 80 drops per nozzle, to check the consistency of results. Figure 6.1 shows the change in the number coefficient of variation with swath width, for different numbers of drops per nozzle. The number coefficient of variation is the coefficient of variation based on the number of drops at each location, without considering the volume of each drop. Since all drops are the same diameter for this test, the coefficient of variation based on the volume would give the same result. Figure 6.1 appears to show only two lines, but closer examination reveals that the lower line is the result of the overlapping of that for 40 and 80 drops per nozzle. Hence, when 40 drops were used the results were consistent with the results with 80 drops per nozzle. However, there is some deviation with only 20 drops per nozzle. From this result, the use of 40 drops per nozzle was judged to be sufficient.

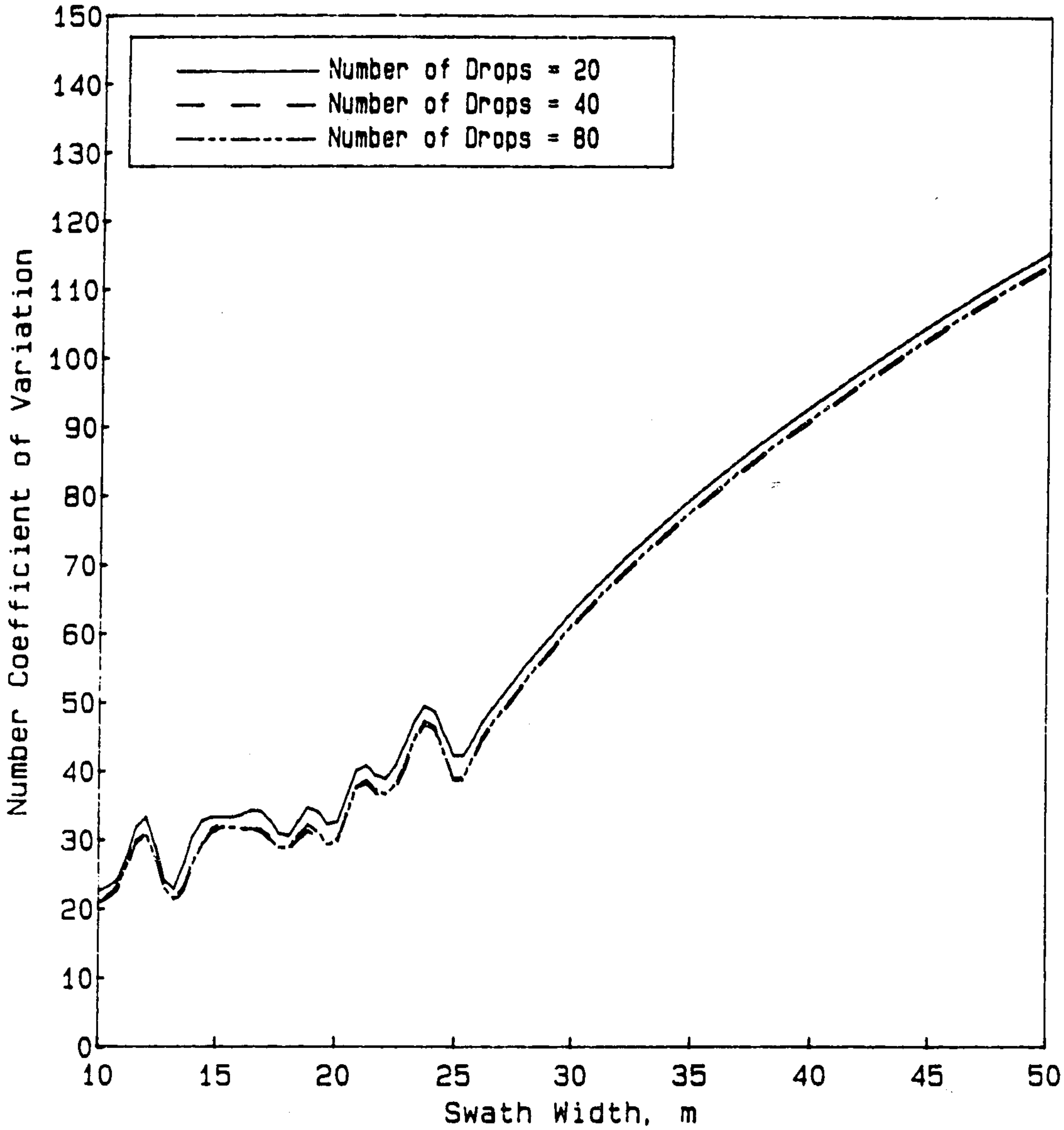


Figure 6.1 Effect of Number of Drops per Nozzle on the Coefficient of Variation

6.2 Wet Bulb Depression Effects

The model can show the effects of evaporation on the spray distribution for the first 20 seconds. The model stops at that time since either the drop would be so small as to be of no consequence and is considered lost, or the drop would have drifted out of the spray target area. Thus, the rate of evaporation can be tested to check its effect on the spray distribution.

The wet bulb depression, ΔT, was set to 10°C for one test and 0°C for the other. The initial drop diameter was 300μm for both tests. As is apparent from Figure 6.2 there is no significant difference between the coefficient of variation for either wet bulb depression. The result is expected since, recalling Figure 2.3, the droplife at ΔT = 10°C, for a 300μm droplet is about 45 seconds, with little change in diameter within the first few seconds. Most of the drops would have hit the ground within two seconds, so the movement is not far removed from the case of ΔT = 0°C. From the results of the test, the model predicted that 94% of the initial volume of spray was deposited with ΔT = 10°C

Using much smaller drops would render significantly different results, since the movement time of a 50μm droplet would be large relative to its drop life time. Both the movement and the amount of spray deposited would be considerably different for such a small drop. However, no test was done to actually show this.

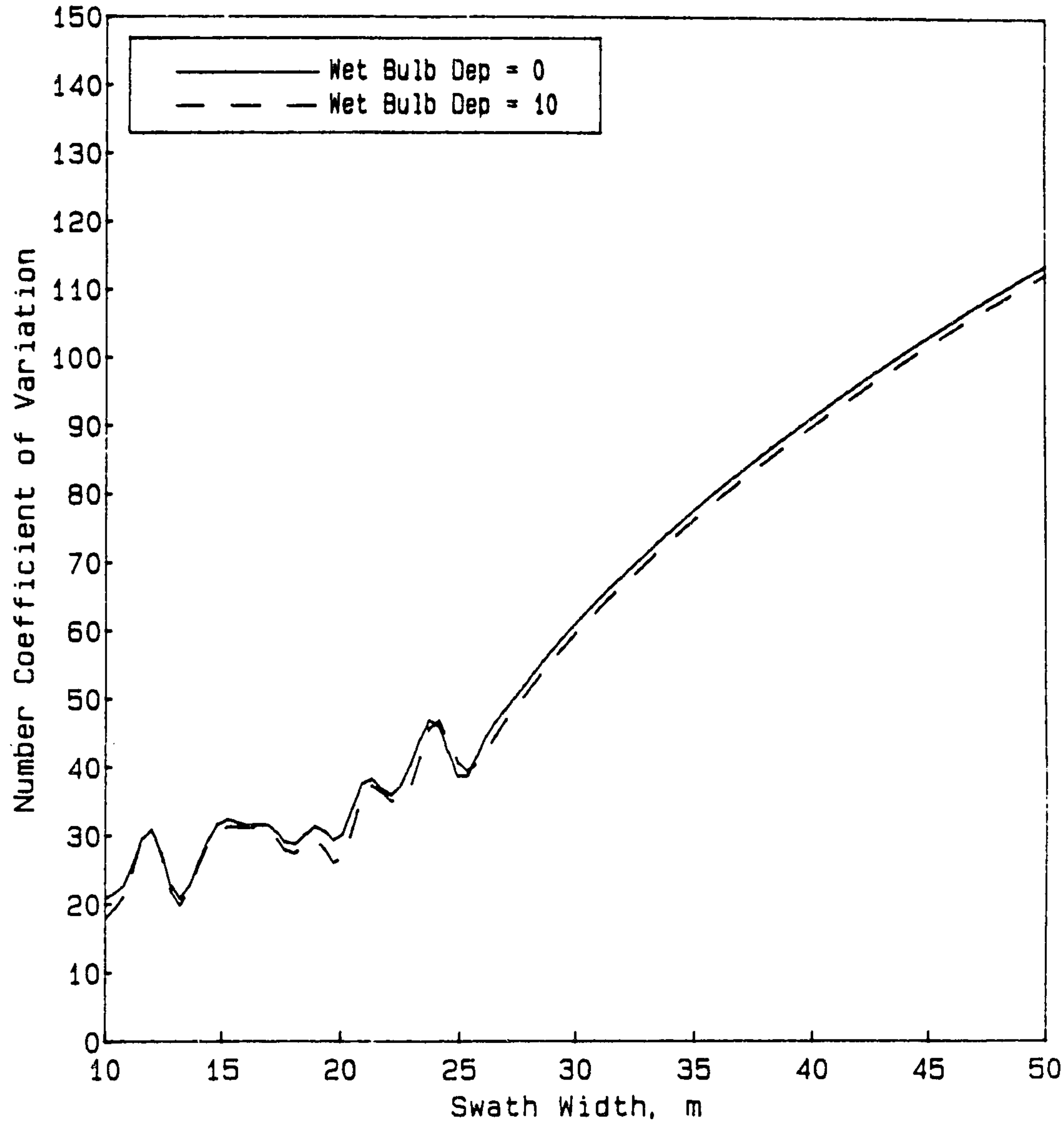


Figure 6.2 Effect of Wet Bulb Depression on the Coefficient of Variation

6.3 Effect of Spanwise Extent of Nozzles

The spanwise extent of a spray boom, which is the extent to which the nozzles can be located, is a significant factor to consider when configuring any spray plane. The following shows the significance of proper spray boom lengths, assuming that the nozzles are placed to the end of the boom. Three different spanwise extents were considered, all starting at 5% span from the centerline and extending to 80, 90, and 100% span.

Figure 6.3 shows the change of the coefficient of variation with swath width. It is apparent that the 100% span boom gives the poorest coefficient of variation over the swath widths of concern. The 90% span boom shows the best performance. The 80% span boom gives very good performance at lower swath widths, but shows a rapid increase in the coefficient of variation with swaths wider than 17m. The reason that the 80% span boom has a narrower usable swath width is probably linked to the size of the drops considered, along with the 90° downward nozzle orientation, since the initial momentum of the 300μm drops can carry them close to the ground before reaching terminal velocity. The coefficient of variation increases rapidly at swath widths greater that 17m, for the 80% span boom, because then the spray from adjacent passes no longer completely overlaps.

Figures 6.4 to 6.6 show the deposition patterns for the various spray boom lengths. A phenomenon which appears vividly in Figure 6.4, is a significant spike of deposition at the edges of the deposition pattern for a 100% span boom. This is as a result of the spray outwards from the trailing vortex being caught up in the intense flow of the vortex and being carried upwards and spanwise inwards before finally being deposited on the ground. The 90% span boom shows no sign of this occurence in Figure 6.5, nor is it apparent for the 80% span boom in Figure 6.6.

The width of swath can be seen to be affected significantly by the boom span as well. The 80% span boom only gives a usable swath width of 17m, while the 90% span boom gives a 20m swath width for the same coefficient of variation. For this configuration, it can be seen that the 80% span boom provides a narrow a spray pattern, while the 90% span boom shows up as the best choice for the configuration used.

The effect of the boom length was shown to be independent of the airspeed. Figures 6.3 to 6.6 are for an airspeed of 61.77 m/s (120 knots), while Figures 6.7 to 6.9 show similar effects of spray boom length at an airspeed of 45 m/s.

These results indicate that there will be an optimum boom length which should be used for spray application over the entire range of normal operating airspeeds.

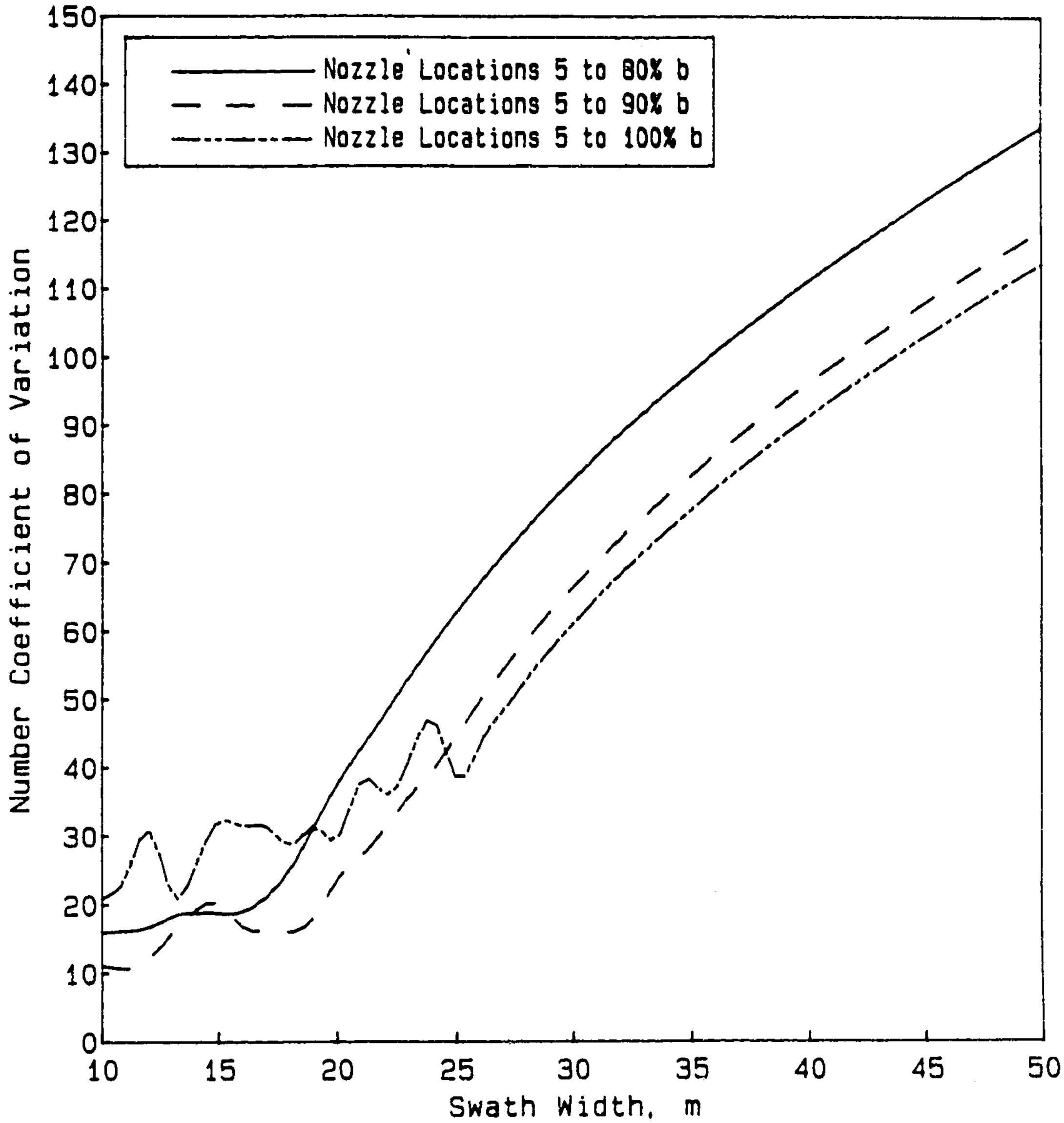


Figure 6.3 Effect of Nozzle Spanwise Location on the Coefficient of Variation, at 61.77 m/s

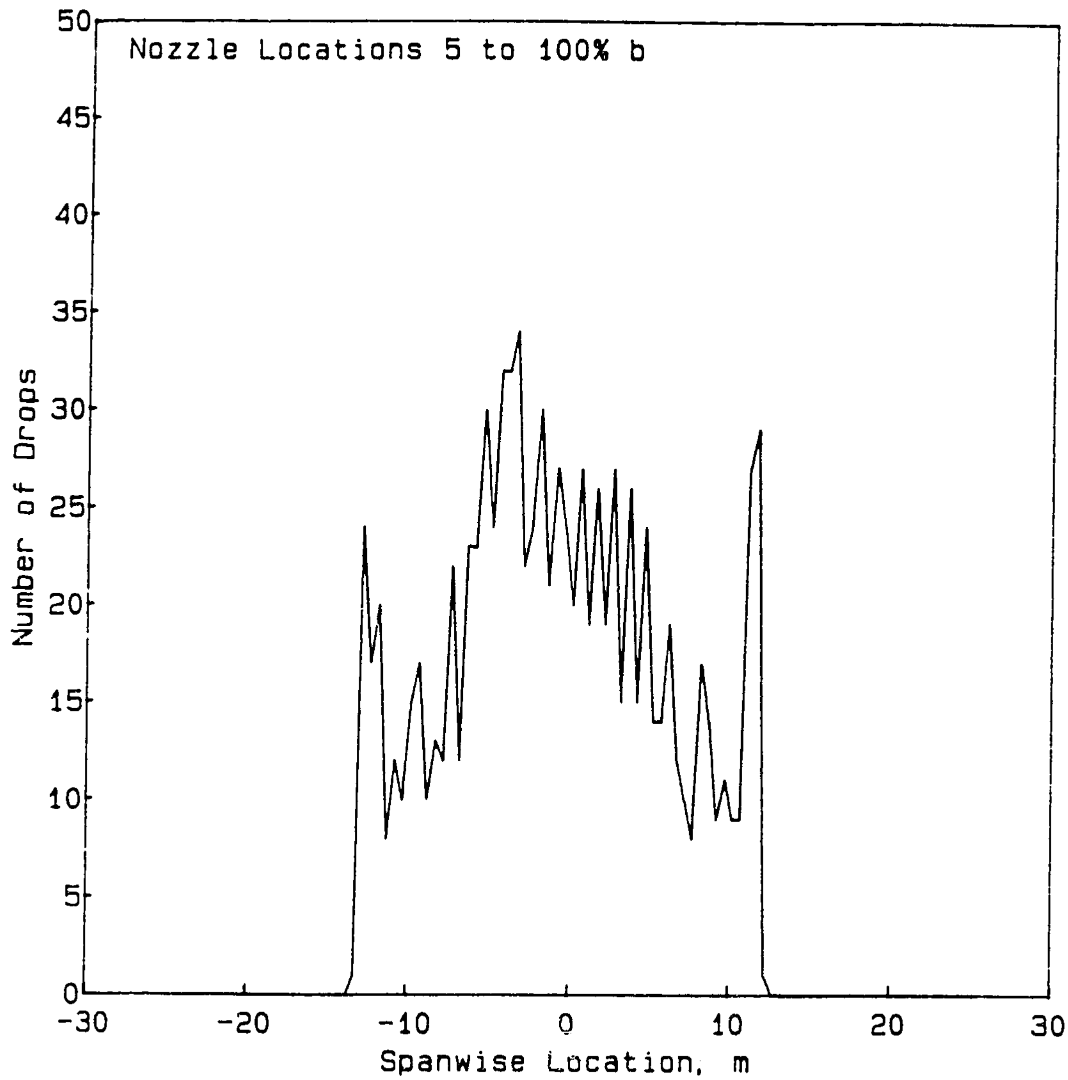


Figure 6.4 Deposition Pattern for 5 to 100% *b* spray boom

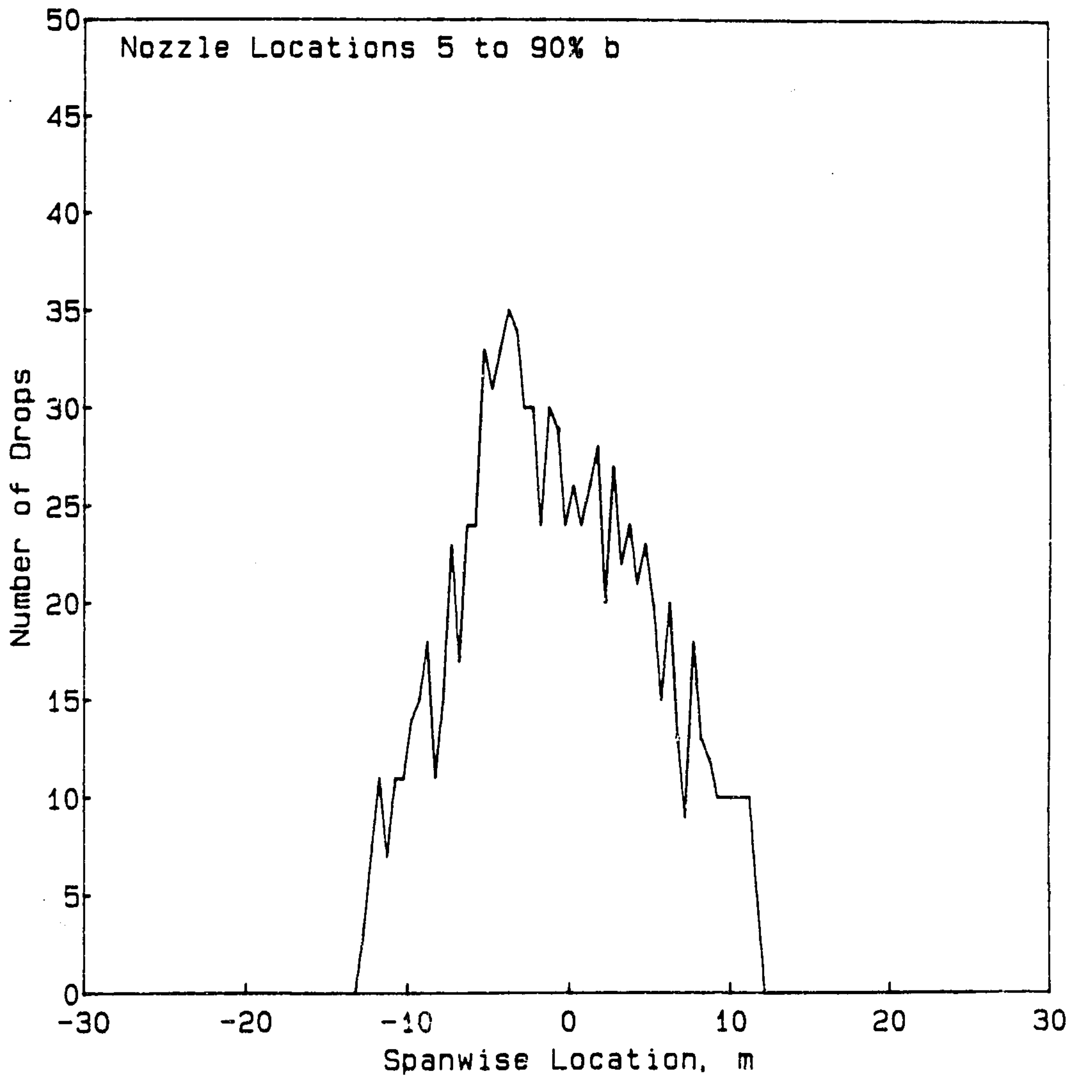


Figure 6.5 Deposition Pattern for 5 to 90% *b* spray boom

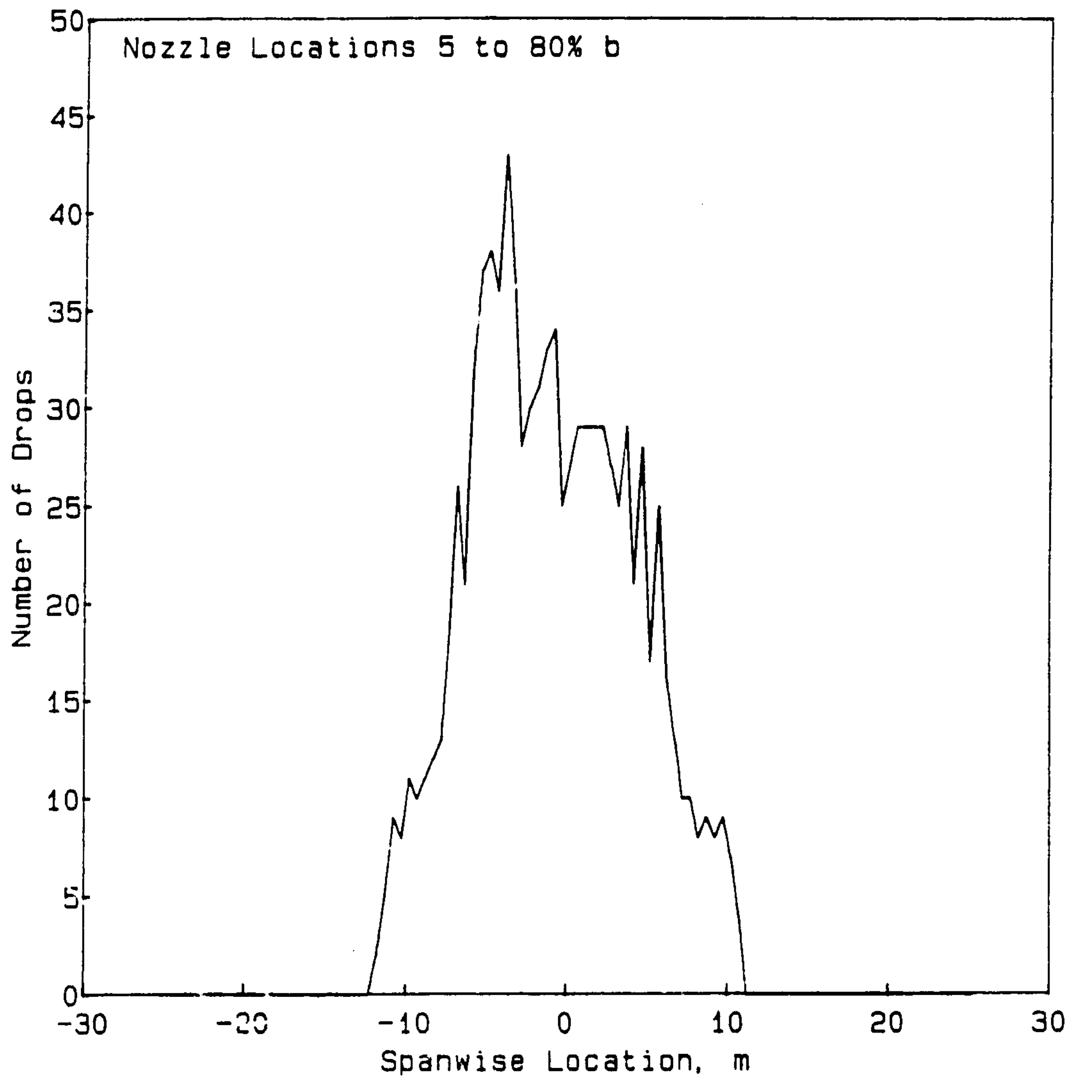


Figure 6.6 Deposition Pattern for 5 to 80% *b* spray boom

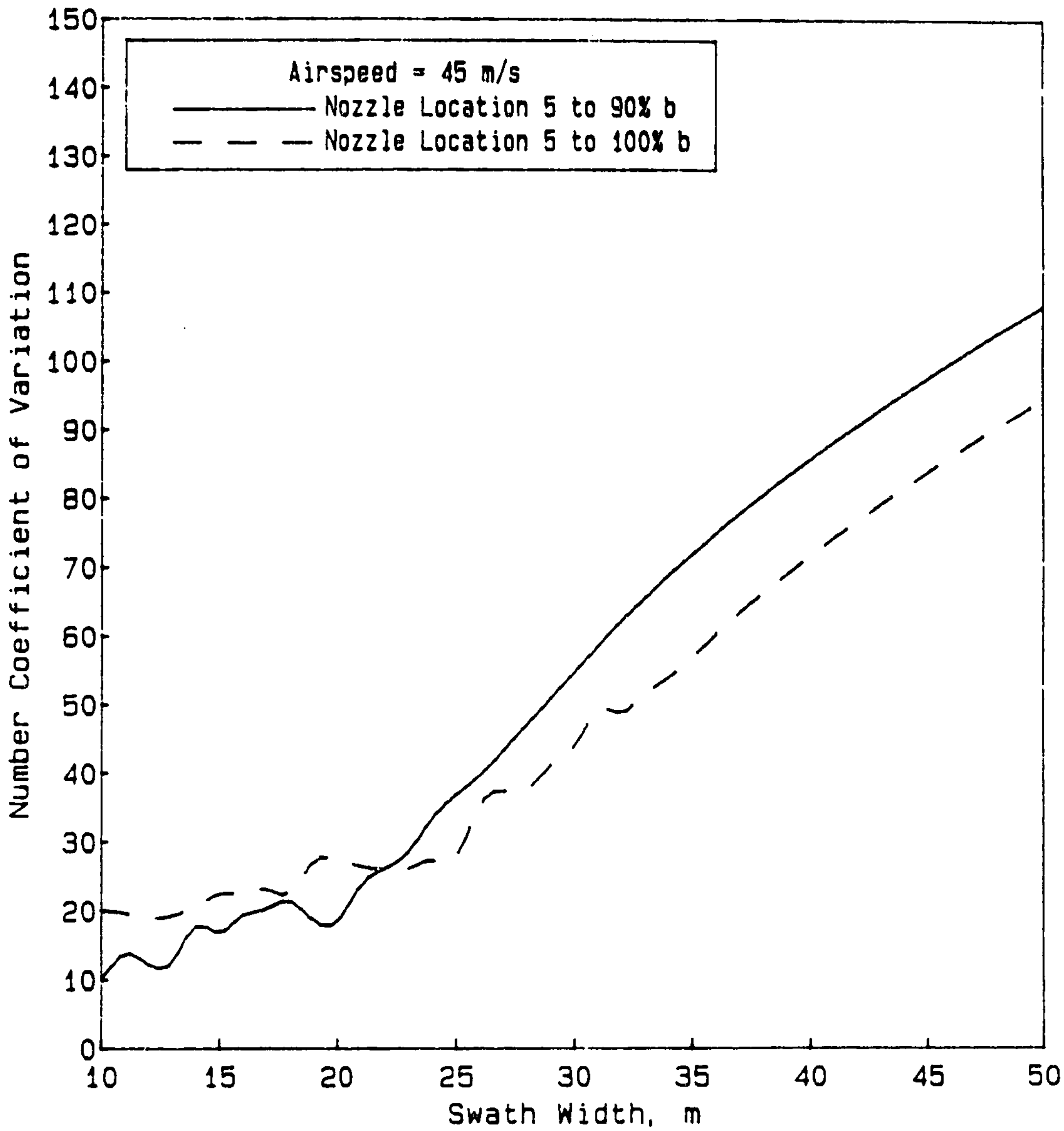


Figure 6.7 Effect of Nozzle Spanwise Location on the Coefficient of Variation, at 45 m/s

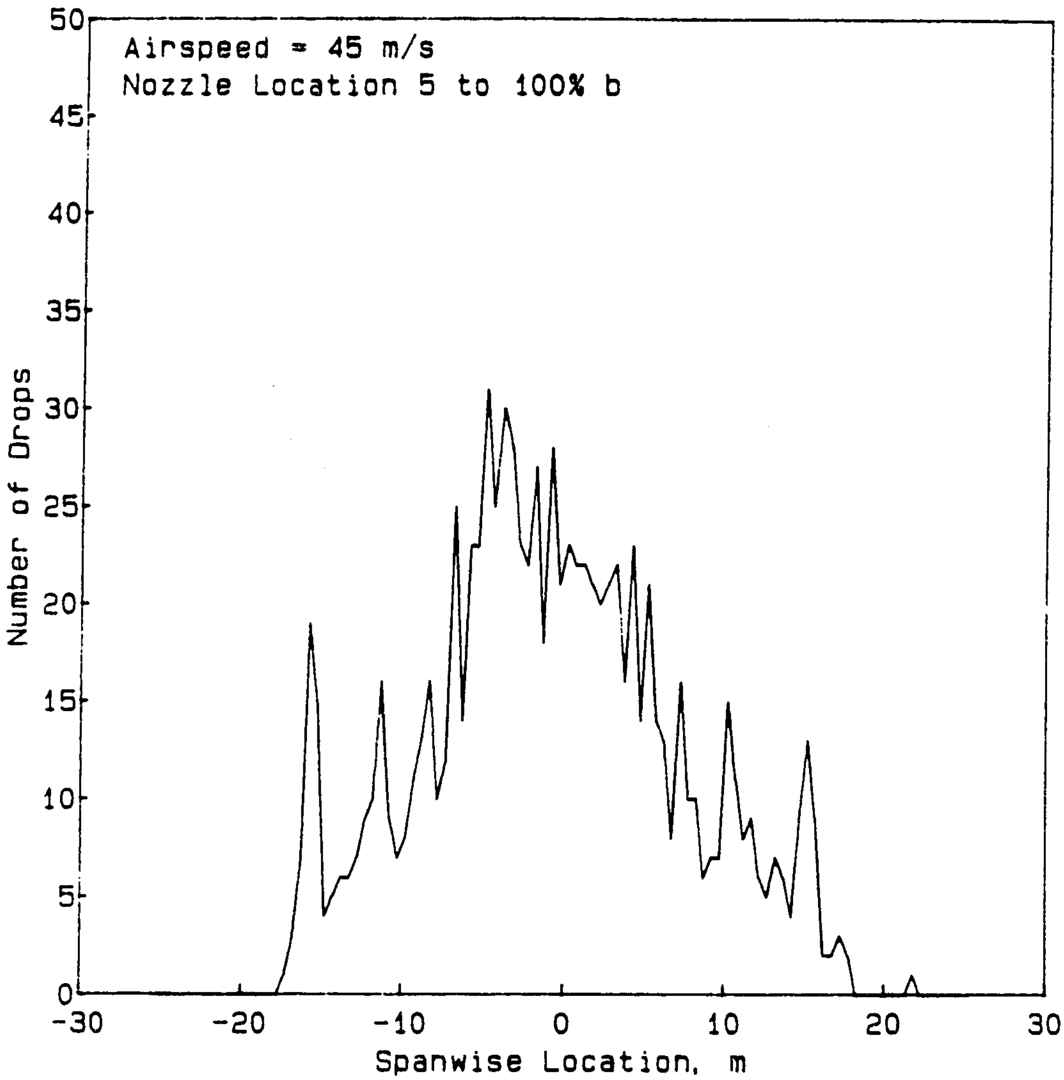


Figure 6.8 Deposition Pattern for 5 to 100% *b* spray boom

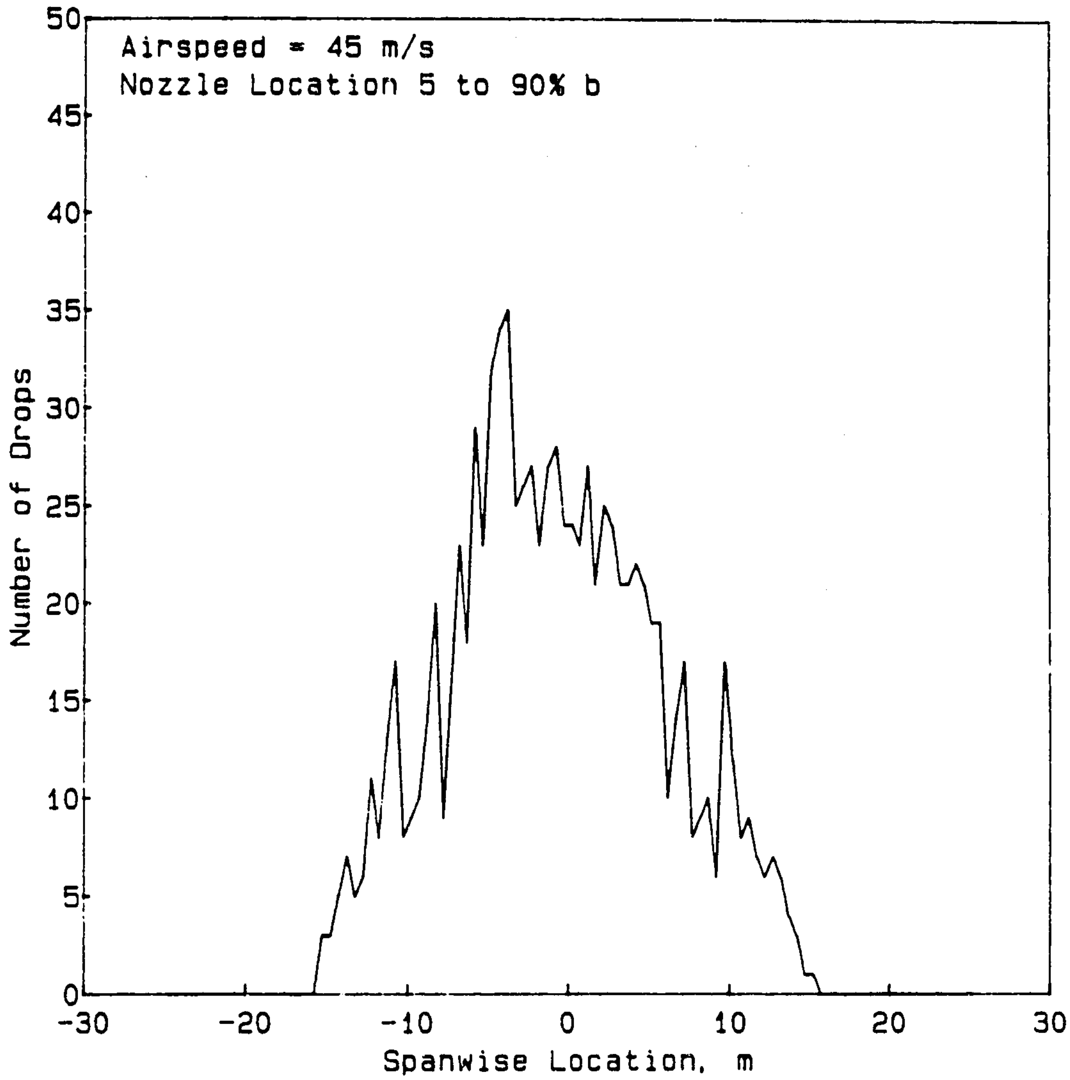


Figure 6.9 Deposition Pattern for 5 to 90% *b* spray boom

6.4 Effect of Number of Nozzles Along the Spray Boom

The number of nozzles spaced along the spray boom was varied in order to determine the effects on the spray distribution. The same total number of drops were emitted for each test. Two configurations were checked, one with the standard 12 nozzles per semispan and 40 drops per nozzle, and a second with 24 nozzles and 20 drops per nozzle. Both configurations were such that the extent of the boom was from 5 to 90% span.

Figure 6.10 shows the coefficient of variation versus the swath width for both configurations. A third configuration was added to show the comparison with a boom from 5 to 100% span using the 12 nozzles per semispan and 40 drops per nozzle.

Figure 6.10 shows the rather curious effect, that the configuration with fewer nozzles gave better uniformity for swath widths up to 22m. The 24 nozzle configuration did, however, give the expected marked improvement in uniformity for wider swaths. However, the expectation would be that a greater number of nozzles would lead to greater uniformity for all swath widths. The exact reason for this discrepency is not apparent until Figures 6.11 and 6.12 are compared. Figure 6.11 shows that with the 24 nozzle arrangement the spike of deposition near the edge of the spray distribution reappears (as with the boom from 5 to 100% span). Hence, the pattern is further from the ideal trapezoidal pattern which would result in the best uniformity. Figure 6.12 shows the 12 nozzle configuration to result in a pattern closer to trapezoidal, with no sign of the edge spikes. Figure 6.11 shows that the 24 nozzles from 5 to 90% span do improve slightly the uniformity of the spray distribution as compared to the 12 nozzles from 5 to 100% span.

The results show that, in order to optimize the spray distribution uniformity, the number of nozzles along the span of the boom needs to be considered in conjunction with the swath width to be used for the application.

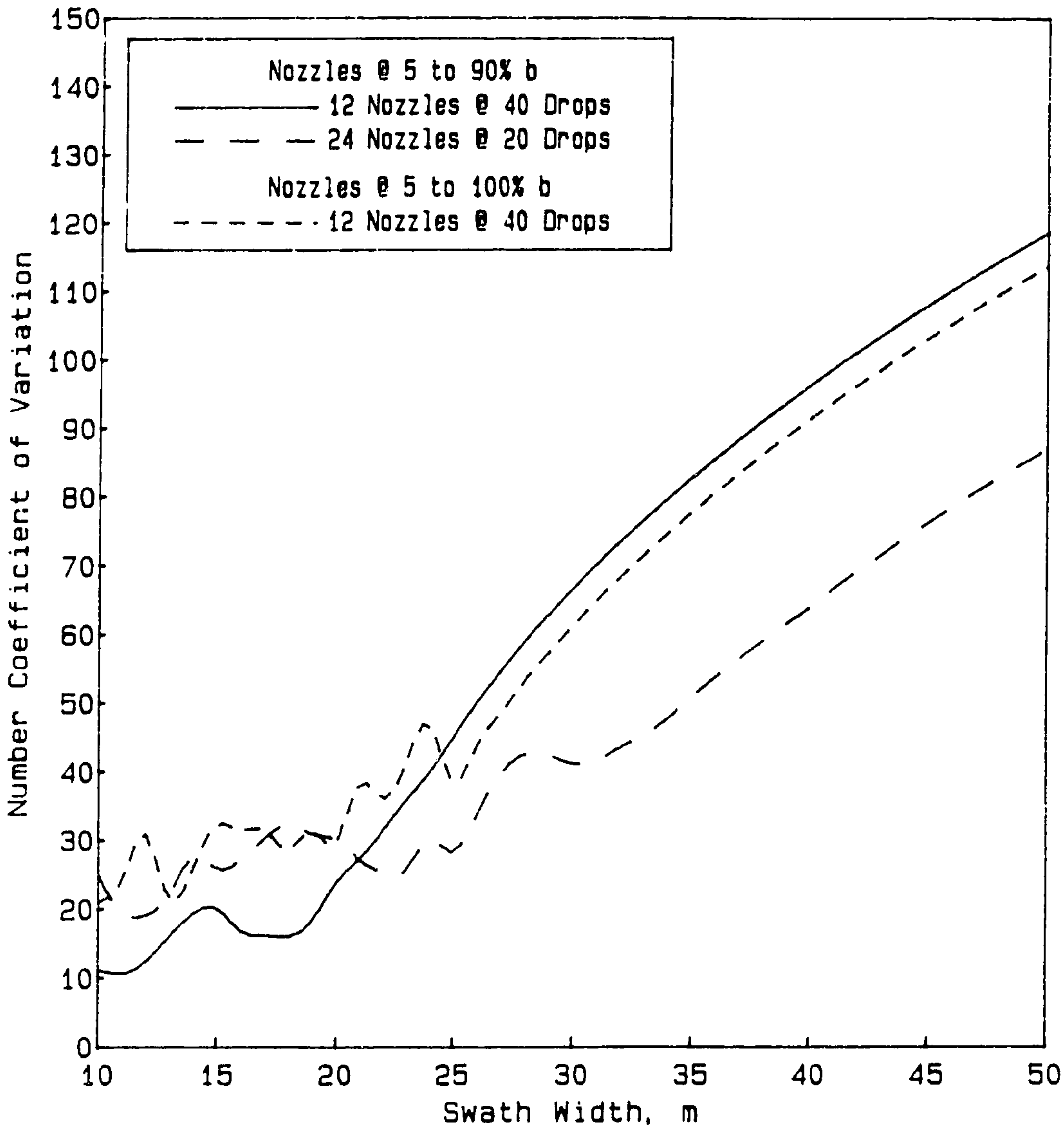


Figure 6.10 Effect of Number of Nozzles Along the Spray Boom

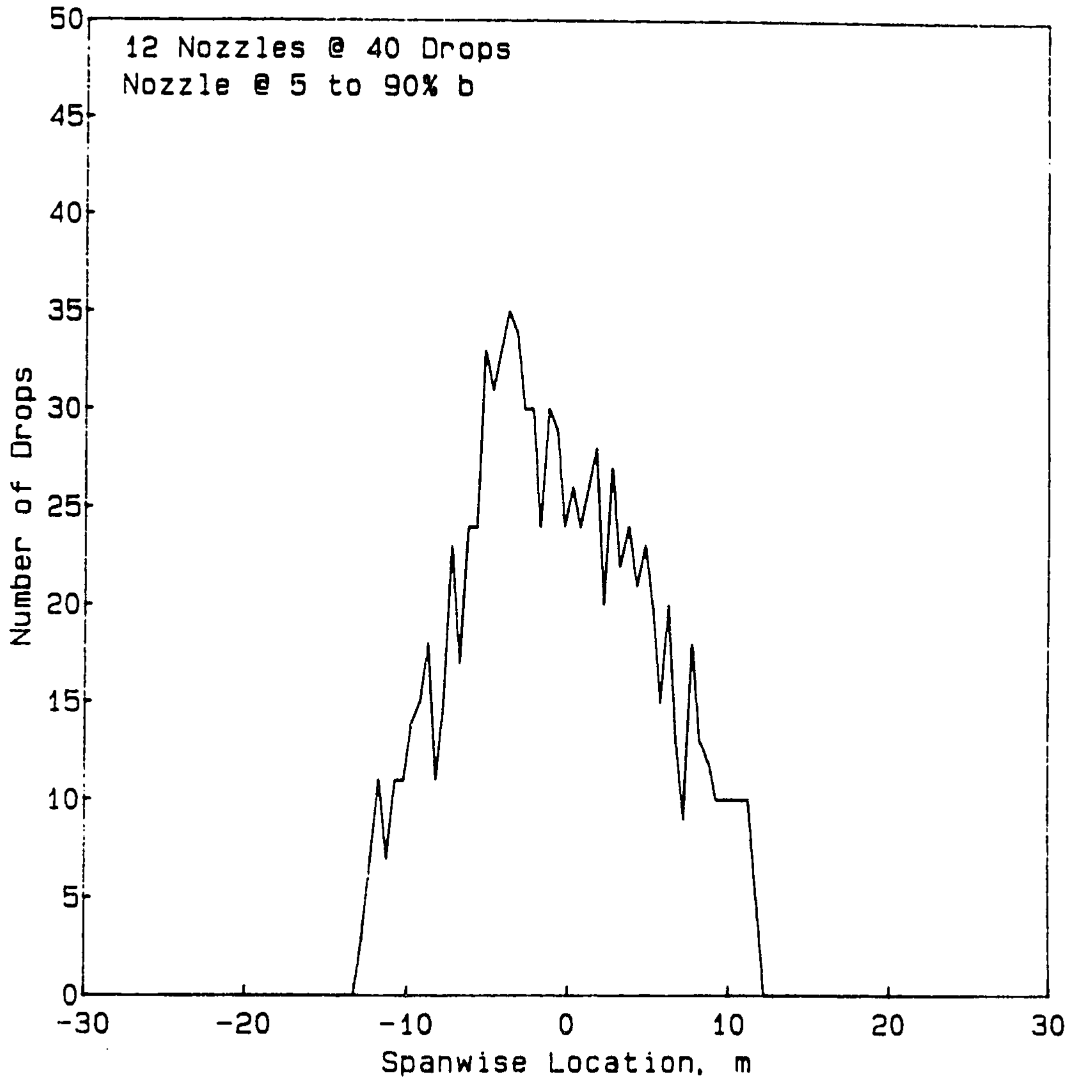


Figure 6.11 Deposition Pattern for 12 nozzles along boom

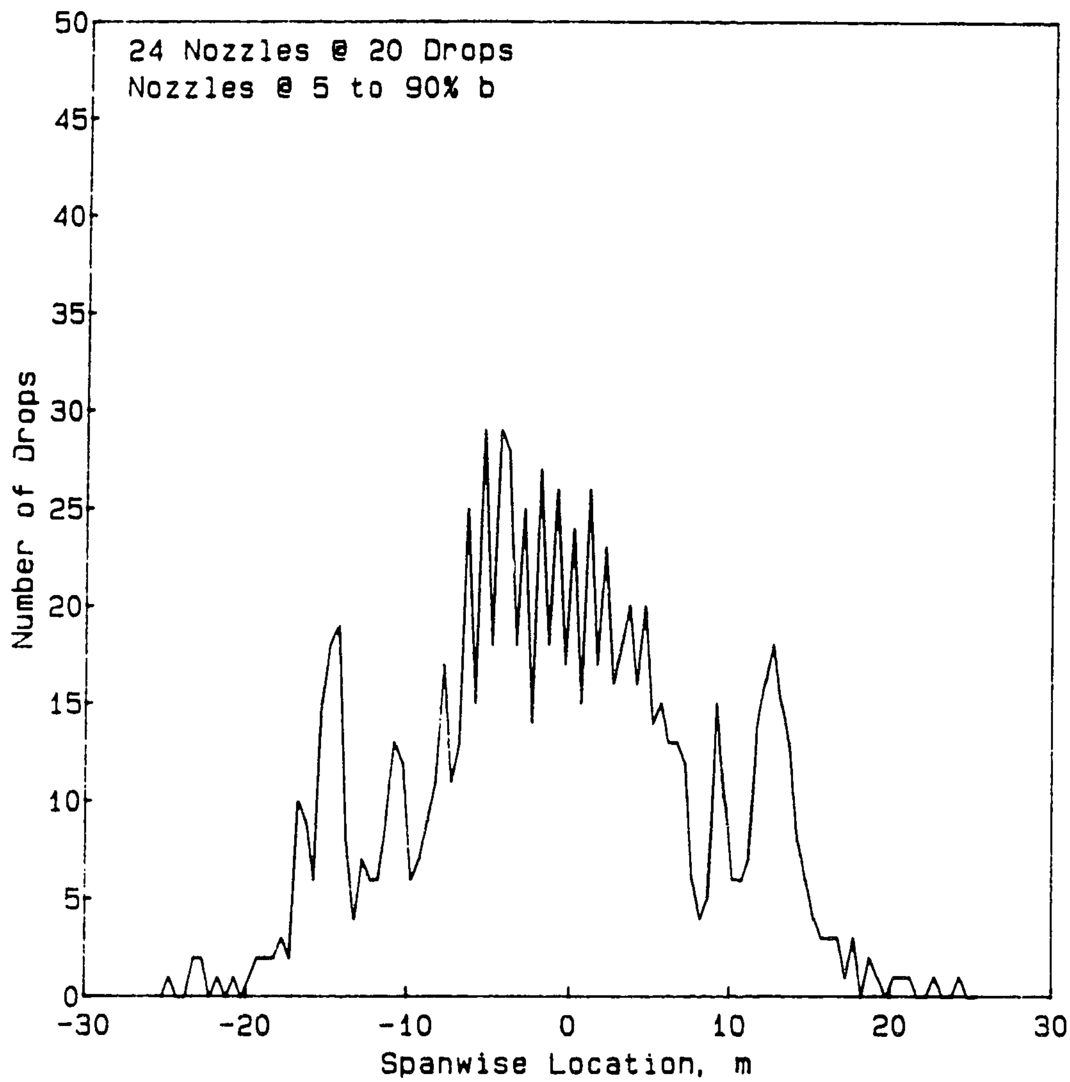


Figure 6.12 Deposition Pattern for 24 nozzles along boom

6.5 Droplet Diameter Effects

Figure 6.13 shows the effect of various drop diameters. As can be seen, the uniformity with the 300μm drops is considerably better than for the other two smaller diameters. Considering the test conditions, this result follows logically. The aircraft is only 3.0 m above the ground surface and the nozzles are oriented 90° downwards. For the larger drop diameters the initial spray velocity will have a significant role, moving the spray downwards more than for the smaller drops which, as a result, would be affected much more by the trailing vortices and induced propeller swirl. Since the nozzles are evenly spaced across the span, the spray can be expected to be most evenly deposited for the larger droplets when the aircraft is this close to the ground.

The phenomena of the spikes of deposition at the edge of the swath pattern also show up readily for all drop sizes considered, as seen in Figures 6.14 to 6.16. All tests were conducted with the nozzle placement extending to 100% span. The spikes are most pronounced for the 200μm drops. They are significant for the 150μm drops as well, but because of the increased spanwise movement of the spray and, hence, decreased deposition per unit width of swath, the spikes are less pronounced than for the 200μm droplets.

The spanwise movement of the spray is seen to be considerably increased for the smaller drops, since they are most susceptible to entrainment into the trailing vortices. The smaller droplets yield improved uniformity for wider swaths. However, the magnitude of the coefficient of variation is considerably larger at greater swath widths and, hence, renders them relatively unsatisfactory.

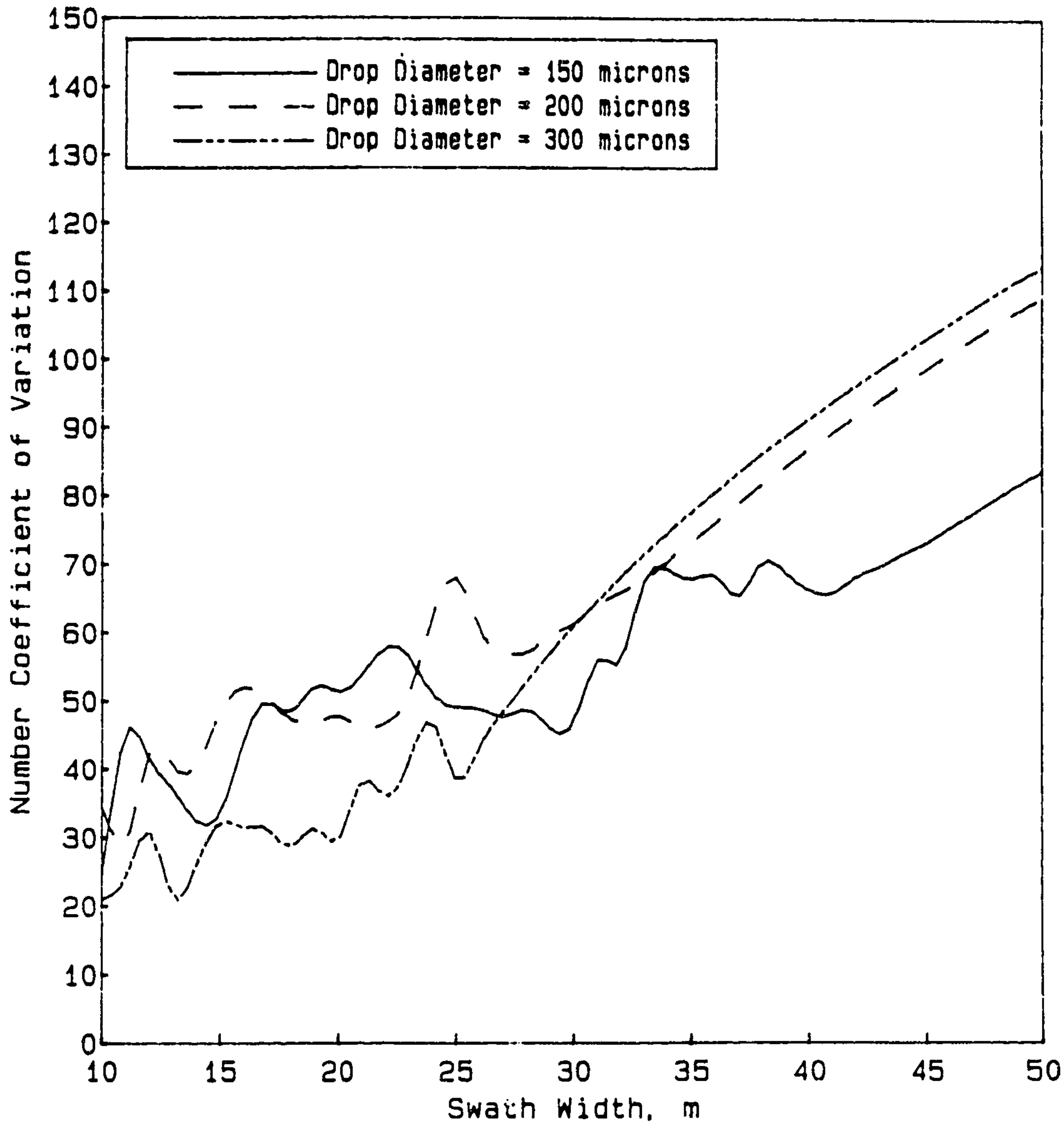


Figure 6.13 Effect of Drop Diameter on the Coefficient of Variation

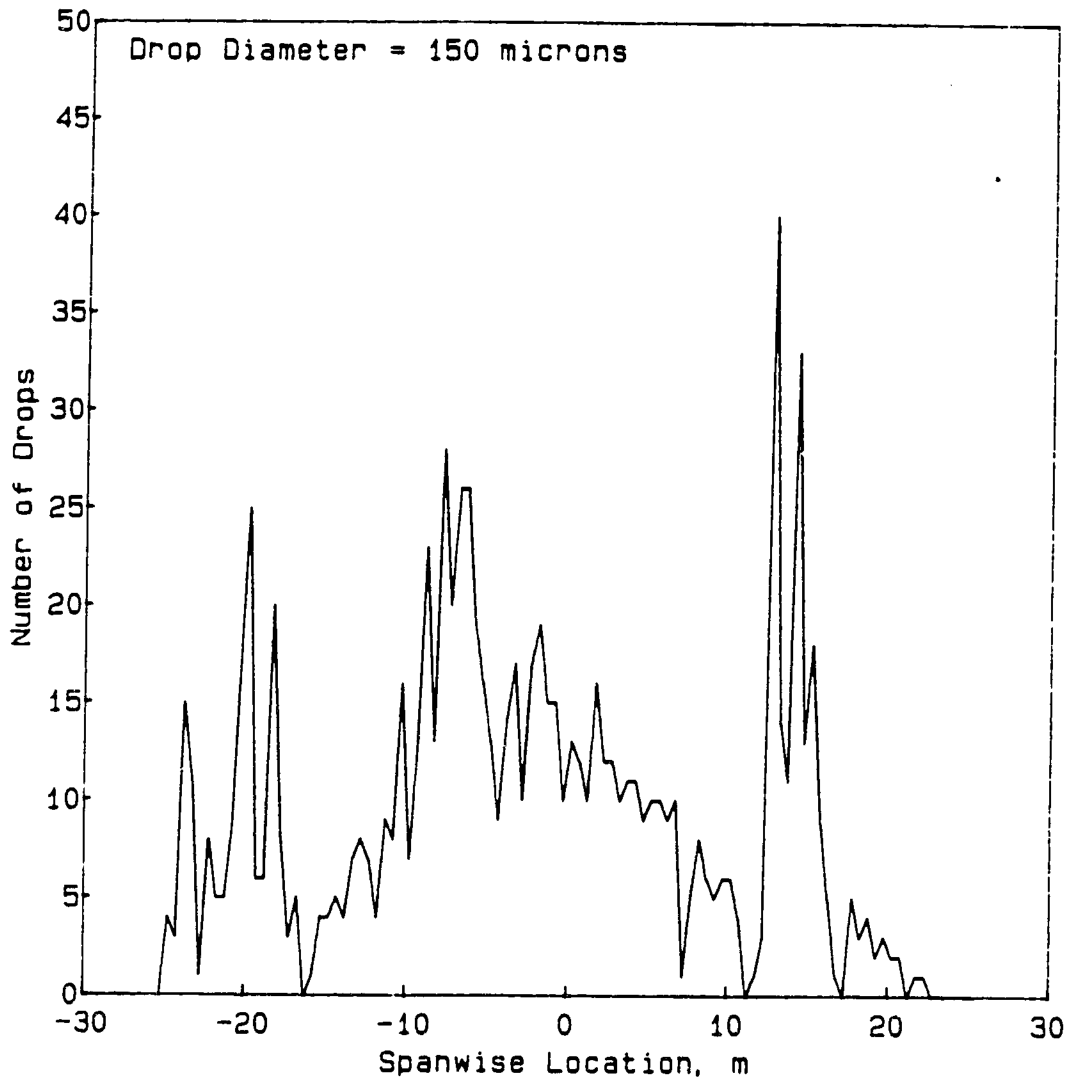


Figure 6.14 Deposition Pattern for Diameter = 150μm

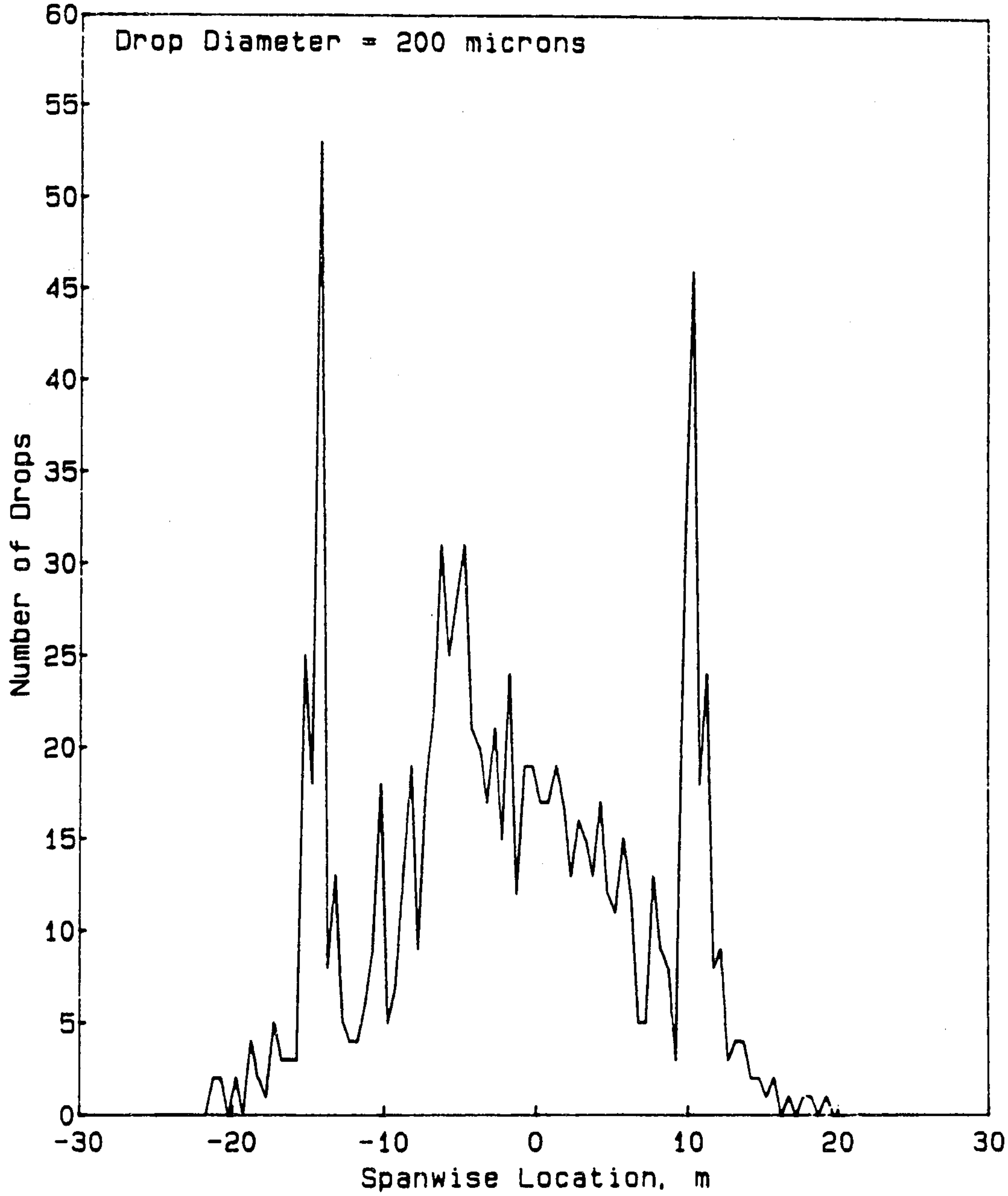


Figure 6.15 Deposition Pattern for Diameter = 200μm

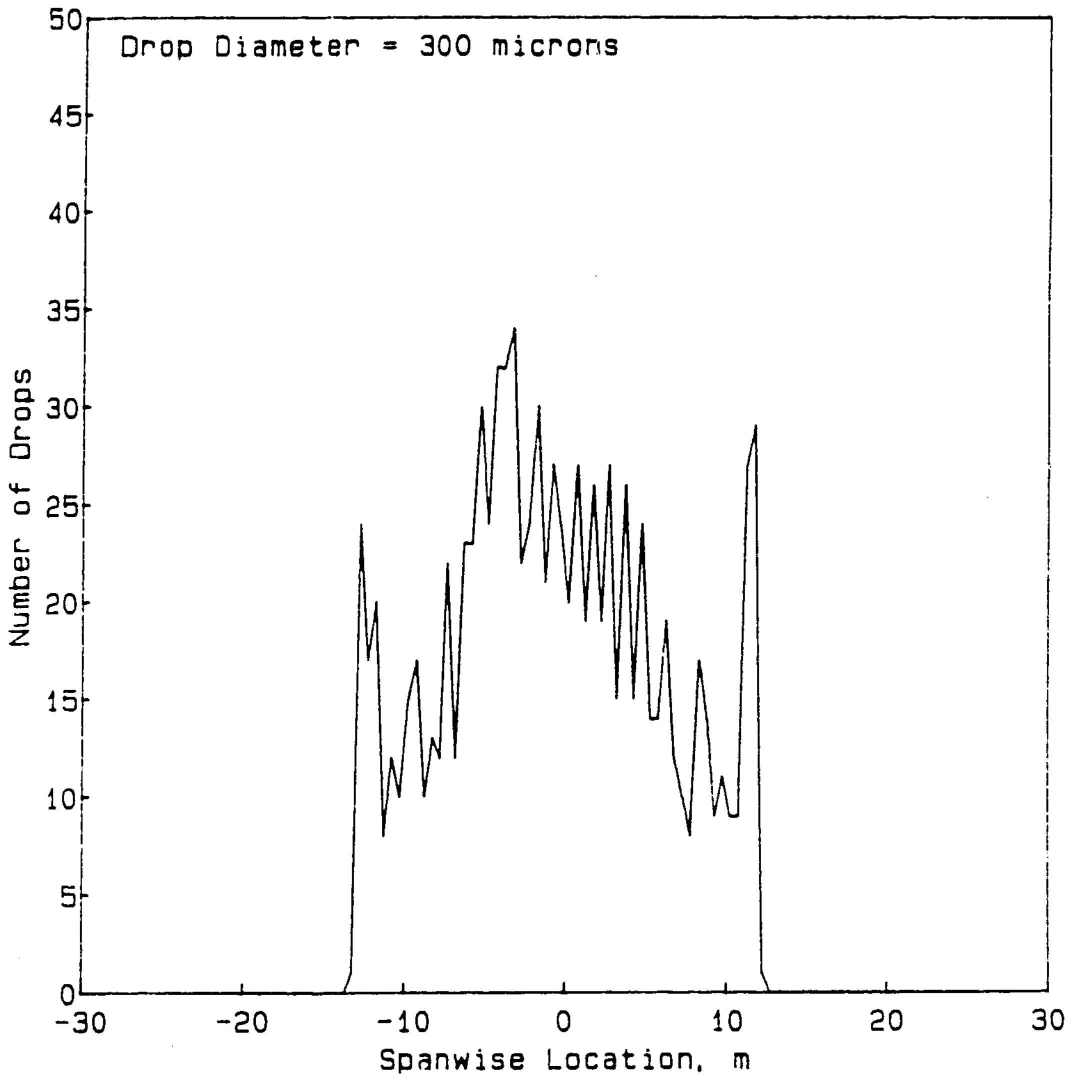


Figure 6.16 Deposition Pattern for Diameter = 300μm

6.6 Airspeed Effects

Figure 6.17 shows the effect of airspeed on the coefficient of variation. Lower airspeeds cause a proportionate increase in the value of the total circulation for the aircraft and, thus, greater induced velocities due to the trailing vortices. Figures 6.18 to 6.20 show that the width of the deposition pattern is greater for decreased airspeeds, as would be expected, and leads to improved uniformity over greater swath widths. Apparently, an airspeed of 45 m/s produces the best uniformity over the whole range of swath widths of concern. A likely contributing factor would be the disappearance of the usual spikes of deposition at the edges of the pattern as can be seen by examining Figures 6.18 to 6.20. The reason is not clear, but it is apparent that for any particular configuration there will likely be an optimum airspeed for uniformity considerations.

It should be noted that the airspeed may not be as flexible as this test appears to indicate. The safe operation of the spray aircraft would be the most important consideration and would restrict the normal operating airspeed.

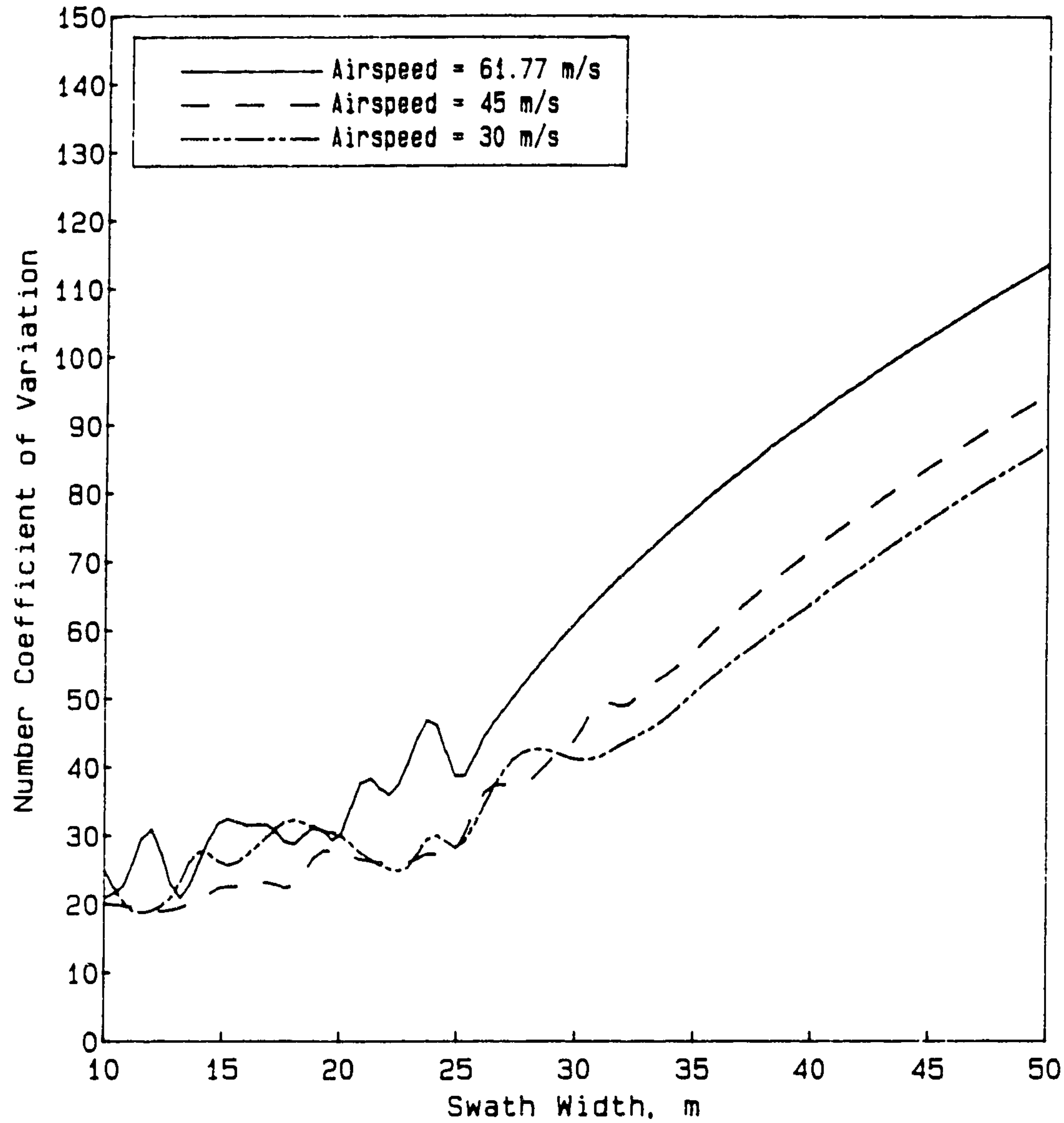


Figure 6.17 Effect of Airspeed on the Coefficient of Variation

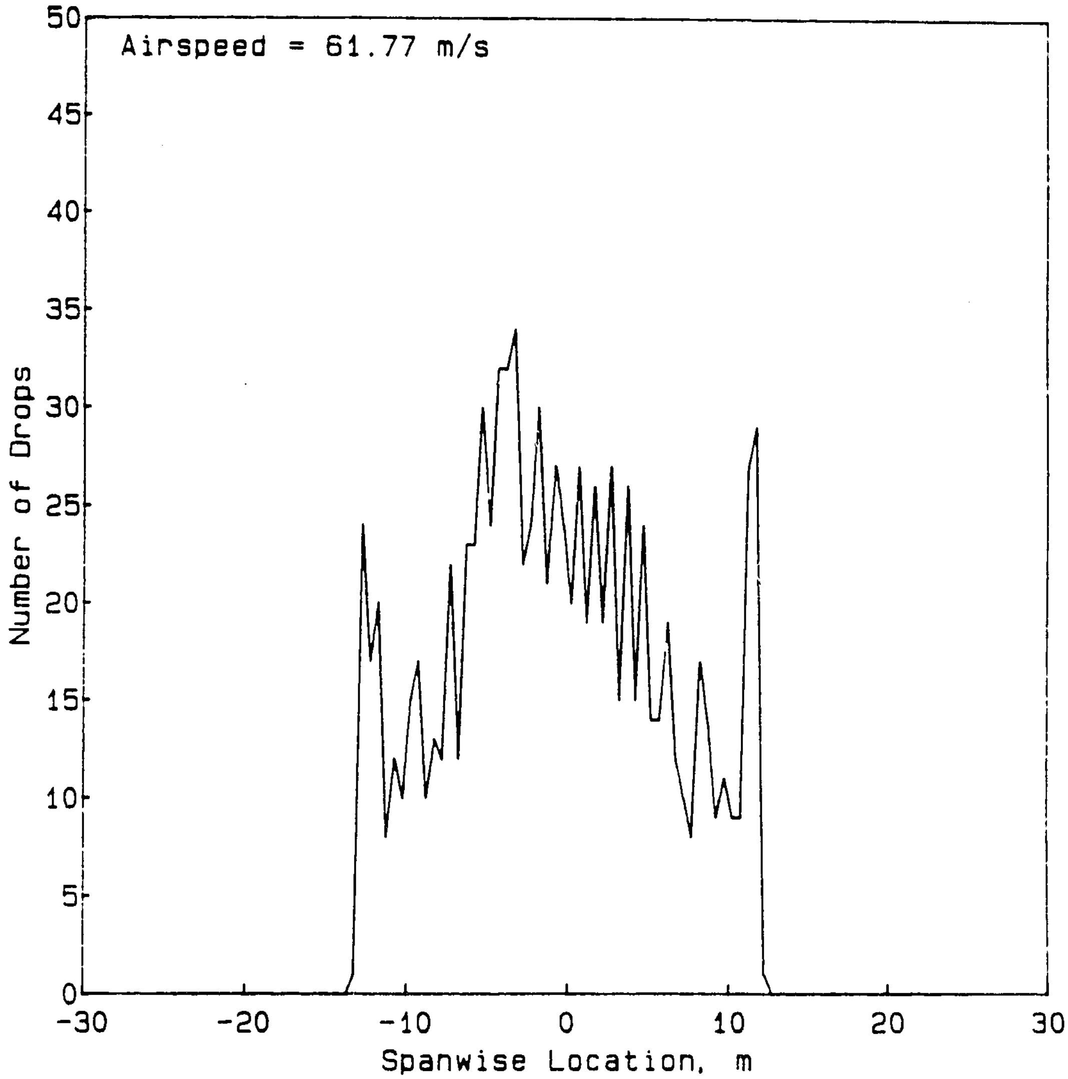


Figure 6.18 Deposition Pattern for Airspeed = 61.77 m/s

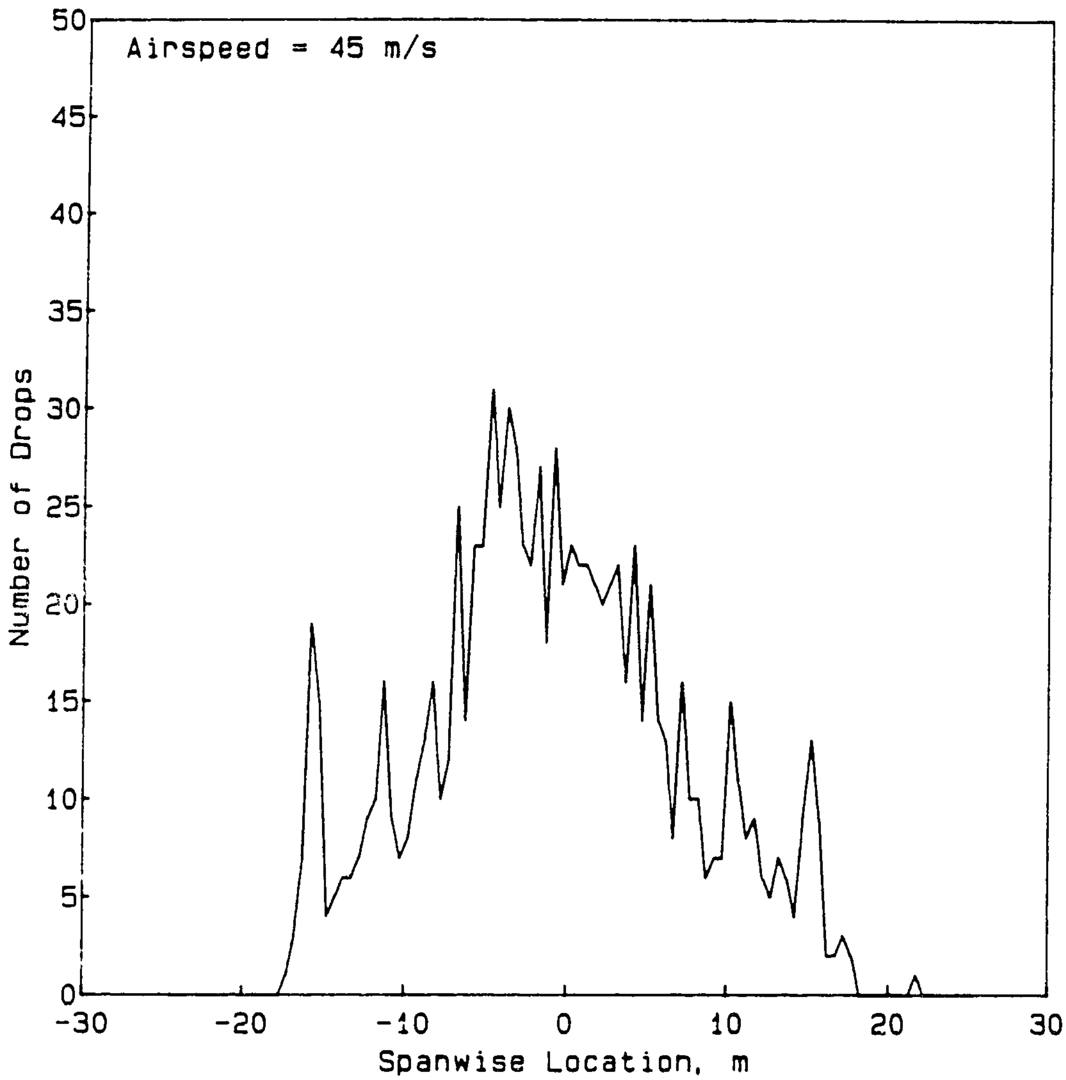


Figure 6.19 Deposition Pattern for Airspeed = 45 m/s

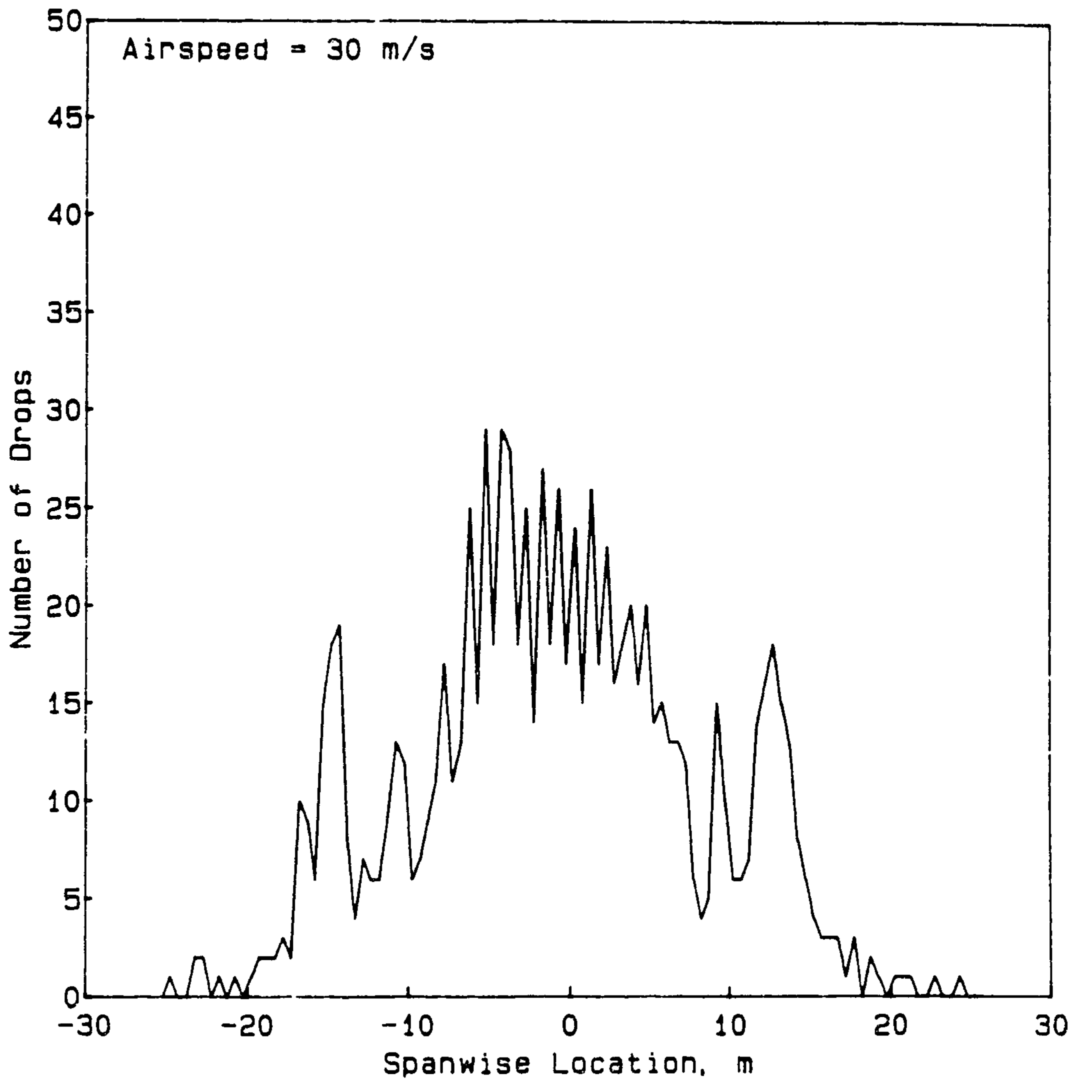


Figure 6.20 Deposition Pattern for Airspeed = 30 m/s

6.7 Gross Aircraft Weight Effects

The gross weight of the aircraft was varied from the initial weight of 26689N to 13344N and 6677N. These weights correspond to the gross weight, one half of the gross weight and one quarter of the gross weight for a typical spray operation, for the aircraft used for model testing.

Figure 6.21 shows the effect of gross weight of the aircraft on the coefficient of variation. As can be seen, the lowest weight produces the lowest coefficient of variation, but at a swath width of only 17m. The weight of 13344N shows the best uniformity at greater swath widths, and gives a usable swath width up to about 21m without significant change in the coefficient of variation. The gross weight of 26689N produces slightly poorer uniformity for narrow swath widths than the lower weights, but is better at widths greater than 22m. The fact that increasing weight produces a wider usable swath, follows from the direct relation between the circulation of a wing and the lift produced by it. Figures 6.22 to 6.24 show the deposition patterns for the three gross weights, and the increasing swath width for the higher gross weight. The coefficient of variation for a weight of 26689N is higher than that for the weight of 13344N at a width of 21m, and the coefficient of variation for a weight of 13344N is higher than that for a weight of 6677N at a width of 17M.

Thus, increasing gross weight increases the usable swath width, but the uniformity of deposition is sacrificed.

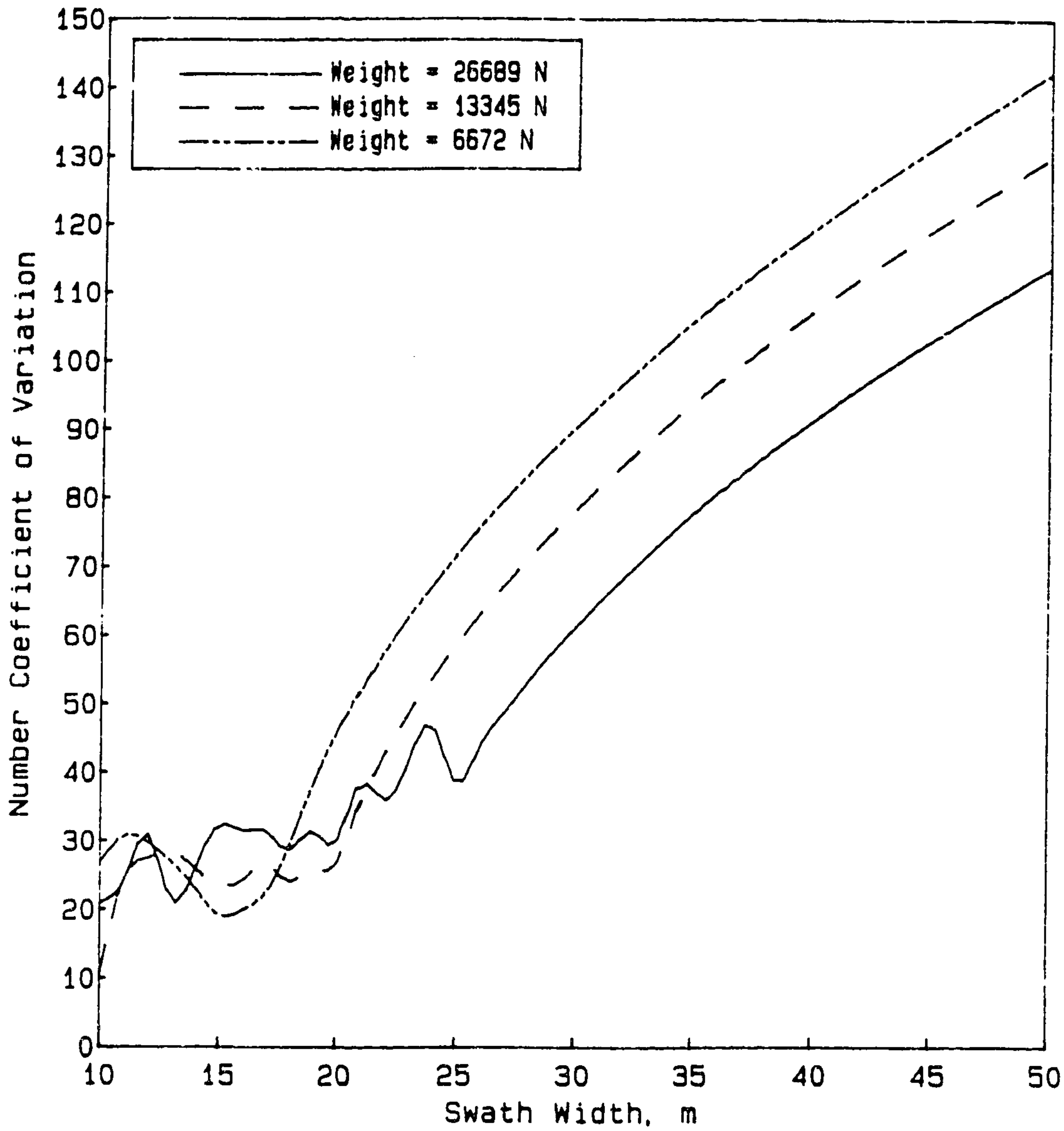


Figure 6.21 Effect of Aircraft Gross Weight on the Coefficient of Variation

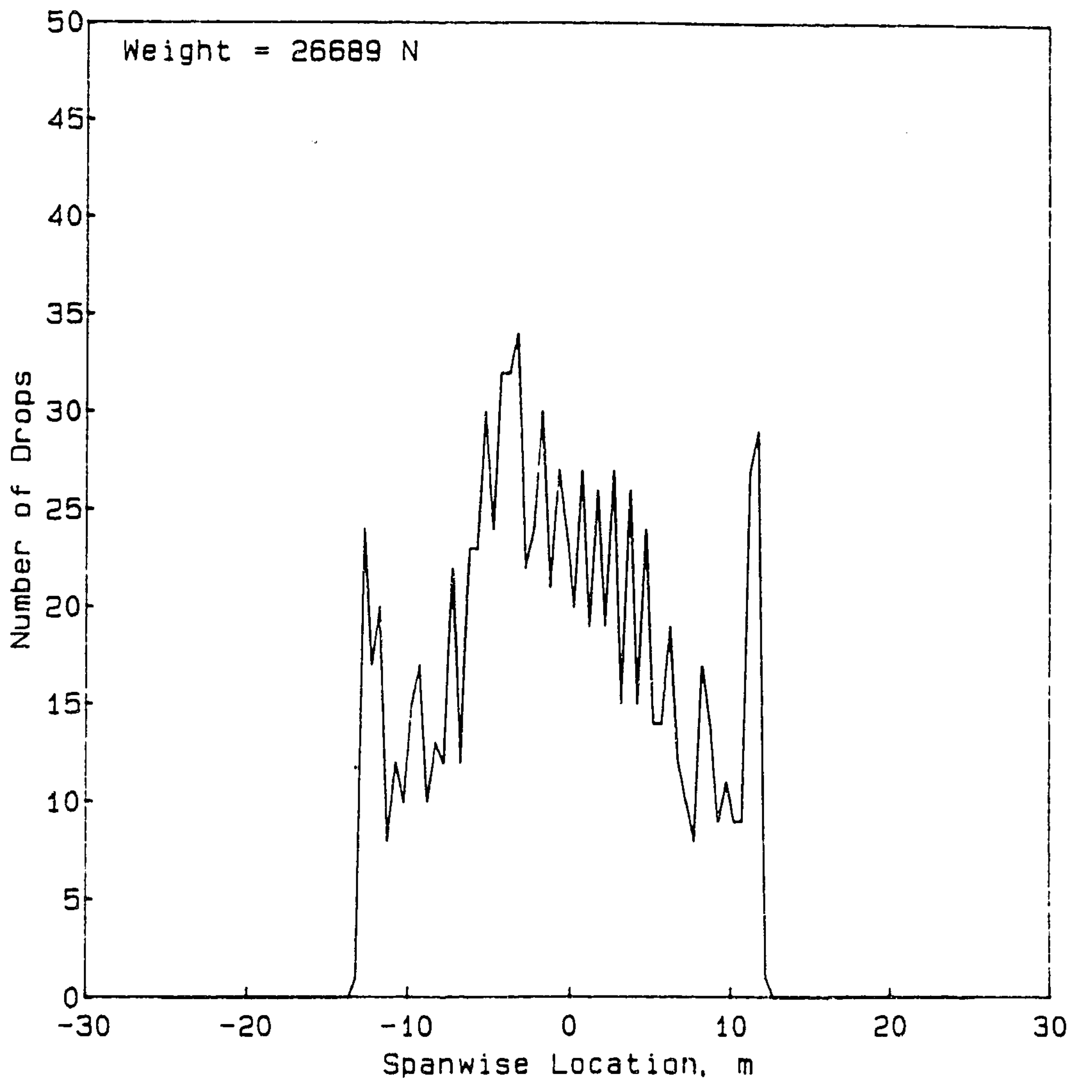


Figure 6.22 Deposition Pattern for Weight = 26689N

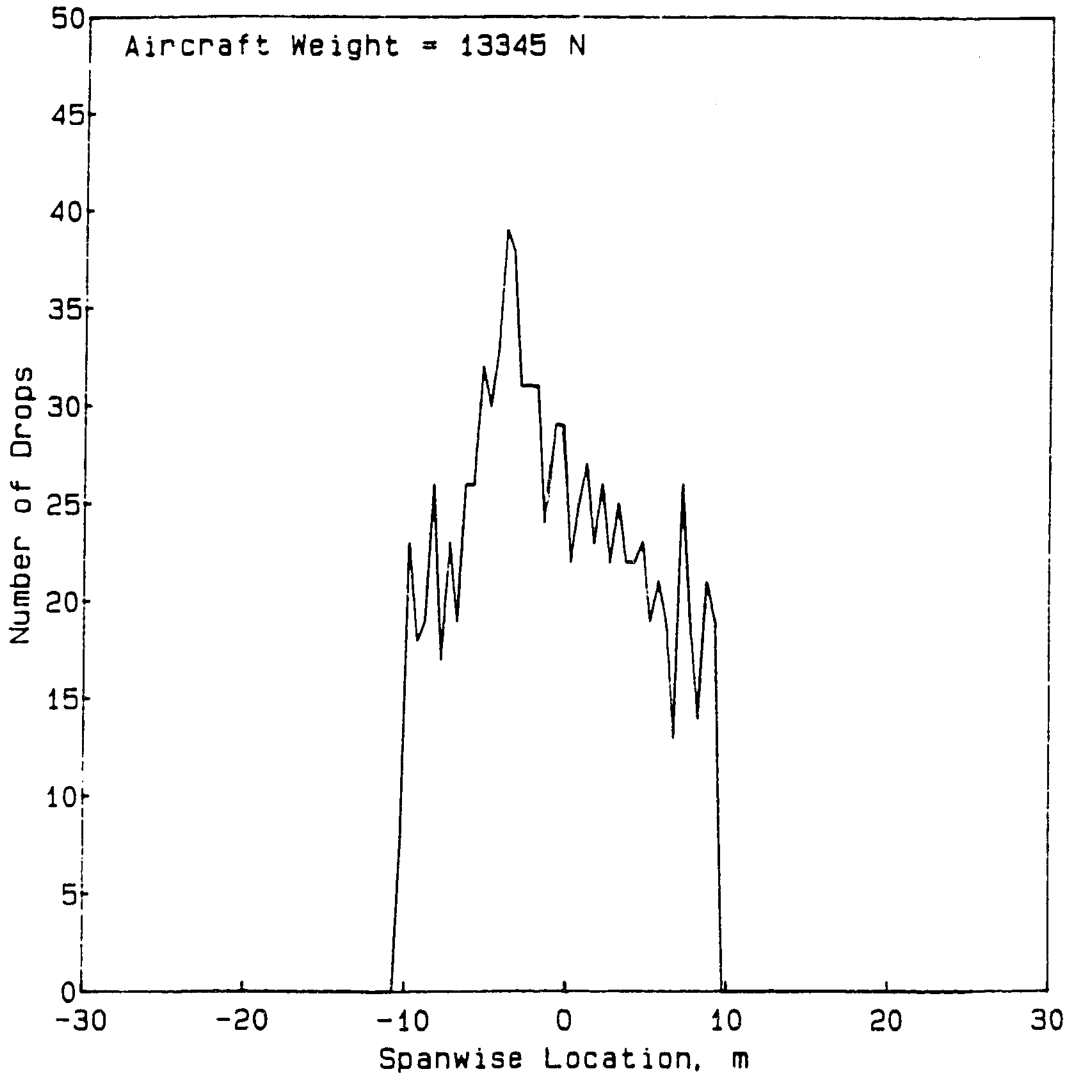


Figure 6.23 Deposition Pattern for Weight = 13344N

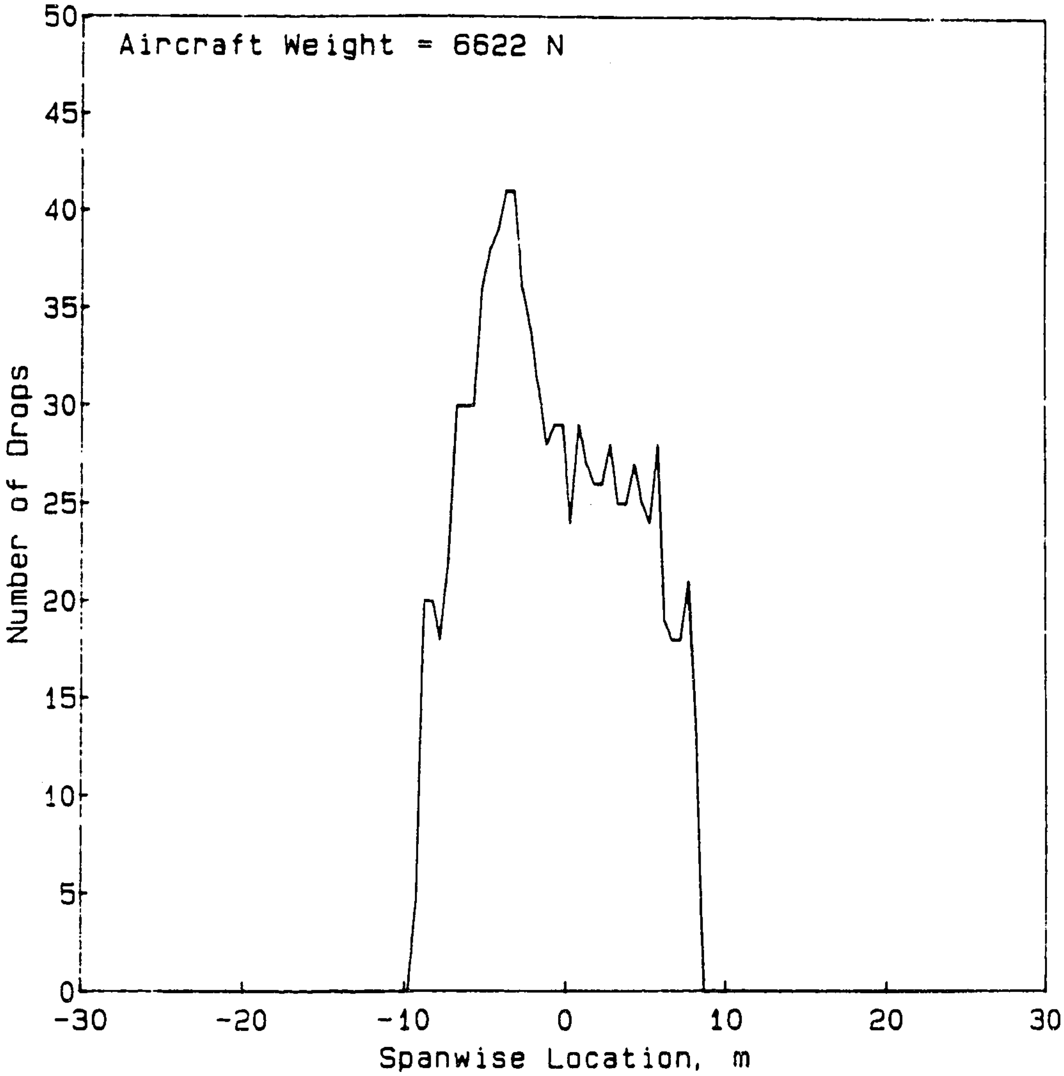


Figure 6.24 Deposition Pattern for Weight = 6677N

6.8 Aircraft Height Effects

The height of a spray aircraft during spray application is a variable which is readily controllable by the pilot. However, the close proximity of the ground definitely increases the possibility of aircraft crashes during spray applications, hence, the higher the aircraft is operated during spraying, the greater will be the safety margin for maneuvering.

The effect of three different heights on the spray uniformity is shown in Figure 6.25. The spray nozzle configuration is the standard 12 nozzles (aiming straight down) per semispan evenly spaced from 5 to 100% span. The height of 2.0m gave the best uniformity for narrow swath widths, up to 19m, as would be expected with this nozzle configuration. The best uniformity was obtained by the lowest height for narrow swath widths, but for increased swath width the best uniformity was produced by the greater height.

Figures 6.26 to 6.28 show the deposition patterns for the three heights. The spikes of spray deposit near the edge of the pattern (due to using the 100% span boom) are seen to be most pronounced for the 5.0m height and least for the 2.0m height. Undoubtedly, this contributes to the decreased uniformity with increased height. The plots, also, show that the pattern is more uniform for a single pass for the 5.0m height, than the other heights. The coefficient of variation is based on the combination of overlapping swaths, hence, the uniformity is lowest for the 5.0m height for usable swath widths.

Again, as for some other parameters, there appears to be a particular value for height which will give the optimum uniformity for each particular configuration. The final decision on the height at which to spray would have to be one which combines the considerations regarding spray distribution uniformity and the safety of the spray operation.

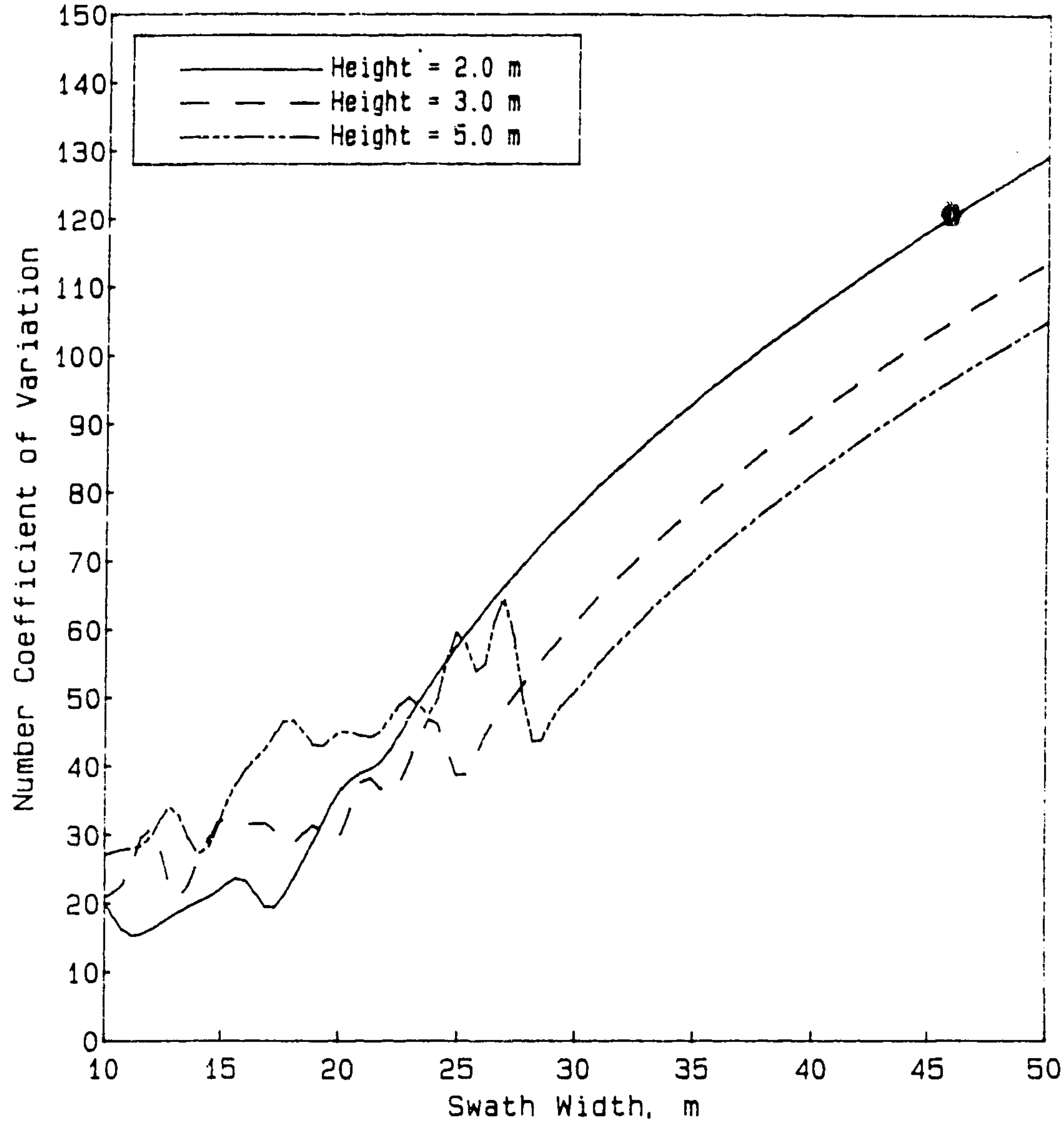


Figure 6.25 Effect of Aircraft Height on the Coefficient of Variation

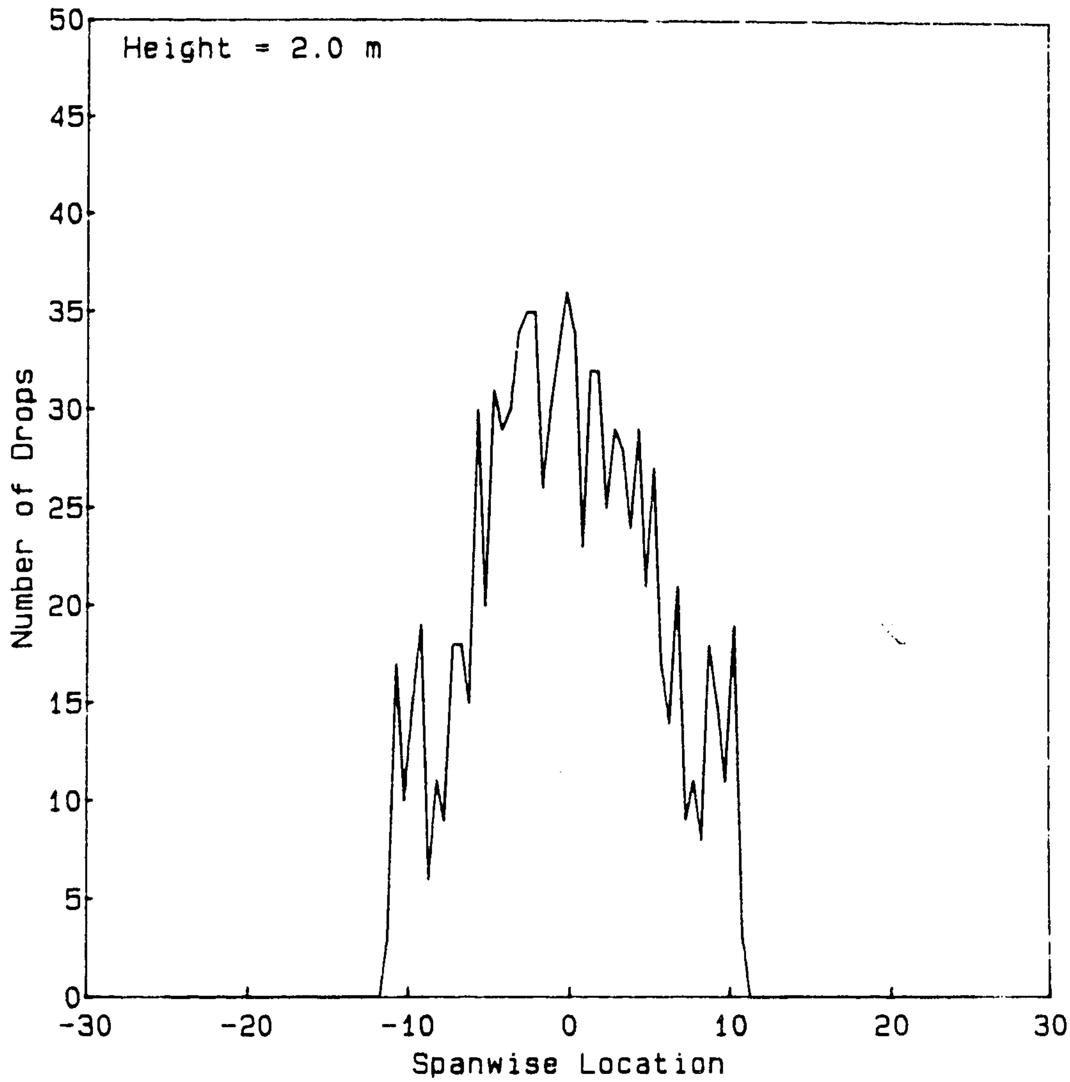


Figure 6.26 Deposition Pattern for Height = 2.0m

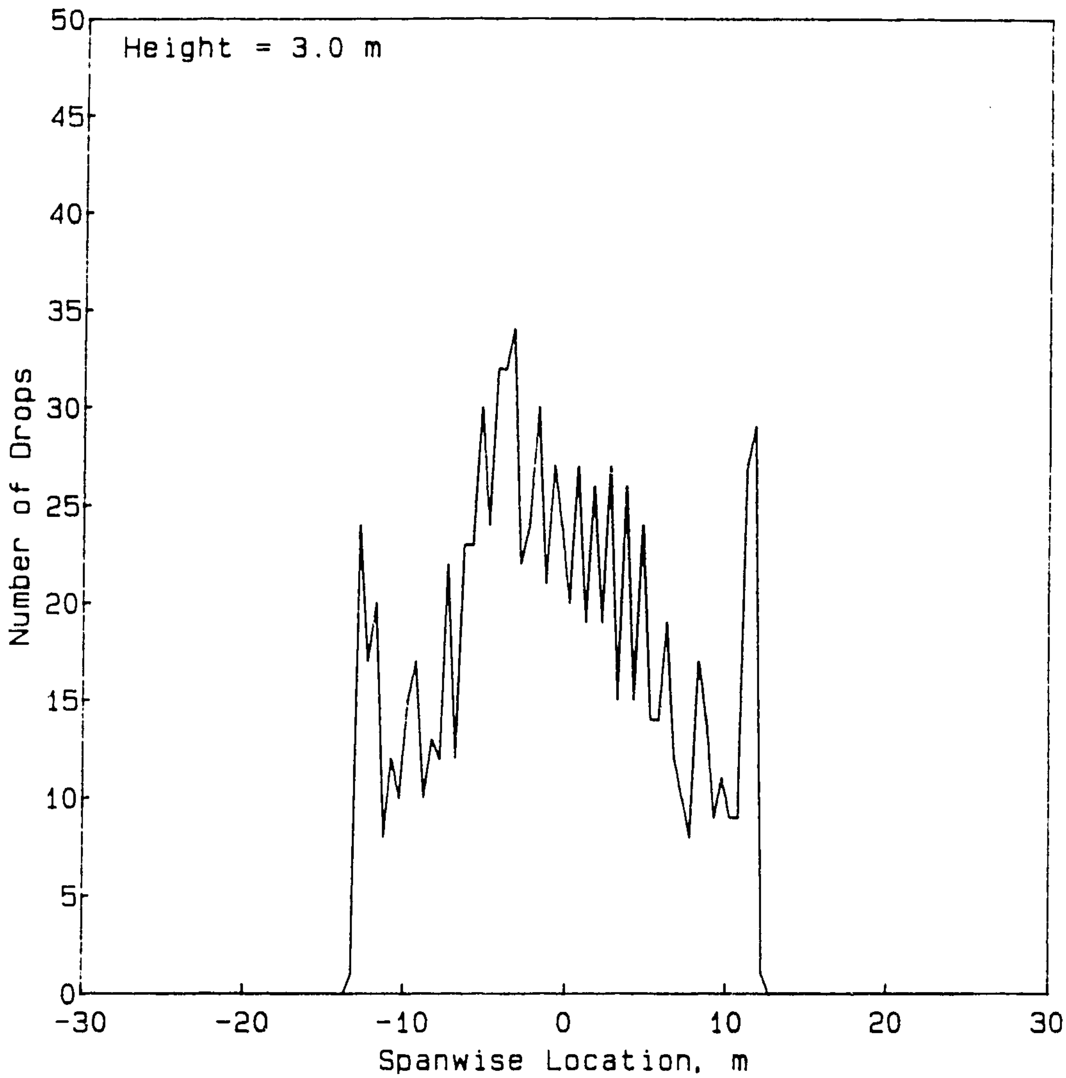


Figure 6.27 Deposition Pattern for Height = 3.0m

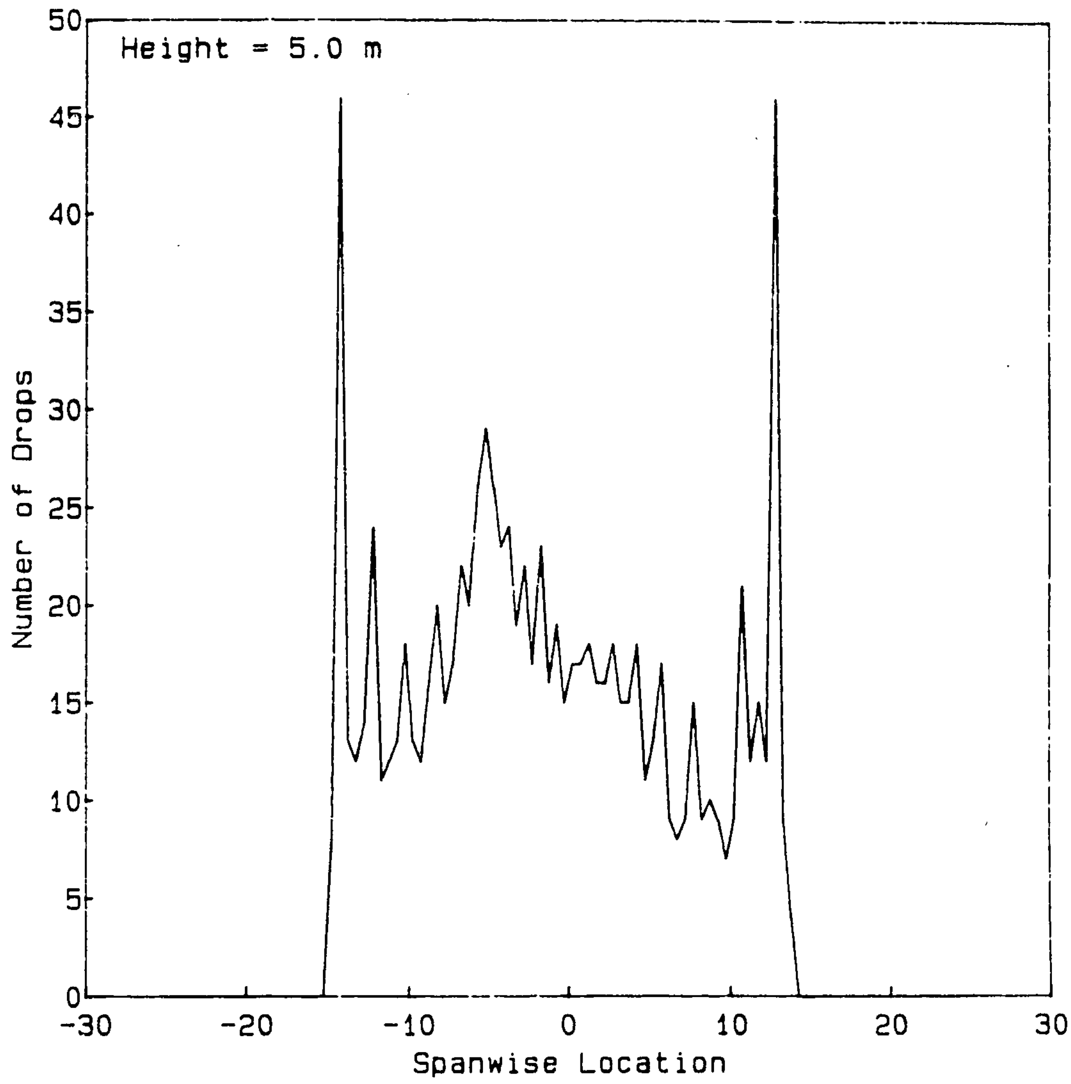


Figure 6.28 Deposition Pattern for Height = 5.0m

6.9 Effects of Nozzle Horizontal Angle

The effect of the horizontal angle of the spray nozzle was demonstrated using the three settings, 0°, 45°, and 90°. The nozzle aims straight back for a setting of 0° and straight down for a setting of 90°.

Figure 6.29 shows that the horizontal angle does not have a very significant effect on the coefficient of variation. The 0° setting is slightly worse than the other two over the entire range of swath widths considered, but all three settings result in fairly similar coefficients of variation.

Comparing the actual deposition, as seen in Figures 6.31 and 6.32, shows the pattern for the 45° and 90° nozzle angles to be very similar. Figure 6.30 shows the 0° setting to have the deposition spikes at the edge of the pattern (due to the 100% span boom) moved somewhat inwards and a slightly greater lateral movement of the spray.

Over all, the horizontal angle of the spray nozzle does not significantly affect the spray distribution for a constant drop size. However, this would not be the case in practice. The reason being that in reality the size of the drop emitted from a nozzle is highly dependent on the nozzle horizontal angle, all other conditions being held constant.

Angles such that the drops are emitted more in line with the airflow result in larger drops being formed, and angles which are more inclined to the local airflow cause the spray to break up into smaller drops (Yates *et al.*, 1985).

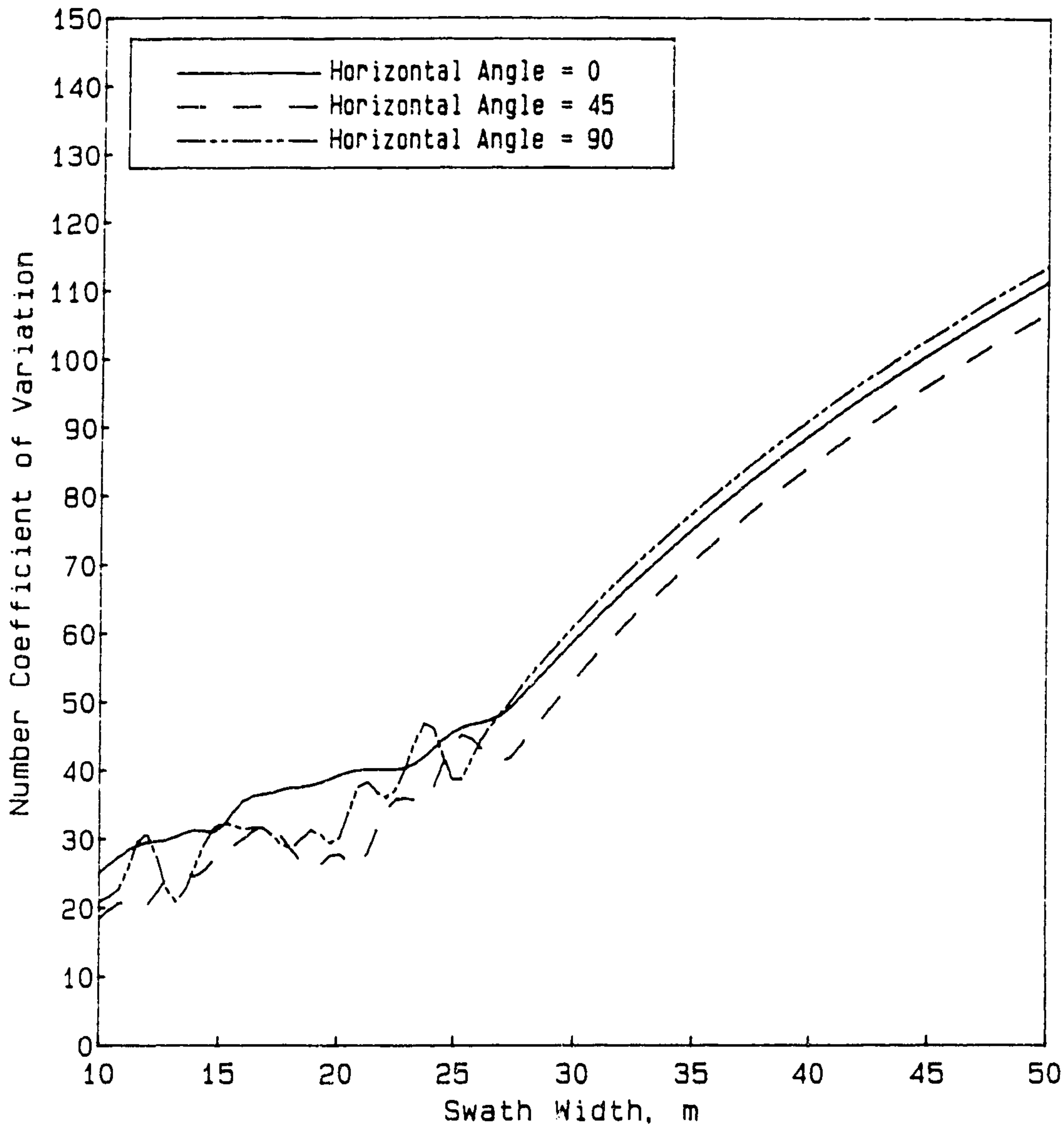


Figure 6.29 Effect of Nozzle Horizontal Angle on the Coefficient of Variation

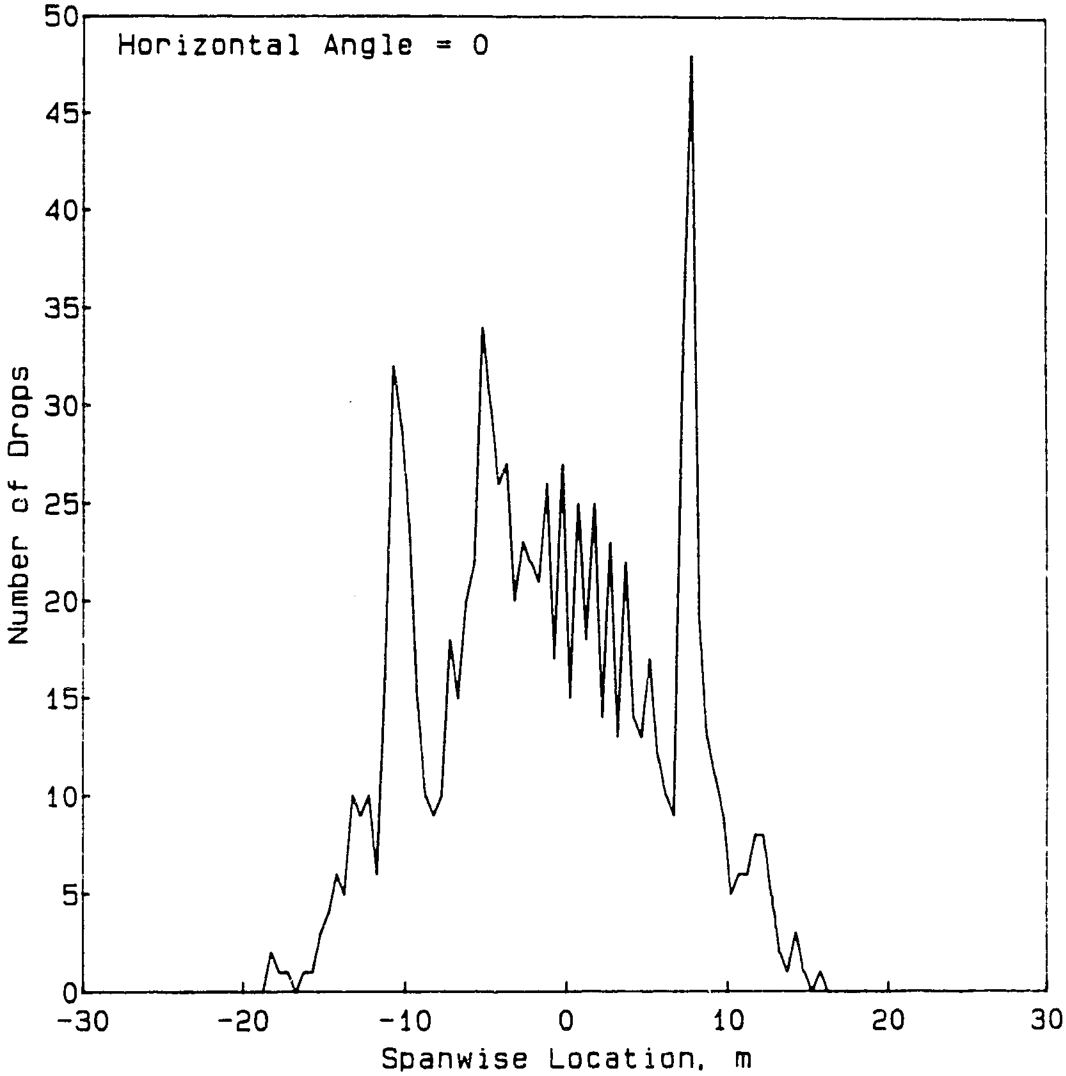


Figure 6.30 Deposition Pattern for Horizontal Angle = 0°

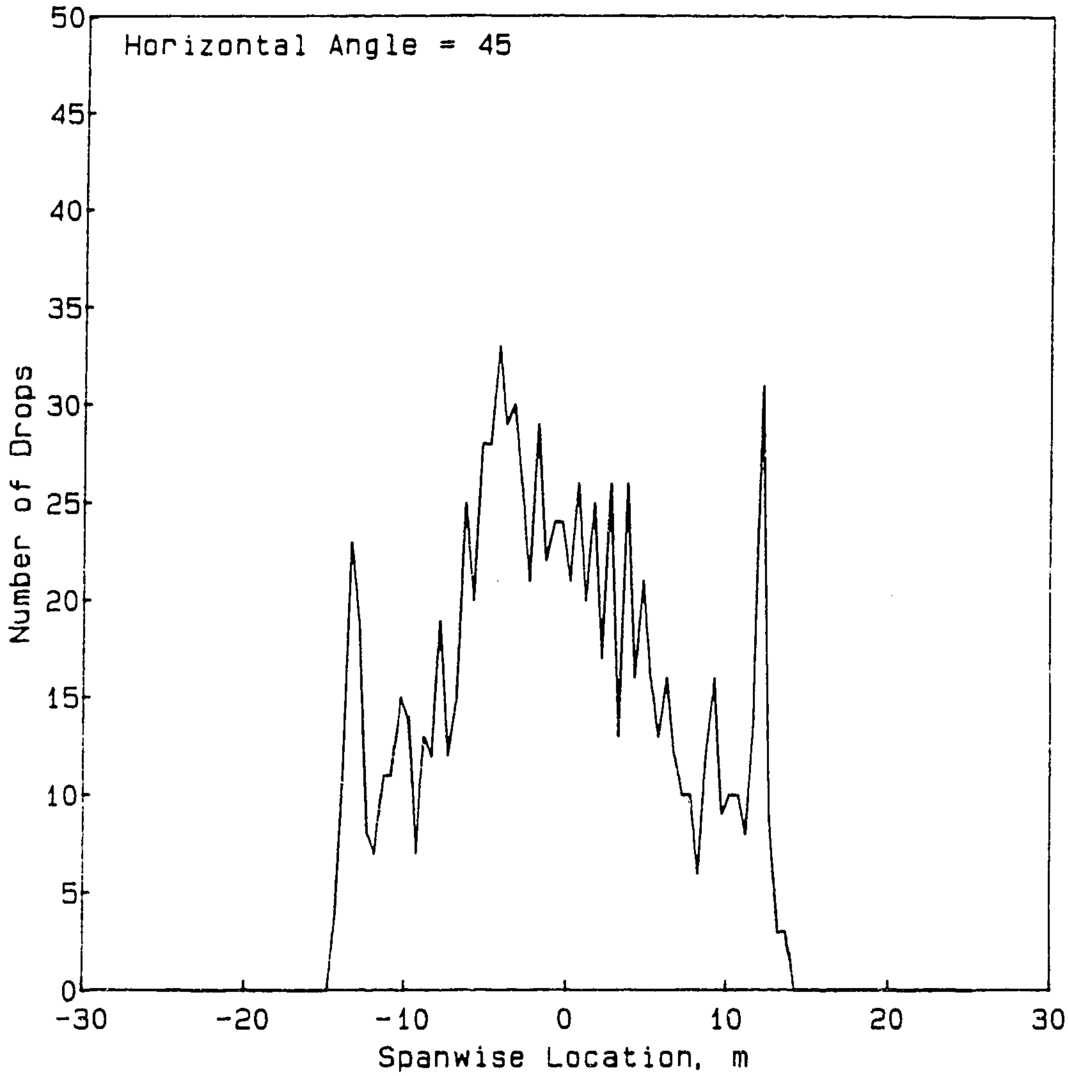


Figure 6.31 Deposition Pattern for Horizontal Angle = 45°

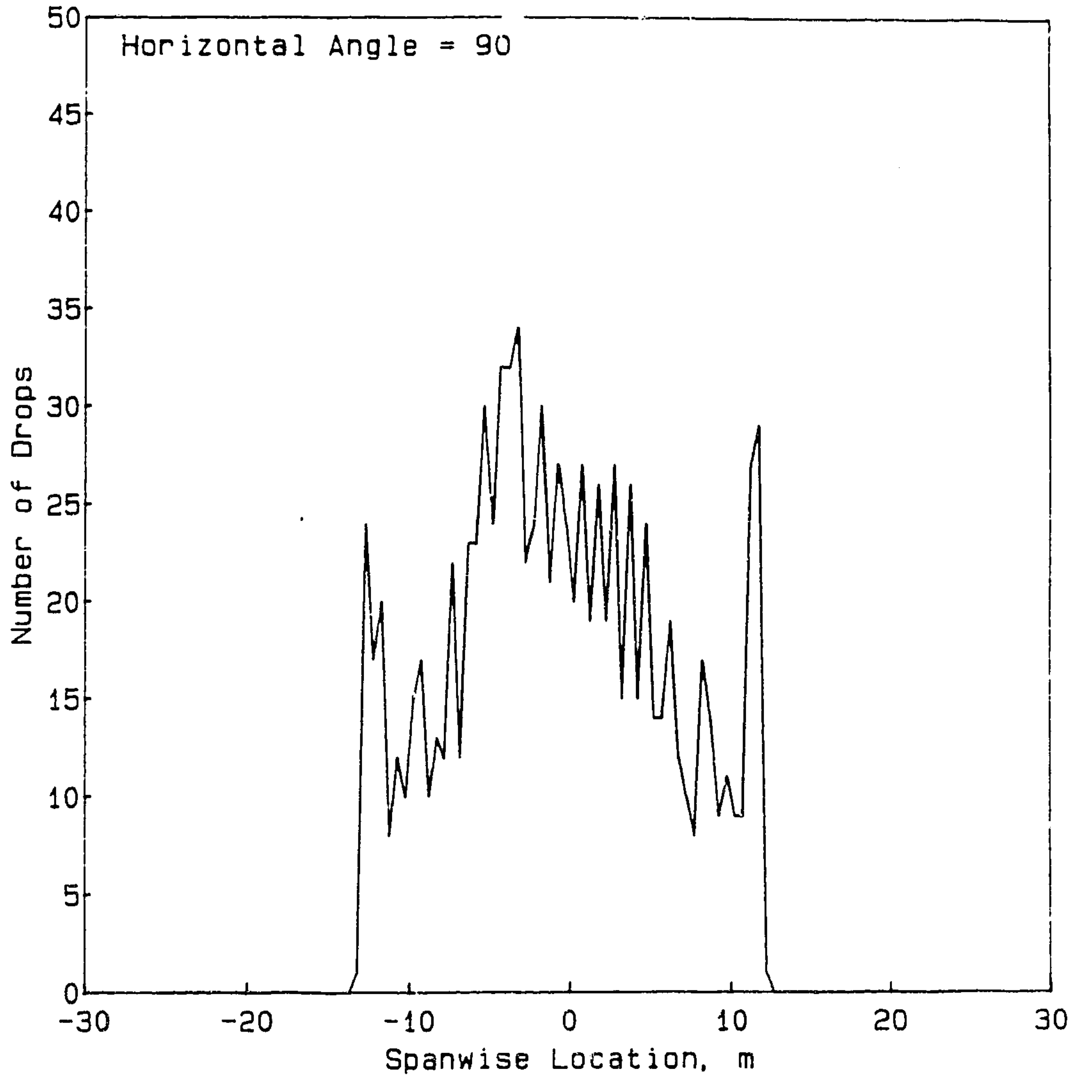


Figure 6.32 Deposition Pattern for Horizontal Angle = 90°

6.10 Effects of Crosswind Strength

The nature of the effect of crosswinds on spray movement was examined using three different crosswind component values, 5 m/s, 0 m/s, and -5 m/s. Positive crosswinds are from the left to the right, with respect to the pilot looking forward from the cockpit. As a result, the negative crosswind combined with the induced velocity from the propeller to increase the air velocity to the left for positions lower than the centerline of the propeller axis (since the propeller rotation is clockwise). These cases were run with the standard configuration and using a 5 to 90% span spray boom with uniform nozzle spacing, symmetric about the aircraft centerline.

Examination of Figure 6.33 shows significant reduction of uniformity for either nonzero crosswind conditions. For swath widths less than 15m, the negative crosswind produced a less uniform spray distribution than the positive crosswind. However, at greater swath widths the positive crosswind was slightly better.

Figure 6.36 shows the spray distribution from the -5 m/s crosswind. The figure shows a double spike of deposition, which would be a result of the induced velocity of the propeller combined with the negative crosswind. The spray from the right side is moved leftwards due to the crosswind, and downwards due to the propeller induced air velocity, hence, producing the spike at the 2m position. The spike at -9m would be due to the propeller induced air velocity, upwards and leftwards, such that the drops from inboard positions would remain airborne longer, hence, moving leftwards more. Figure 6.34 shows that the positive crosswind produced a greater deposit just right of the centerline. This would be a result of the spray from the left side moving towards the right, but when near the centerline being forced downwards due to the propeller swirl. Also, comparison of Figure 6.35 with Figures 6.34 and 6.36 shows that a 5 m/s crosswind, which is a fairly light breeze, is seen to skew the deposition downwind. The spray drop used for the test was 300μm, which is a fairly coarse spray for many spray operations, hence, for smaller drop sizes the effect would even be more pronounced. Overall, the crosswind has a major effect on the spray distribution and uniformity, but the direction of the propeller swirl will also be a factor for any particular case.

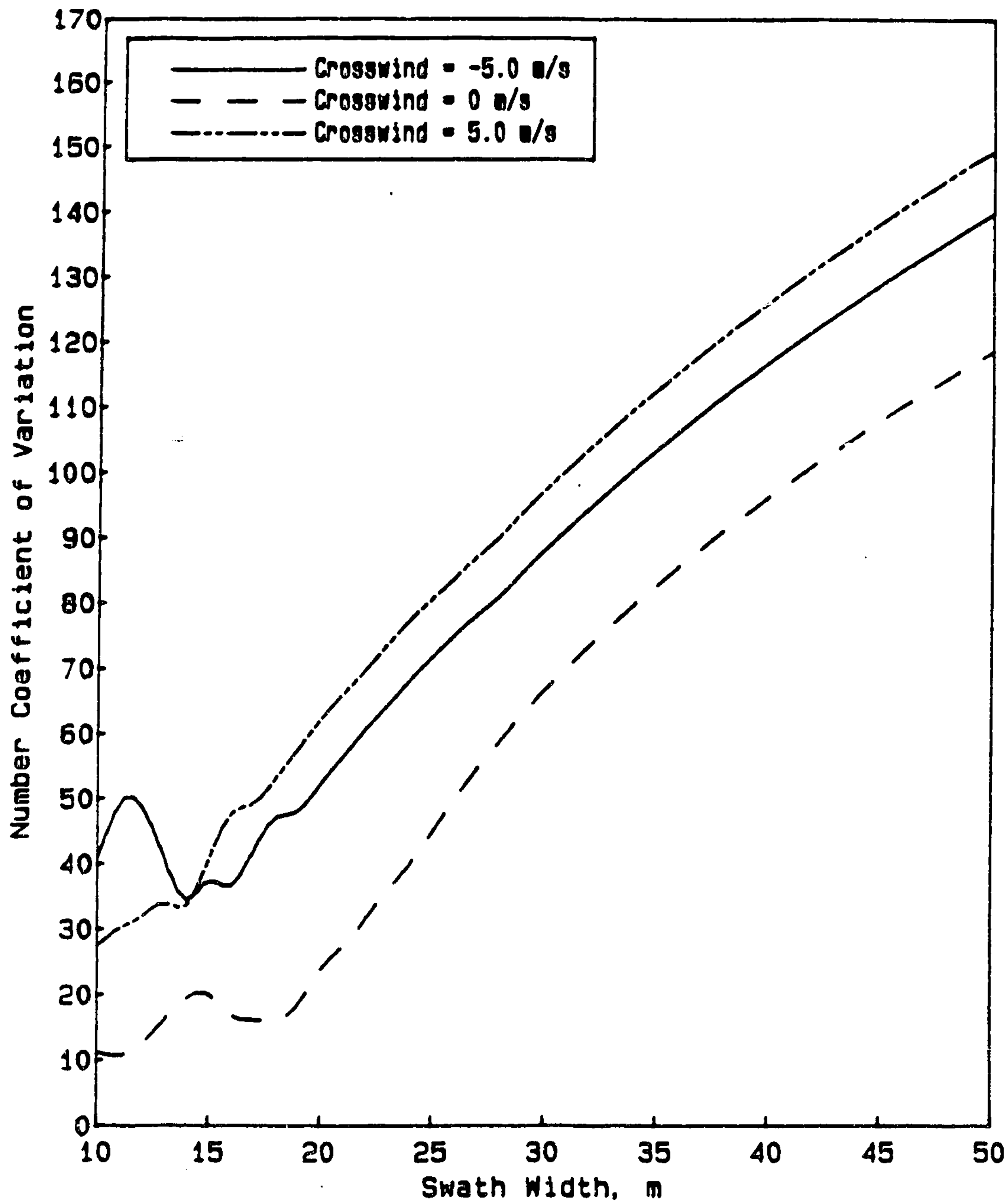


Figure 6.33 Effect of Crosswind Strength on the Coefficient of Variation

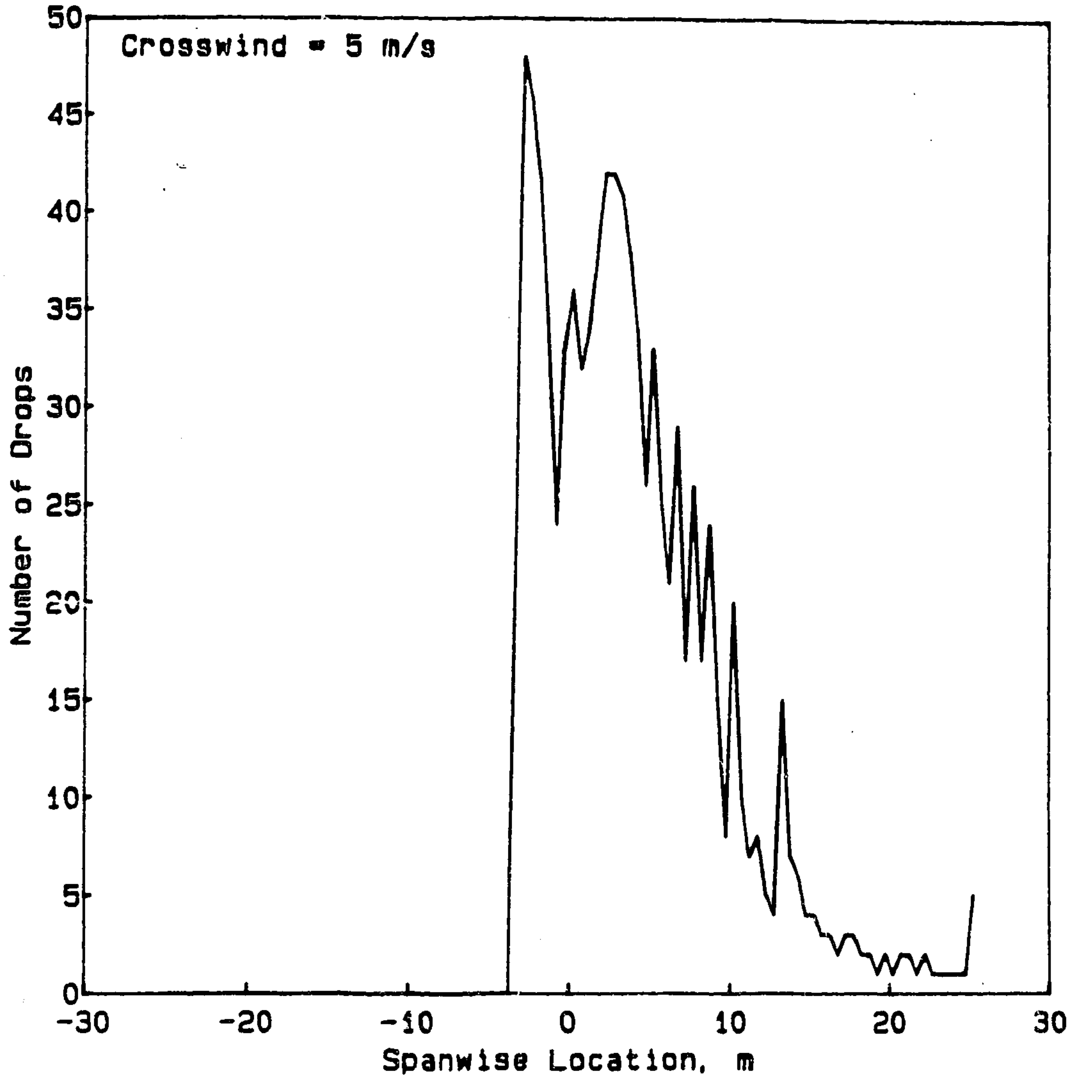


Figure 6.34 Deposition Pattern for Crosswind = 5 m/s

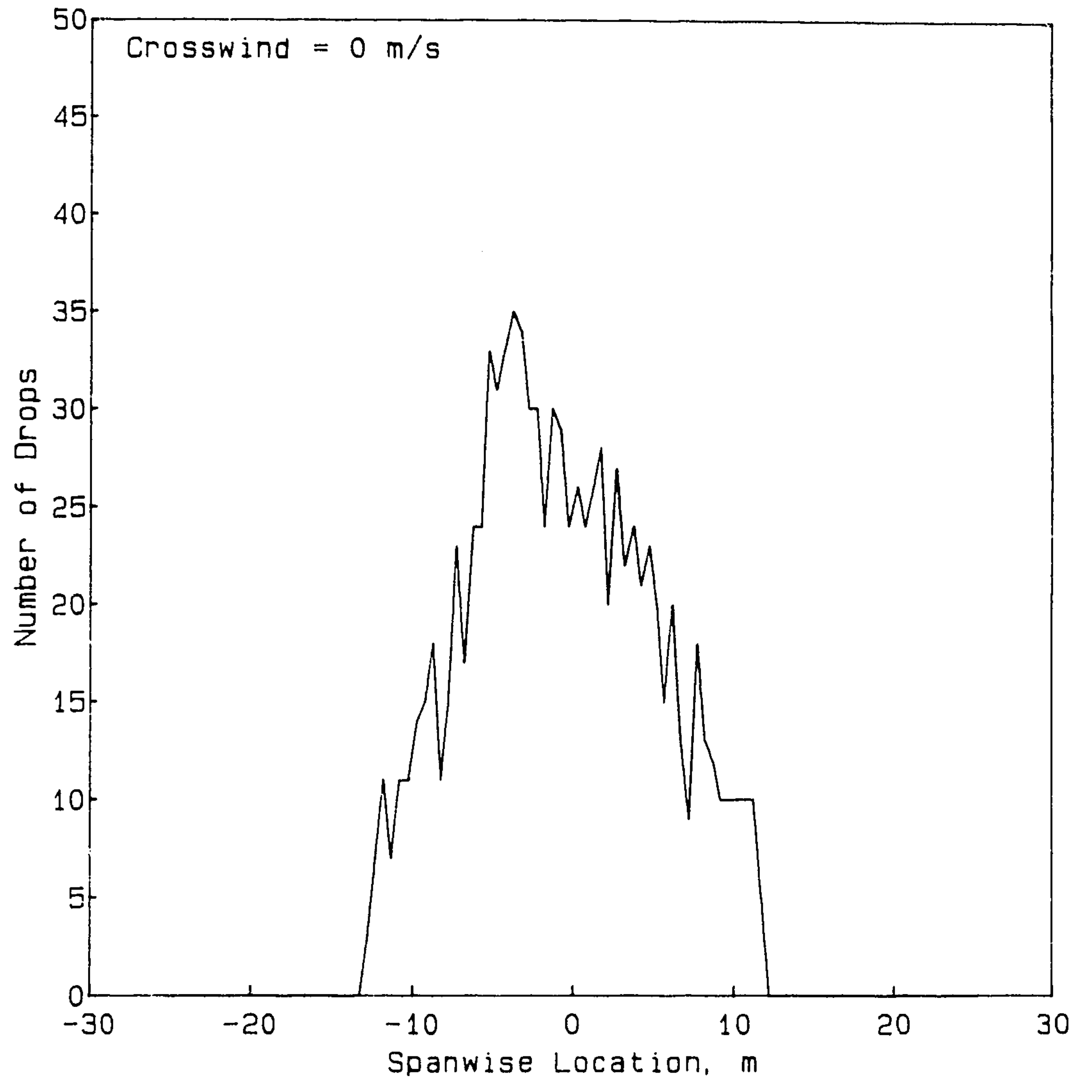


Figure 6.35 Deposition Pattern for Crosswind = 0 m/s

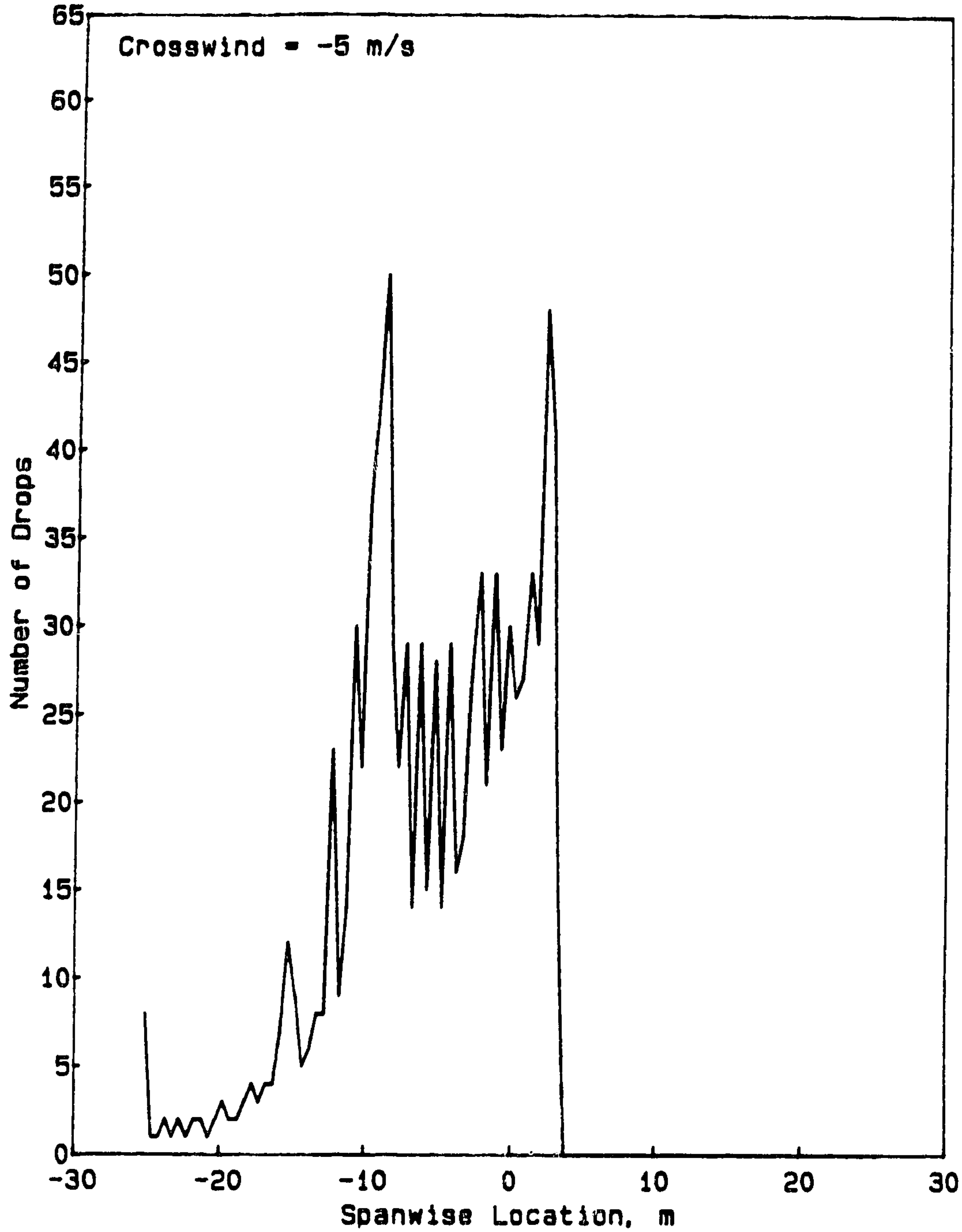


Figure 6.36 Deposition Pattern for Crosswind = -5 m/s

7. CONCLUSION

The major emphasis of this work was the development of a computer model to simulate spray movement in the wake of an aircraft. The model was then tested using some flight test data acquired from D.J. Morris from NASA (Morris *et al.*, 1984). Then the final portion dealt with the application of the computer model in order to examine some significant parameters which affect the depostion of spray

The model development was a major part of the work, the end product being two Fortran 77 computer programs. The first was WAKE77, the program which was specifically configured to be applied to hypothetical aircraft spraying conditions. WAKE77 was the program used to generate the results shown in Chapter 6, and was the desired end result of the work. The second was NASA77, the program which was specifically oriented to handle the predicted movement of spray in the aircraft wake, for actual flight tests. NASA77 accepted flight test conditions from data files and predicted the final drop locations for each of the flight tests. Both NASA77 and WAKE77 were based on the same subroutines for calculating the spray movement, but the input data and its manipulation differed vastly between the two programs.

Chapter 5 dealt with the computer model verification and evaluation of the magnitude of several parameters. These included the value of the total circulation, the swirl coefficient, the vortex core coefficient, and the initial spanwise separation of the trailing vortices. A value for the swirl coefficient which was lower than predicted by propeller theory for an unconfined propeller, was found to give good agreement with flight test data. The most suitable value was near 0.0035 to 0.0040. This was reasonable due to the straightening influence of the aircraft fuselage and tail empennage. The vortex core coefficient which determined the radial extent of the viscous core of the vortex was shown to fall near the theoretical value as discussed in Chapter 3. The value of the vortex core coefficient was found to be less critical to the results than initially suspected. Values of 0.036 and 0.018 gave identical results under the same flight test conditions, and these differed only slightly from the results using a core coefficient of 0.0775 (the value suggested by Spreiter and Sacks, 1951). The value of the total circulation was not as easy to determine exactly. There appeared to be some dependency on the initial separation of the trailing vortices. Considering the accuracy of prediction of spray deposits as well as the consistency of that prediction, the value corresponding to the average of the circulation generated by an elliptical and rectangular wing loading appeared to give the most satisfactory results. For that value of the total circulation, the initial vortex separation which provided the best correlation seemed to be near 90% span for an aircraft operating at a height of approximately one half wingspan.

Once the computer model was verified using the program NASA77, then the program WAKE77 could be used with confidence to predict the spray deposition for several hypothetical aircraft spraying configurations. That was the main thrust of Chapter 6. The exact nature of the results was dependent on the exact configuration and operating conditions for each test. For most parameters tested, there appeared to be an optimum value for the given flight test conditions.

Although the exact effect of each parameter is dependent on the conditions for the particular pass of an aircraft, some general remarks may now be made with some certainty. The use of a spray boom extending to approximately 90% span gave the best results for most flight conditions, while a boom extending to 100% span consistently produced a spike of spray deposition near the edge of the pattern. The spray nozzle horizontal angle did not have significant effects on the uniformity, with the rather significant assumption that the drop size was the same for different angles. The effect of a crosswind was examined and found to be a major factor deteriorating the uniformity of deposition by a significant amount.

The decision concerning the distance between adjacent passes of the aircraft is often a compromise between several conflicting goals. Although the desire is to optimize the spray operation efficiency and uniformity of deposition, the need to consider the safe operation of the aircraft should predominate in any decisions concerning the value of parameters such as airspeed and spraying height. For instance, the operation at reduced height may lead to better uniformity, but the distances between passes of the aircraft will have to be reduced and there will be greater risk of an aircraft accident. The airspeed for spray application, also, must be a compromise value. The airspeed may be reduced in order to improve spray distribution uniformity for greater swath widths, but at the same time the decreased speed reduces the margin of safety for maneuvering in flight and compensating for adverse atmospheric phenomenon, such as wind gusts.

Overall, the prediction of spray movement in the wake of an aircraft using a simplified model of trailing vortices has led to satisfactory results. Although the results may not be perfect, they do show that the WAKE77 computer program can be useful for analysing the effects of various parameters on the uniformity of spray deposition in the wake of an aircraft.

Trayford and Welch (1977), Bragg (1977) and others have developed models for prediction of spray movement as well. However, to date it appears that there has not been much use of these models for the improvement of spraying systems which are in commercial usage. This likely is because their programs are not readily available. The proper objective of such a study as this is that the operator of a spray aircraft could apply the results to improve spray deposition. Thus, the computer program which was developed herein will be readily available to interested persons.

8. REFERENCES

Amberg, A.A. and B.J. Butler. 1969. High Speed Photography as a Tool for Spray Droplet Analysis. ASAE Paper No. 69-140. Am. Soc. Ag. Eng., St. Joseph, Michigan 49085.

ASAE Standards. 1985. Calibration and Distribution Pattern Testing of Agricultural Aerial Application Equipment. ASAE S386.1. Am. Soc. Ag. Eng., St. Joseph, Michigan 49085.

Batchelor, G.K. 1964. Axial flow in trailing line vortices. Journal of Fluid Mechanics 20(4):645-658.

Beard, K.V. and H.R. Pruppacher. 1969. The determination of the terminal velocity and drag of small water drops by means of a wind tunnel. Journal of the Atmospheric Sciences 26:1066-1072.

Betz, A. 1933. Behaviour of Vortex Systems. National Advisory Committee for Aeronautics, Technical Memorandum, NACA-TM-713.

Bilanin, A.J. and C.duP. Donaldson. 1975. Estimation of velocities and roll-up in aircraft vortex wakes. Journal of Aircraft 12(7):578-585.

Bilanin, A.J., M.E. Teske, and J.E. Hirsh. 1978. Neutral atmospheric effects on the dissipation of aircraft vortex wakes. AIAA Journal 16(9):956-961.

Bragg, M.B. 1977. The Trajectory of a Liquid Droplet Injected into the Wake of an Aircraft in Ground Effect. M.Sc. Thesis. University of Illinois at Urbana-Champaign.

Brown, C.E. 1973. Aerodynamics of wake vortices. AIAA Journal 11(4):531-536.

Corsiglia, V.R., N.A. Chigier, and R.G. Schwind. 1973. Rapid scanning, three dimensional hot-wire anemometer surveys of wing-tip vortices. Journal of Aircraft 10(12):752-757.

Dee, F.W. and O.P. Nicholas. 1969. Flight Measurements of Wing-Tip Vortex Motion near the Ground. Aeronautical Research Council. Current Papers CP-1065. London, England.

Dodes, I.A. 1978. Numerical Analysis for Computer Science. North-Holland, New York. pp.552-563.

Donaldson, C. duP., R.S. Snedeker, and R.D. Sullivan. 1974.

Calculation of aircraft wake velocity profiles and comparison with experimental measurements. Journal of Aircraft 11(9):547-555.

Eisner, von F. 1930. Das Widerstands Problem. Proceedings of the 3rd International Congress for Applied Mechanics 1:23-42. Stockholm, Sweden.

George, M.F. 1985. Tip Vortex Cavitation Research Literature Review. Department of National Defence Canada. Defence Research Establishment Atlantic CR-85-409.

Goering, C.E., L.E. Bode, and M.R. Gebhardt. 1972. Mathematial modeling of spray droplet deceleration and evaporation. Trans ASAE 15(2):220-225.

Harvey, J.K. and F.J. Perry. 1971. Flowfield produced by trailing vorticies in the vicinity of the ground. AIAA Journal 9(8):1659-1660.

Houghton, E.L. and N.B. Carruthers. 1982. Aerodynamics for Engineering Students, 3rd Edition. p256. Edward Arnold Pub. Ltd. London.

Hughes, R.R. and E.R. Gilliland. 1952. The mechanics of drops. Chemical Engineering Progress 48(10):497-504.

Lamb,H. 1932. Hydrodynamics, 6th Ed. pp220-224. Dover Publications, New York.

Langmuir, I. and K. Blodgett. 1946. A Mathematical Investigation of Water Droplet Trajectories. AAF TR No. 5418 (Contract No. W-33-038-ac-9151). Air Technical Service Command, U.S. Army Air Forces.

Lapple, C.E. and C.B. Shepherd. 1940. Calculation of particle trajectories. Industrial and Engineering Chemistry 32(5):606-617.

Manning, W.P. and W.H. Gauvin. 1960. Heat and mass transfer to decelerating finely atomized sprays. A.I.Ch.E. Journal 6(2):184-190.

Marsden, D.J. 1988. Personal Communication. Department of Mechanical Engineering, University of Alberta.

Marshall, W.R. 1954. Atomization and Spray Drying. Chemical Engineering Progress Monograph Series 50(2).

Mason, B.J. 1971. The Physics of Clouds, 2nd Edition. Claredon Press, Oxford.

McCormick, B.W. 1954. A Study of the Minimum Pressure in a Trailing Vortex System. Ph.D. Thesis. Dept. of

Aeronautical Engineering, Pennsylvania State University, Pennsylvania.

McCormick, B.W., J.L. Tangler, and H.E. Sherrieb. 1968. Structure of trailing vortices. Journal of Aircraft 5(3):260-267.

Morris, D.J. 1978. Analytical Prediction of Agricultural Aircraft Wakes. ASAE Paper No. 78-1506. Am. Soc. Ag. Eng., St. Joseph, Michigan 49085.

Morris, D.J., C.C. Croom, C.P. vanDam, and B.J. Holmes. 1984. An Experimental and Theoretical Investigation of Deposition Patterns from an Agricultural Airplane. NASA Technical Paper 2348.

Ormsbee, A.I. and M.B. Bragg. 1978. Trajectory Scaling Laws for a Particle Injected into the Wake of an Aircraft. ARL 78-1 . Aviation Research Laboratory, University of Illinois at Urbana-Champaign.

Prandtl, L. 1921. Applications of Modern Hydrodynamics to Aeronautics. NACA Report No. 116.

Ranz, W.E. and W.R. Marshall. 1952a. Evaporation from droplets, Part I. Chemical Engineering Progress 48(3):141-146.

Ranz, W.E. and W.R. Marshall. 1952b. Evaporation from droplets, Part II. Chemical Engineering Progress 48(4):173-180.

Roberts, L. 1983. On the Structure of the Turbulent Vortex. AGARD CP 342. Aerodynamics of Vortical Type Flows in Three Dimensions. Advisory Group for Aeronautical Research and Development, NATO, France.

Sjenitzer, F. 1952. Spray drying. Chemical Engineering Science 1(3):101-117.

Spreiter, J.R. and A.J. Sacks. 1951. The rolling up of the trailing vortex sheet and its effect on the downwash behind wings. Journal of the Aeronautical Sciences 18(1):21-23.

Trayford, R.S. and L.W. Welch. 1977. Aerial spraying: A simulation and recovery of liquid droplets. Journal of Agricultural Engineering Research 22:183-196.

Walton, L.R. and J.N. Walker. 1970. The trajectory of an evaporating water droplet falling in an airstream. Trans ASAE 13:158-161,167.

Yates, J.E. 1974. Calculation of initial vortex roll up in aircraft wakes. Journal of Aircraft 11(7): 397-400.

Yates, W.E., R.E. Cowden, and N.B. Akesson. 1985. Drop size spectra from nozzles in high-speed airstreams. Trans ASAE 28(2):405-410,414.

9. APPENDIX I. WAKE MODEL OPERATION

It should be noted that in order to model the left wing, the sign of both the swirl coefficient and crosswind was changed, then the model data was input just as for the right wing. This was due to the method in which the model was developed, which prevents the initial location of the drop from being left of centerline (in the negative coordinate direction).

A listing of the program WAKE77 is included after the explanation of the program input and output.

Program Input

The program WAKE77 is written such that all of the required input can be entered in an interactive fashion when the program is executed. As the program runs, the user is prompted for the necessary information regarding the aircraft operation, atmospheric conditions and the droplet statistics.

Alternately, the user may input the needed information via two data files. The first file, that labelled AIRCRAFT, contains the information about the aircraft and its operation. The second file is called DROPINFO and contains the required information regarding the droplet, and pertinent atmospheric conditions. During the program run the user will be queried to determine whether the input will be via the input files or interactively.

The following two example input files show the order of the files. The explanation of terms as given below is not part of the file.

AIRCRAFT

26689	Weight of aircraft, N
61.77	Velocity of aircraft, m/s
3.048	Height of aircraft wing trailing edge at the centerline,m
0	Slope of the ground, m/100m
12.625	Span, m
2.286	Chord, m
3.5	Dihedral per wing, degrees
94	Percent span initial vortex separation
0.0775	Core coefficient
0.6096	Propeller height above wing root at centerline, m
2.7432	Propeller diameter, m
1300	Propeller RPM
0.0040	Swirl coefficient for propeller
-0.4572	Z Position of nozzle from t.e. of wing, m
-0.3048	Y Position of nozzle from t.e. of wing, m

DROPINFO

0	Measured cross wind, m/s
3.048	Measured cross wind height, m
50	Width of spray collection strip, m
0	Final Height, m
0	Slope of collector array, m/100m
0.3048	Physical Roughness height, m
0	Wet Bulb Depression, °C
300	Drop diameter, microns
276000	Nozzle pressure, Pa

Program Output

A complete file of data regarding the conditions for the particular run is written into a temporary file called -SPRAYINFO. Also included in that file is the final drop position for each droplet.

The user is queried as to whether droplet trajectories for individual droplets are desired. If the response is affirmative, the temporary file -DROPDATA will have the information regarding the individual droplet trajectories written to it.

Another temporary file -SWATHDIST has written into it the droplet distribution over the entire swath width. This information will be recorded both on the basis of the remaining volume of the droplets as well as the number of the droplets at the spanwise stations.

Lastly the temporary file -UNIFORMTY has written to it the distributions for varying widths of adjacent passes. At the end of the file is given the distance between adjacent passes for the most uniform distribution. Also given is the measure of uniformity, the amount of spray (both number of drops and volume of spray) which is not lost from recovery area, and the amount of spray not evaporated.

Program Listing

A listing of the program WAKE77 starts on the next page.

```
*        *****************************************************************
*        SIMULATION OF AIRCRAFT WAKE - PARTICLE INTERACTION
*        for M.Sc. Thesis                Filename = WAKE77
*        CRAIG S. MERKL, B.Sc.                  March 29, 1989
*        University of Alberta
*           The model presents the vortex system describing the aircraft
*             and the vortex images for the ground plane, the crosswind,
*             the spray nozzle emission velocity, and the aircraft velocity.
*           The spray droplet size, etc. is entered, then the program
*             determines the resulting motion of the droplet in the wake,
*             and the final position of the droplet, hence the distribution
*             of spray.
*        Language:  FORTRAN 77
*
*        SUBROUTINEs included:       INISHL
*                                           CONSTS
*                                           INPUTS
*                                           INITAL
*                                           INDVEL
*                                           NZVELS
*                                           DRPLYF
*                                           TRMVEL
*                                           EVPDIA
*                                           RUNKUT
*                                           VRTCOR
*                                           DRAGCF
*                                           PRNFIL
*                                           PNTOUT
*                                           CHDELT
*                                           NUMVOL
*                                           VOLDST
*                                           NOZLOC
*                                           NOZZLE
*
*        Input Files:            AIRCRAFT      aircraft and operation data
*                                                     (UNIT=12)   see INPUTS
*                                     DROPINFO     droplet information, collection
*                                           locations, atmospheric conditions, etc
*                                                     (UNIT=13)   see INPUTS
*        Output Files:           -DROPDATA    for individual drop trajectories
*           (temporary files)                   (UNIT=10)
*                                     -SPRAYINFO   for information for spraying
*                                            conditions, etc and final drop locations
*                                                     (UNIT=11)
*                                     -SWATHDIST spray distribution for one pass
*                                                     (UNIT=8)
*                                     -UNIFORMTY gives uniformity measure of
*                                            combined passes for each swath width
*                                                     (UNIT=9)
*
*        *****************************************************************
         PROGRAM  WAKE
*        ================================================================
*
         CHARACTER*1    ADJSWA,DRPOUT,INPERR,NOZLCN,NOZTYP
         CHARACTER*3    NAME3,NAME4
         CHARACTER*8    NAME5,NAME6
         CHARACTER*9    NAME1
         CHARACTER*10  NAME2,NAME7,NAME8
         INTEGER  CHK,DIVS,DIV1S,DRPRUN,LCCNTR,LFTNOZ,LFTWNG,RITNOZ,SWACNT
         REAL  ADJNUM(152),ADJVOL(152),LOCXX(152),
     +        NUMB(152),VOL(152),
     +        ACELX,ACELY,ACELZ,
     +        CD,CFXWND,CHORD,CONST1,CORCOF,CORRAD,CRCLAT,
     +        DELT,DENAIR,DENDRP,DIA,DIAMIC,DIHDRL,DNPRNZ,DRPLIF,DRPNUM,
```

```
     +        ENZVEL,FHIGHT,HEIGHT,HT,HTCOLA,INIDIA,INISEP,INIVOL,
     +        INIVLX,INIVLY,INIVLZ,INILOX,INILOY,INILOZ,
     +        KINVSC,LOCX,LOCY,LOCZ,MSXWND,MSXWNI,MSXWHT,
     +        NOZNUM,NOZX,NOZY,NOZZ,NZPRES,NZVELX,NZVELY,NZVELZ
      REAL  PCNSPN,PI,PINDIA,POSIZ,POSIY,PREVRE,
     +        PRPDIA,PRPHIT,PRPRPM,PSVSEP,RE,RELAST,SLCOLA,
     +        SLPANG,SLPGND,SLPGNI,SPAN,SPRWID,SRFHIT,SWRCOF,SWRROT,
     +        SXWHT,TIME,UEPVTL,
     +        VELREL,VELX,VELY,VELZ,VELXVT,VELYVT,VELZVT,VLPLAN,VLXWND,
     +        VLXREL,VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLXVT7,VLXVT8,
     +        VLYREL,VLYVT5,VLYVT6,VLZREL,VLZVT1,VLZVT2,
     +        VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,VOLUM,
     +        VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +        VRTLY1,VRTLY2,VRTLY3,VRTLY4,
     +        VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8,
     +        VX,VY,VZ,WEIGHT,WTBDEP,X,Y,Z
*
        NAME1='-DROPDATA'
        NAME2='-SPRAYINFO'
        NAME3='NEW'
       OPEN  (UNIT=10,FILE=NAME1,STATUS=NAME3)
       OPEN  (UNIT=11,FILE=NAME2,STATUS=NAME3)
*
 10     CALL  INISHL(DRPRUN,DRPNUM,LCCNTR,FRPRNT,
     +      LFTWNG,SWACNT,UEPVTL,INIVOL)
*             Initialize variables NOT in repeated part of program
*
 330    CALL  NOZLOC(DRPNUM,DRPRUN,LCCNTR,LFTNOZ,LFTWNG,
     +                            NOZLCN,NOZNUM,PCNSPN,RITNOZ,SWACNT)
*                        Nozzle location along span
*
 340    CALL  NOZZLE(DNPRNZ,DRPNUM,DRPRUN,ENZVEL,NOZTYP,NOZX,NOZY,NOZZ)
*                        INIT.VEL - nozzle release directions
*
 350    CALL  CONSTS(CD,CHK,CL,DENAIR,DENDRP,GRAV,KINVSC,
     +        PI,PRNT,RE,TIMCHK,TIME,UNIFRM,VLSQSM,VOLAVE,VOLOST,VOLSUM)
*                        Constants and Coefficients
*
      CALL  INPUTS(ADJSWA,CHORD,CORCOF,DELT,DENDRP,DIA,DIAMIC,
     +              DIHDRL,DRPOUT,DRPRUN,ENZVEL,HEIGHT,HT,HTCOLA,
     +              INIDIA,INISEP,INPERR,LFTWNG,MSXWND,MSXWNI,MSXWHT,
     +              NOZTYP,NOZX,NOZY,NOZZ,NZPRES,NZVELX,NZVELY,
     +              NZVELZ,POSIY,POSIZ,PRPDIA,PRPHIT,PRPRPM,PSVSEP,
     +              SLCOLA,SLPGND,SLPGNI,SPAN,SPRWID,SRFHIT,SWACNT,SWRCOF,
     +              SWRROT,VLPLAN,WEIGHT,WTBDEP)
*                        Input values
       IF  (INPERR.EQ.'Y'.OR.INPERR.EQ.'y')  GOTO  10
*
      CALL  INITAL(ADJSWA,CFXWND,CHORD,CONST1,CORCOF,CORRAD,
     +       CRCLAT,DELT,DENAIR,DENDRP,DIA,DIAMIC,DIHDRL,DRPLIF,DRPNUM,
     +       DRPRUN,HEIGHT,HT,HTCOLA,INIDIA,INISEP,INILOX,INILOY,INILOZ,
     +       INIVLX,INIVLY,INIVLZ,KINVSC,LFTWNG,LOCX,LOCY,LOCZ,
     +       MSXWHT,MSXWND,NOZX,NOZY,NOZZ,NZPRES,NZVELX,NZVELY,NZVELZ,
     +       PCNSPN,POSIY,POSIZ,PREVRE,PRPDIA,PRPHIT,PRPRPM,PSVSEP,
     +       RE,RELAST,SEP,SLCOLA,SLPANG,SLPGND,SPAN,SPRWID,SRFHIT,
     +       SWRCOF,SWRROT,TIME,VELX,VELY,VELZ,VX,VY,VZ,
     +       VLPLAN,VLXWND,VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +       VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8,WEIGHT,WTBDEP)
*                        Initial conditions
*
       IF  (LOCX.LT.0)  GOTO  590
*
      IF  (DRPOUT.EQ.'n'.OR.DRPOUT.EQ.'N')  GOTO  410
*                                                      also see 4044
      IF  (DRPRUN.GT.0)  GOTO  380
```

```
         WRITE  (10,370)
 370       FORMAT  ('   LOCX          LOCZ          LOCY          TIME          VRTLX1       VRT
      +LZ1          DIA    ')
 380       WRITE  (10,390)  LOCX,LOCZ,LOCY,TIME,VRTLX1,VRTLZ1,DIA
*                                                       data file for single drops
 390       FORMAT  (6(F8.4,2X),E11.4)
*
*        ============================
*
 410     CALL  RUNKUT(ACELX,ACELY,ACELZ,CD,CFXWND,CHK,CONST1,
      +     CORRAD,CRCLAT,DELT,DIA,DRPLIF,DRPNUM,DRPOUT,FHIGHT,
      +     FRPRNT,HEIGHT,HTCOLA,INIDIA,KINVSC,LCCNTR,LOCX,LOCY,LOCZ,
      +     PRPDIA,RE,RELAST,
      +     SEP,SLCOLA,SLPANG,SRFHIT,SWACNT,SWRROT,TIMCHK,TIME,
      +     VELREL,VELX,VELY,VELZ,VELXVT,VELYVT,VELZVT,VLXWND,VLPLAN,
      +     VLXREL,VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLXVT7,VLXVT8,
      +     VLYREL,VLYVT5,VLYVT6,VLZREL,
      +     VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
      +     VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
      +     VRTLY1,VRTLY2,VRTLY3,VRTLY4,
      +     VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8,
      +     VX,VY,VZ,X,Y,Z)
*                       Simulation  routine  (Runge-Kutta  fourth  order)
*
              IF  (DRPOUT.EQ.'n'.OR.DRPOUT.EQ.'N')  GOTO  440
*                                                       see  4044
*
         IF  (TIME.GE.PRNT)  THEN
          CALL  PRNFIL(ACELX,ACELY,ACELZ,CD,DELT,
      +        PRNT,RE,TIME,VELXVT,VELYVT,VELZVT,
      +        VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
      +        VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
      +        VLXREL,VLYREL,VLZREL,VELREL,
      +        VRTLX1,VRTLZ1,VRTLX2,VRTLZ2,
      +        VRTLX3,VRTLZ3,VRTLX4,VRTLZ4,
      +        VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
      +        VX,VY,VZ,X,Y,Z)
*                                 Print  intermediate  results  to  file
         ENDIF
*
 440     CALL  CHDELT(CHK,DELT,DIA,RE)
*                       Subroutine  to  Modify  time  increment,  DELT
              IF  (TIME.GE.(20.0))  GOTO  480
*                                 End  droplet  simulation  at  20  sec
 460          IF  (DIA.LT.(0.5*INIDIA))  THEN
                    PRINT*,  '  DROP  at  half  INItial  DIAmeter  '
                    GOTO  480
                ENDIF
 470     IF  (LOCZ.GT.FHIGHT)  GOTO  410
*         ==============================
 480          IF  (DRPOUT.EQ.'n'.OR.DRPOUT.EQ.'N')  GOTO  520
*
 510     CALL  PRNFIL(ACELX,ACELY,ACELZ,CD,DELT,
      +        PRNT,RE,TIME,VELXVT,VELYVT,VELZVT,
      +        VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
      +        VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
      +        VLXREL,VLYREL,VLZREL,VELREL,
      +        VRTLX1,VRTLZ1,VRTLX2,VRTLZ2,
      +        VRTLX3,VRTLZ3,VRTLX4,VRTLZ4,
      +        VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
      +        VX,VY,VZ,X,Y,Z)
*                           PRINT  last  set  of  results  to  -SPRAYINFO
*
 520     CALL  PNTOUT(CD,DELT,DIA,DRPNUM,DRPRUN,
      +                 INILOX,INILOY,INILOZ,INIVLX,INIVLY,INIVLZ,
```

```
      +                 LCCNTR,LOCX,LOCY,LOCZ,RE,SWACNT,TIME)
*                Closing statement of droplet results to file
*
 530        DRPRUN = DRPRUN + 1
*                         number of drops, or times thru program
 540        PINDIA = INIDIA
*                         previous initial diameter, see 9025
*
 550     CALL NUMVOL(ADJNUM,ADJVOL,DIA,DIVS,DIV1S,DRPRUN,INIDIA,INIVOL,
      +                 LOCX,LOCXX,LOCZ,NUMB,SPRWID,SWACNT,TIME,UEPVTL,VOL)
*                        Volume distribution subroutine
*
 570     CALL NOZZLE(DNPRNZ,DRPNUM,DRPRUN,ENZVEL,NOZTYP,NOZX,NOZY,NOZZ)
*                        Initial Velocity - nozzle release directions
            IF (DRPNUM.LE.DNPRNZ) GOTO 350
*
 580     CALL NOZLOC(DRPNUM,DRPRUN,LCCNTR,LFTNOZ,LFTWNG,
      +                          NOZLCN,NOZNUM,PCNSPN,RITNOZ,SWACNT)
*                        nozzle locations on wing
*
         IF (NOZLCN.EQ.'E'.OR.NOZLCN.EQ.'e') THEN
            IF (LCCNTR.LE.NOZNUM) GOTO 340
           ELSE
            GOTO 590
           ENDIF
*
         IF (NOZLCN.EQ.'U'.OR.NOZLCN.EQ.'u') THEN
           IF (LFTWNG.EQ.0) THEN
               IF (LCCNTR.LE.RITNOZ) THEN
                 GOTO 340
                ELSE
                 GOTO 590
                ENDIF
            ENDIF
           IF (LFTWNG.EQ.1) THEN
               IF (LCCNTR.LE.LFTNOZ) THEN
                 GOTO 340
                ELSE
                 GOTO 590
                ENDIF
            ENDIF
          ENDIF
*
 590     SWACNT = SWACNT + 1
 600    IF (SWACNT.GT.2) THEN
            IF (ADJSWA.EQ.'O'.OR.ADJSWA.EQ.'o') THEN
                 GOTO 640
                ELSE
                 GOTO 690
             ENDIF
          ENDIF
*                 now do LEFT wing
 620     LFTWNG = 1
           LCCNTR = 0
           DRPNUM = 0
         IF ((NOZLCN.EQ.'U'.OR.NOZLCN.EQ.'u').AND.(LFTNOZ.EQ.0)) GOTO 590
           GOTO 330
*
*       -- if (SWACNT .GE.3) then opposite direction adjacent passes done
 640    IF (SWACNT.EQ.3) THEN
           LFTWNG = 1
           LCCNTR = 0
           DRPNUM = 0
         IF ((NOZLCN.EQ.'U'.OR.NOZLCN.EQ.'u').AND.(LFTNOZ.EQ.0)) GOTO 590
           GOTO 330
```

```
          ENDIF
  650   IF (SWACNT.EQ.4) THEN
            LFTWNG = 0
          . LCCNTR = 0
            DRPNUM = 0
         IF ((NOZLCN.EQ.'U'.OR.NOZLCN.EQ.'u').AND.(RITNOZ.EQ.0)) GOTO 690
            GOTO 330
          ENDIF
*
*
 690     CALL VOLDST(ADJNUM,ADJSWA,ADJVOL,DIVS,DIV1S,INIVOL,LOCXX,
     +             NOZTYP,NUMB,SPRWID,UEPVTL,VOL)
*                        calculate UNIFoRMity with overlap, etc
*
          END
*       ******************************************************************
*       ======= End of Main Program =====================================
*       ******************************************************************
*
*
*
*
*       ***** INISHL ******** 900 ****************************************
*       Subroutine to Initialize variables not in repeated part of program
*       ==================================================================
*
          SUBROUTINE INISHL(DRPRUN,DRPNUM,LCCNTR,FRPRNT,
     +       LFTWNG,SWACNT,UEPVTL,INIVOL)
*
           INTEGER  DRPRUN,LCCNTR,LFTWNG,SWACNT
           REAL  DRPNUM,FRPRNT,UEPVTL,INIVOL
*
           DRPNUM = 0
*                    see 18060, # of drops currently done per nozzle
           DRPRUN = 0
*                  see 15040 ,# of drops or times thru program
           FRPRNT = 0.1
*                    frequency of output to data file (sec) see11080
           LCCNTR = 0
*                    see 17060, # of nozzles currently done
           LFTWNG = 0
*                    Right wing, LFTWNG done if LFTWNG =1  see 620
           SWACNT = 1
*            opposite direction adjacent swath done if .GE. 3  see 620
*                                      see also 17535-700, 4920, 15090
           UEPVTL = 0
*                    Total unevaporated volume from final drop diameters
           INIVOL = 0
*                    Total volume based on initial diameters
          RETURN
          END
*         =================================================================
*
*
*          ****** CONSTS ********* 1000 **********************************
*          Constants and Coefficients
*          ===============================================================
*
          SUBROUTINE CONSTS(CD,CHK,CL,DENAIR,DENDRP,GRAV,KINVSC,
     +       PI,PRNT,RE,TIMCHK,TIME,UNIFRM,VLSQSM,VOLAVE,VOLOST,VOLSUM)
*
          INTEGER CHK
          REAL CD,CL,DENAIR,DENDRP,DYVISC,GRAV,KINVSC,
     +       PI,PRNT,RE,TIMCHK,TIME,UNIFRM,VLSQSM,VOLAVE,VOLOST,VOLSUM
*
```

```
      PRNT = -0.00001
      CHK=0
*                               used as counters, see 430, 12130, and 13060
      CD = 0
*                     drag coefficient of droplet
*     CHORD      chord of aircraft wing, m
      CL = 0
*                     lift coefficient of wing
*     CRCLAT                    circulation around wing, m**2/s
      DENAIR = 1.2256
*                            density of air, kg/m**3   (Sea Level)
      DENDRP = 1000.0
*                            density of droplet, kg/m**3
*     DIA                     diameter of droplet, m
*     DIAMIC                 Initial DIAmeter of droplet in microns
*     DIVS       number of lateral divisions for spray deposition
      DYVISC = 0.0000178
*                            dynamic viscosity of air, N*s/m**2 (Sea Level)
*     FHIGHT                    Final Height where drop hits ground
      GRAV = 9.80665
*                            acceleration of gravity, m/s**2
*     HEIGHT                    height of Trailing Edge of wing ROOT AGL, m
*     HT = HEIGHT + POSIZ       height of nozzle above ground, m
      KINVSC = DYVISC / DENAIR
*                            kinematic viscosity of air
*     LIFT                   lift of wing, N
*     M                        mass of droplet, kg
*     NZPRES      Nozzle pressure, Pa   (but entered in kPa)
*     PCNSPN     percent span location of nozzle
      PI = 3.141593
*     POSIZ      POSIY        position of nozzle below and behind wing, m
*     PSHGHT     Physical height of surface covering (m) (nonzer0)
*
      RE = 0
*                     Reynolds number of droplet
*     RELAST     Reynolds number of droplet previous loop
*     SEP         half separation between trailing vortices, m
*     SPAN       wingspan of aircraft wing, m
*     SPRWID          Width of spray collection strip (m, Max.= 75m)
      TIMCHK = 0
*                     time check for file data output, every FRPRNT sec
      TIME = 0
*                     time from emission of droplet, sec
      UNIFRM = 0
*                     UNIFoRMity coefficient
      VLSQSM = 0
*                     sum of squared spray volume that lands
      VOLAVE = 0
*                     average spray volume that lands per division,m**3
      VOLSUM = 0
*                     sum of spray volume that lands, m**3
      VOLOST = 0
*                     amount of spray that doesn't land, m**3
*     WEIGHT                 weight of airplane, N
*     WTBDEP      Wet bulb depression, degrees C
*
*
*     Distance definitions and Initialization of variables
*     ====================================================
*     LOCX                          location of drop on X-axis, m
*     LOCY                          location of drop on Y-axis, m
*     LOCZ                          location of drop on Z-axis, m
*
*     R1                             distance from drop to vortex core #1, m
*     R2                             distance from drop to vortex core #2, m
```

```
*        R3                          distance from drop to vortex core #3, m
*        R4                          distance from drop to vortex core #4, m
*        R5                          distance from drop to vortex core #5, m
*        R6                          distance from drop to vortex core #6, m
*        R7                          distance from drop to slipstream center, m
*
*        X1                          local x component for vortex 1, m
*        X2                          local x component for vortex 2, m
*        X3                          local x component for vortex 3, m
*        X4                          local x component for vortex 4, m
*        X5                          local x component for vortex 5, m
*        X6                          local x component for vortex 6, m
*        X7                          local x component for slipstream, m
*        X8                          local x component for slipstream image, m
*        Y1                          local y component for vortex 1, m
*        Y2                          local y component for vortex 2, m
*        Y3                          local y component for vortex 3, m
*        Y4                          local y component for vortex 4, m
*        Y5                          local y component for vortex 5, m
*        Y6                          local y component for vortex 6, m
*        Y7                          local y component for slipstream, m
*        Z1                          local z component for vortex 1, m
*        Z2                          local z component for vortex 2, m
*        Z3                          local z component for vortex 3, m
*        Z4                          local z component for vortex 4, m
*        Z5                          local z component for vortex 5, m
*        Z6                          local z component for vortex 6, m
*        Z7                          local z component for slipstream, m
*        Z8                          local z component for slipstream image, m
*
*        velocity definitions and initialization
*        =======================================
*        VELX                  actual x velocity of drop, m/s
*        VELY                  actual y velocity of drop, m/s
*        VELZ                  actual z velocity of drop, m/s
*
*        NZVELX               initial x vel of fluid relative to nozzle, m/s
*        NZVELY               initial y vel of fluid relative to nozzle, m/
*        NZVELZ               initial z vel of fluid relative to nozzle, m/s
*        VLPLAN               velocity of plane (along y axis), m/s
*        VLXWND               x component of wind (crosswind), m/s
*        VLYWND               y component of wind, m/s
*
*        VLXVT1               x component of induced velocity of vortex 1, m/s
*        VLXVT2               x component of induced velocity of vortex 2, m/s
*        VLXVT3               x component of induced velocity of vortex 3, m/s
*        VLXVT4               x component of induced velocity of vortex 4, m/s
*        VLYVT5               y component of induced velocity of vortex 5, m/s
*        VLYVT6               y component of induced velocity of vortex 6, m/s
*        VLZVT1               z component of induced velocity of vortex 1, m/s
*        VLZVT2               z component of induced velocity of vortex 2, m/s
*        VLZVT3               z component of induced velocity of vortex 3, m/s
*        VLZVT4               z component of induced velocity of vortex 4, m/s
*        VLZVT5               z component of induced velocity of vortex 5, m/s
*        VLZVT6               z component of induced velocity of vortex 6, m/s
*
*        VELXVT               resultant induced x velocity, m/s
*        VELYVT               resultant induced y velocity, m/s
*        VELZVT               resultant induced z velocity, m/s
*
*        VLXREL               relative x velocity, drop to air, m/s
*        VLYREL               relative y velocity, drop to air, m/s
*        VLZREL               relative z velocity, drop to air, m/s
*        VELREL               resultant relative velocity, drop to air, m/s
*
```

```
*
*       acceleration definitions and initialization
*       ==========================================
*       ACELX              x acceleration of drop, m/s**2
*       ACELY              y acceleration of drop, m/s**2
*       ACELZ              z acceleration of drop, m/s**2
*
*       angle definitions and initialization
*       ===================================
*       BETA1              angle from drop to end of vortex 1
*       BETA2              angle from drop to end of vortex 2
*       BETA3              angle from drop to end of vortex 3
*       BETA4              angle from drop to end of vortex 4
*       BETA5A             angle from drop to one end of vortex 5
*       BETA5B             angle from drop to other end of vortex 5
*       BETA6A             angle from drop to one end of vortex 6
*       BETA6B             angle from drop to other end of vortex 6
*
*       ALPHA1             angle from horizontal plane of vortex 1 to drop
*       ALPHA2             angle from horizontal plane of vortex 2 to drop
*       ALPHA3             angle from horizontal plane of vortex 3 to drop
*       ALPHA4             angle from horizontal plane of vortex 4 to drop
*       ALPHA5             angle from horizontal plane of vortex 5 to drop
*       ALPHA6             angle from horizontal plane of vortex 6 to drop
*
        RETURN
        END
*       ============================================================
*
*
*
*
*
*         ******* INPUTS ******* 4000 ***********************************
*         SUBROUTINE for Input values
*         ==============================================================
*
*      Input Files:          AIRCRAFT      aircraft and operation data
*       (if chosen for)                      (UNIT=12)
*       (input method)       DROPINFO      droplet information, collection
*                                          locations, atmospheric conditions, etc
*                                                    (UNIT=13)
*
        SUBROUTINE INPUTS(ADJSWA,CHORD,CORCOF,DELT,DENDRP,DIA,DIAMIC,
     +             DIHDRL,DRPOUT,DRPRUN,ENZVEL,HEIGHT,HT,HTCOLA,
     +             INIDIA,INISEP,INPERR,LFTWNG,MSXWND,MSXWNI,MSXWHT,
     +             NOZTYP,NOZX,NOZY,NOZZ,NZPRES,NZVELX,NZVELY,
     +             NZVELZ,POSIY,POSIZ,PRPDIA,PRPHIT,PRPRPM,PSVSEP,
     +             SLCOLA,SLPGND,SLPGNI,SPAN,SPRWID,SRFHIT,SWACNT,SWRCOF,
     +             SWRROT,VLPLAN,WEIGHT,WTBDEP)
*
        CHARACTER*1 ADJSWA,AIRDAT,DRPDAT,DRPOUT,INPERR,NOZTYP,UNITS
        CHARACTER*3  NAME4
        CHARACTER*8  NAME5,NAME6
        INTEGER DRPRUN,LFTWNG,SWACNT
        REAL CHORD,CORCOF,DELT,DENDRP,DIA,DIAMIC,DIHDRL,ENZVEL,
     +       GRAV,HEIGHT,HT,HTCOLA,INIDIA,INISEP,MSXWND,MSXWNI,MSXWHT,
     +       NOZX,NOZY,NOZZ,NZCOEF,NZPRES,NZVELX,NZVELY,NZVELZ,
     +       PI,POSIY,POSIZ,PRPDIA,PRPHIT,PRPROT,PRPRPM,PSHGHT,PSVSEP,
     +       SLCOLA,SLPGND,SLPGNI,SPAN,SPRWID,SRFHIT,SWRCOF,SWRROT,
     +       VLPLAN,WEIGHT,WTBDEP
*
          GRAV = 9.80665
          PI = 3.141593
*
```

```
      IF (DRPRUN.GT.0) GOTO 380
      PRINT*, 'Droplet output takes lots of extra time and disk.'
      PRINT*, ' Hence maximum 20 drops!'
44    PRINT*, 'Do you want output for drop trajectories (Y/N)'
      PRINT*, ' Be sure to enclose letter with apostrophes!'
      READ*, DRPOUT
       IF (DRPOUT.EQ.'N'.OR.DRPOUT.EQ.'n'.OR.DRPOUT.EQ.'y'.OR.
     +                              DRPOUT.EQ.'Y') THEN
          GOTO 50
         ELSE
          GOTO 44
       ENDIF
50    PRINT*, 'Are there any errors in INPUT yet ? (Y/N)'
      PRINT*, ' Be sure to enclose letter with apostrophes!'
      READ*, INPERR
      IF (INPERR.EQ.'Y'.OR.INPERR.EQ.'y') GOTO 80
      IF (INPERR.EQ.'N'.OR.INPERR.EQ.'n') THEN
          GOTO 90
         ELSE
          GOTO 50
       ENDIF
80    PRINT*, '** Redo Locations and Nozzles INPUT from start! ***'
         RETURN
*
90    PRINT*, 'Adjacent passes are in the Same or Opposite directions
     + (S/O)'
      PRINT*, ' Be sure to enclose letter with apostrophes!'
      READ*, ADJSWA
      IF (ADJSWA.EQ.'s'.OR.ADJSWA.EQ.'S'.OR.ADJSWA.EQ.'o'.OR.
     +                                    ADJSWA.EQ.'O') THEN
          GOTO 100
         ELSE
          GOTO 90
       ENDIF
100   PRINT*, 'SI units or Imperial --- (S or I) '
      PRINT*, ' Be sure to enclose letter with apostrophes!'
      READ*, UNITS
      IF (UNITS.EQ.'S'.OR.UNITS.EQ.'s'.OR.UNITS.EQ.'I'.OR.
     +                           UNITS.EQ.'i') THEN
          GOTO 120
         ELSE
          GOTO 100
       ENDIF
*
*     ----------Input regarding the aircraft and operation
120   PRINT*, 'Is aircraft data from file   AIRCRAFT   (Y/N) '
      PRINT*, ' Be sure to enclose letter with apostrophes!'
      READ*, AIRDAT
         NAME4='OLD'
         NAME5='AIRCRAFT'
      IF (AIRDAT.EQ.'n'.OR.AIRDAT.EQ.'N') GOTO 145
      IF (AIRDAT.EQ.'y'.OR.AIRDAT.EQ.'Y') THEN
          OPEN (UNIT=12,FILE=NAME5,STATUS=NAME4)
         ELSE
          GOTO 120
       ENDIF
      READ (12,*) WEIGHT,VLPLAN,HEIGHT,SLPGND,SPAN,CHORD,DIHDRL,
     +      PSVSEP,CORCOF,PRPHIT,PRPDIA,PRPRPM,SWRCOF,POSIZ,POSIY
       GOTO 260
145   PRINT*, 'The weight of the plane (N or lb) is '
        READ*, WEIGHT
      IF (UNITS.EQ.'i'.OR.UNITS.EQ.'I') WEIGHT=WEIGHT*0.453592*GRAV
       PRINT*, 'The speed of the plane (m/s or knots) is '
        READ*, VLPLAN
      IF (UNITS.EQ.'i'.OR.UNITS.EQ.'I') VLPLAN=
```

```
     +                                   (VLPLAN*6080.0/3600.0)*0.3048
        PRINT*, 'Height of wing root trailing edge (m or ft) AGL is '
          READ*, HEIGHT
      IF (UNITS.EQ.'i'.OR.UNITS.EQ.'I')  HEIGHT=HEIGHT*0.3048
        PRINT*, 'What is percentage slope of ground at centerline?'
        PRINT*, '           (positive is down off right wing)'
          READ*, SLPGND
            SLPGNI = SLPGND
        PRINT*, 'The wingspan of the plane (m or ft) is '
          READ*, SPAN
      IF (UNITS.EQ.'i'.OR.UNITS.EQ.'I')  SPAN=SPAN*0.3048
        PRINT*, 'The wing chord of the plane (m or ft) is '
          READ*, CHORD
      IF (UNITS.EQ.'i'.OR.UNITS.EQ.'I')  CHORD=CHORD*0.3048
        PRINT*, 'The wing DIHeDRaL (degrees per wing) is '
          READ*, DIHDRL
        PRINT*, 'The percent span initial vortex separation ='
          READ*, PSVSEP
        PRINT*, 'The default value of CORCOF = 0.0775 '
            CORCOF = 0.0775
        PRINT*, 'Nozzle Z location w.r.t. trailing edge   (m) ='
        PRINT*, '          (upwards is positive)'
          READ*, POSIZ
        PRINT*, 'Nozzle Y location w.r.t. trailing edge   (m) = '
        PRINT*, '                    (forward is positive) '
          READ*, POSIY
*
*
*       ----------Droplet and movement information
  260  PRINT*, 'Is Droplet data, etc. from file DROPINFO   (Y/N) '
        PRINT*, ' Be sure to enclose letter with apostrophes!'
        READ*, DRPDAT
          NAME4='OLD'
          NAME6='DROPINFO'
      IF (DRPDAT.EQ.'n'.OR.DRPDAT.EQ.'N') GOTO 280
      IF (DRPDAT.EQ.'y'.OR.DRPDAT.EQ.'Y') THEN
            OPEN (UNIT=13,FILE=NAME6,STATUS=NAME4)
          ELSE
            GOTO 260
        ENDIF
      READ (13,*) MSXWND,MSXWHT,SPRWID,HTCOLA,SLCOLA,PSHGHT,WTBDEP,
     +        DIAMIC,NZPRES
          GOTO 370
  280   PRINT*, 'Strength of the crosswind at 3.048 m (m/s or ft/s)'
          READ*, MSXWND
      IF (UNITS.EQ.'i'.OR.UNITS.EQ.'I') MSXWND = MSXWND * 0.3048
        PRINT*, 'HEIGHT at which the crosswind is measured (m or ft)'
          READ*, MSXWHT
      IF (UNITS.EQ.'i'.OR.UNITS.EQ.'I') MSXWHT = MSXWHT * 0.3048
  290   PRINT*, 'Width of spray collection strip (m, Max.= 75m) = '
            READ*, SPRWID
      IF (SPRWID.GT.75) GOTO 290
        PRINT*, 'What is height of collector array (final height) '
        PRINT*, '                    at the centerline where the drop hits ?'
          READ*, HTCOLA
        PRINT*, 'What is percentage slope of collector array ?'
        PRINT*, '          (positive is down off right wing)'
          READ*, SLCOLA
  320   PRINT*, 'Physical height of surface covering (m) (nonzer0) = ?'
          READ*, PSHGHT
      IF (PSHGHT.LE.0) GOTO 320
*                           p 6  Morris et al.  1984
        PRINT*, 'Wet bulb depression, degrees C '
          READ*, WTBDEP
        PRINT*, 'The initial size of the droplet (microns) is '
```

```
          READ*, DIAMIC
         PRINT*, 'Nozzle pressure in kPa =   ? '
          READ*, NZPRES
           NZPRES = NZPRES * 1000
*
*       =========
 370      MSXWNI = MSXWND
         IF (NOZTYP.EQ.'r'.OR.NOZTYP.EQ.'R') THEN
                NZCOEF = 0.8
*         for NZCOEF=0.8, see subroutine NZVELS, Goering (1972)
                NZPRES = ENZVEL*ENZVEL*DENDRP/(2.0*NZCOEF)
*                    See Subroutine NOZZLE,    Goering (1972)
            ENDIF
*
  380    INIDIA = DIAMIC / 1000000.0
             DIA = INIDIA
        CALL NZVELS(DENDRP,ENZVEL,NOZTYP,NOZX,NOZY,NOZZ,
     +          NZPRES,NZVELX,NZVELY,NZVELZ)
*                 Subroutine for nozzle velocities along coordinate axis
*
*
          HT = HEIGHT + POSIZ
          SRFHIT = PSHGHT / 30.0
*        'Notice 1/30 of physical height of surface covering
*                                             p6 Morris, et al.   1984
          DELT = 0.000625
           IF (DIA.LT.(0.0001)) DELT = DELT/10.0
           IF (DIA.LT.(0.00001)) DELT = DELT/10.0
*
*        ---- Propeller information
          IF (DRPRUN.GT.0) GOTO 740
         IF (AIRDAT.EQ.'y'.OR.AIRDAT.EQ.'Y') GOTO 740
          PRINT*, 'What is Propeller height from trailing edge (m) ?'
         READ*, PRPHIT
          PRINT*, ' What is Propeller diameter (m) = '
         READ*, PRPDIA
          PRINT*, 'What is Propeller RPM =   '
          PRINT*, '    positive is clockwise from pilots view   '
          PRINT*, '          negative if counterclockwise '
         READ*, PRPRPM
          PRINT*, '    What is the SWiRlCOeF =   '
          PRINT*, '(= slipstream angular velocity divided by the'
          PRINT*, '(  angular velocity of the propeller, 0.015 suggested)'
         READ*, SWRCOF
 740    PRPROT = PRPRPM*2.0*PI/60.0
*                    angular velocity of the propeller, rad/s
          SWRROT =(PRPDIA/2)* PRPROT * SWRCOF
*                  angular velocity of the slipstream, rad/s.
*                    (positive is clockwise from pilot'view)
*
*        ---- Vortex information
         IF (AIRDAT.EQ.'y'.OR.AIRDAT.EQ.'Y') GOTO 800
          CORCOF = 0.0775
*          (CORCOF =core radius/span=0.0775   Sprieter & Sacks,1951)
 800    INISEP=PSVSEP*SPAN/100.0
*                                initial vortex separation horiz, m
*
          IF (DRPRUN.GT.0) GOTO 900
 820     PRINT*, 'Are there any errors in INPUT DATA ? (Y/N)'
          PRINT*, ' Be sure to enclose letter with apostrophes!'
          READ*, INPERR
          IF (INPERR.EQ.'Y'.OR.INPERR.EQ.'y') GOTO 850
          IF (INPERR.EQ.'N'.OR.INPERR.EQ.'n') THEN
               GOTO 900
              ELSE
```

```
             GOTO 820
          ENDIF
 850    PRINT*, '*** Redo INPUT DATA from start. ***'
          GOTO 100
*
*        =======================
*       Modification of SWRROT and XWND for runs for other wing and passes
 900    IF (SWACNT.GT.2) GOTO 940
         IF (LFTWNG.NE.0) THEN
              SWRROT = -SWRROT
              MSXWND = -MSXWNI
              SLPGND = -SLPGNI
          ENDIF
 940   IF (SWACNT.EQ.3) SWRROT = -SWRROT
*                              Adjacent swath, left wing
        IF (SWACNT.EQ.4) MSXWND = -MSXWNI
                              SLPGND = -SLPGNI
*                              Adjacent swath, right wing
*        =======================
*
         RETURN
         END
*        ==============================================================
*
*
*
*
*         ******** INITAL ******** 5000 *********************************
*         Initial Conditions
*         ==============================================================
*
*       Output file: (UNIT=11) -SPRAYINFO  for information for spraying
*          (see Main Program)  conditions, etc and final drop locations
*
         SUBROUTINE INITAL(ADJSWA,CFXWND,CHORD,CONST1,CORCOF,CORRAD,
     +        CRCLAT,DELT,DENAIR,DENDRP,DIA,DIAMIC,DIHDRL,DRPLIF,DRPNUM,
     +        DRPRUN,HEIGHT,HT,HTCOLA,INIDIA,INISEP,INILOX,INILOY,INILOZ,
     +        INIVLX,INIVLY,INIVLZ,KINVSC,LFTWNG,LOCX,LOCY,LOCZ,
     +        MSXWHT,MSXWND,NOZX,NOZY,NOZZ,NZPRES,NZVELX,NZVELY,NZVELZ,
     +        PCNSPN,POSIY,POSIZ,PREVRE,PRPDIA,PRPHIT,PRPRPM,PSVSEP,
     +        RE,RELAST,SEP,SLCOLA,SLPANG,SLPGND,SPAN,SPRWID,SRFHIT,
     +        SWRCOF,SWRROT,TIME,VELX,VELY,VELZ,VX,VY,VZ,
     +        VLPLAN,VLXWND,VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +        VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8,WEIGHT,WTBDEP)
*
         CHARACTER*1 ADJSWA
         INTEGER DRPRUN,LFTWNG
         REAL ACELX,ACELY,ACELZ,CD,CFXWND,CHORD,CONST1,CORCOF,CORRAD,
     +          CRCLAT,DELT,DENAIR,DENDRP,DIA,DIAMIC,DIHDRL,DRPLIF,DRPNUM,
     +          HEIGHT,HT,HTCOLA,
     +          INIDIA,INISEP,INILOX,INILOY,INILOZ,INIVLX,INIVLY,INIVLZ,
     +          KINVSC,LFTCOF,LFTCVS,LIFT,LOCX,LOCY,LOCZ,LWNGHT,MSXWND,
     +          MSXWHT,NNELCR,NOZX,NOZY,NOZZ,NZPRES,NZVELX,NZVELY,NZVELZ,
     +          PSVSEP,PCNSPN,PI,PINDIA,POSIY,POSIZ,PREVRE,
     +          PRPDIA,PRPHIT,PRPRPM,RE,RELAST,RWNGHT,SSPGND,
     +          SEP,SLCOLA,SLPANG,SLPGND,SPAN,SPRWID,SRFHIT,SWRCOF,SWRROT,
     +          TIME,VELX,VELY,VELZ,VELXVT,VELYVT,VELZVT,VLPLAN,VLXWND,
     +          VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +          VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +          VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
     +          VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
     +          VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +          VTHFTE,VTHTDI,VX,VY,VZ,WEIGHT,WTBDEP,X,Y,Z
*
           PI = 3.141593
```

```
*              location of droplet at the start
          LOCX = (SPAN/2.0) * PCNSPN/100.0
          IF (LOCX.LT.0) RETURN
          LOCY = -0.75 * CHORD + POSIY
          LOCZ = HT + LOCX * TAN(DIHDRL*PI/180.0)
*                        initial X position of drop
*                            '     Y     '     '     '
*                            '     Z     '     '     '
          INILOX = LOCX
          INILOY = LOCY
          INILOZ = LOCZ
          IF (LFTWNG.EQ.1) INILOX = -INILOX
*          location of drop at the start, see 12620 and 12660
*          velocity of drop at the start, also see 12620
          VELX = NZVELX
          VELY = NZVELY + VLPLAN
          VELZ = NZVELZ
           INIVLX = VELX
           INIVLY = VELY
           INIVLZ = VELZ
*
         CFXWND = MSXWND / LOG(MSXWHT/SRFHIT)
         VLXWND = CFXWND * LOG(LOCZ/SRFHIT)
*
         LFTCOF=WEIGHT*2.0/(DENAIR * VLPLAN * VLPLAN * CHORD * SPAN)
         NNELCR = 0.1
*                  Correction for nonelliptical planform
         LFTCVS = (2.0*PI)/(1.0+2.0*(1.0+NNELCR)/(SPAN/CHORD))
*                    LFTCVS = lift curve slope
         VTHFTE = SIN(LFTCOF/LFTCVS) * CHORD*3/4
*             vortex height w.r.t. the trailing edge of the wing tip
*                                                        due to angle of attack
         CORRAD = CORCOF * SPAN
*                                     vortex core radius
         LIFT = WEIGHT
         CRCLAT = LIFT/(DENAIR * VLPLAN * SPAN)
*                                     lift & circulation of wing
         IF (HEIGHT.GE.SPAN) THEN
           INISEP = 0.82 * SPAN
          ELSE
           INISEP = (PSVSEP-(PSVSEP-82)*(HEIGHT/SPAN))*(SPAN/100.0)
         ENDIF
         SEP = INISEP / 2.0
          VTHTDI = SEP * TAN((DIHDRL*PI)/180)
*        added vortex height w.r.t. the wing root, due to the wing dihedral
          CONST1 = 0.75 * DENAIR / DENDRP
*        initial vortex locations
           VRTLX1 = SEP
           VRTLX2 = -VRTLX1
           VRTLX7= 0
           VRTLZ1 = HEIGHT + VTHFTE + VTHTDI
           VRTLZ2 = VRTLZ1
           VRTLZ7 = HEIGHT + PRPHIT
*        initial vortex image locations
         SSPGND = 1.0
         SSPGND = SIGN(SSPGND,SLPGND)
         SLPANG = ATAN(SLPGND/100)
         RWNGHT = VRTLZ1 + VRTLX1 * SLPGND/100
         LWNGHT = VRTLZ1 + VRTLX2 * SLPGND/100
           VRTLX3 = VRTLX1 - (2*RWNGHT*SIN(SLPANG)*SIN(SLPANG)*SSPGND)
           VRTLX4 = VRTLX2 - (2*LWNGHT*SIN(SLPANG)*SIN(SLPANG)*SSPGND)
           VRTLX8 = VRTLX7 - (2*VRTLZ7*SIN(SLPANG)*SIN(SLPANG)*SSPGND)
           VRTLZ3 = VRTLZ1 - (2*RWNGHT*COS(SLPANG)*COS(SLPANG))
           VRTLZ4 = VRTLZ2 - (2*LWNGHT*COS(SLPANG)*COS(SLPANG))
           VRTLZ8 = VRTLZ7 - (2*VRTLZ7*COS(SLPANG)*COS(SLPANG))
```

```
*
          X=LOCX
          Y=LOCY
          Z=LOCZ
          VX=VELX
          VY=VELY
          VZ=VELZ
*
*
        CALL  INDVEL(ACELX,ACELY,ACELZ,CD,CONST1,CORRAD,CRCLAT,
     +        DIA,HEIGHT,KINVSC,PRPDIA,RE,SEP,SLPANG,SWRROT,TIME,VLPLAN,
     +        VELXVT,VELYVT,VELZVT,VLXWND,
     +        VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +        VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +        VLXREL,VLYREL,VLZREL,VELREL,
     +        VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
     +        VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
     +        VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +        VX,VY,VZ,X,Y,Z)
*                           to get initial Reynolds number
           RELAST = RE
*
       CALL  DRPLYF(CD,DENDRP,DENAIR,DIA,DIAMIC,DRPLIF,INIDIA,
     +           KINVSC,PINDIA,PREVRE,RE,WTBDEP)
*                                Subroutine to determine the DRoP LIFe
*
          IF (DRPRUN.GT.0) GOTO 550
        WRITE (11,10)
 10     FORMAT ('            ')
        WRITE (11,20) WEIGHT
 20     FORMAT ('The WEIGHT of the plane (N)=',F12.2)
        WRITE (11,30) VLPLAN
 30     FORMAT ('The SPEED of the plane (m/s)=',F8.3)
        WRITE (11,40) HEIGHT
 40     FORMAT ('HEIGHT of the wing root trailing edge AGL (m)=',F8.3)
        WRITE (11,45)
        WRITE (11,46) SLPGND
 45     FORMAT ('Percentage slope of ground at centerline, ')
 46     FORMAT (' where positive is down off right wing =',F8.2)
        WRITE (11,50) SPAN
 50     FORMAT ('The wing SPAN of the plane (m)=',F8.3)
        WRITE (11,60) CHORD
 60     FORMAT ('The wing CHORD of the plane (m)=',F8.3)
        WRITE (11,70) DIHDRL
 70     FORMAT ('The wing DIHeDRaL (degrees per wing) is ',F8.3)
        WRITE (11,75) PRPHIT
 75     FORMAT ('Propeller height from trailing edge (m) =',F8.3)
        WRITE (11,80) PRPDIA
 80     FORMAT ('Propeller diameter (m) = ',F8.3)
        WRITE (11,90) PRPRPM
 90     FORMAT ('Propeller RPM (revolutions/minute) = ',F8.2)
        WRITE (11,100) SWRCOF
        WRITE (11,110)
        WRITE (11,120)
 100    FORMAT ('Swirl coefficient    ',F8.5)
 110    FORMAT ('= slipstream angular velocity divided by the')
 120    FORMAT ('                    angular velocity of the propeller')
        WRITE (11,130) PSVSEP
 130    FORMAT ('The percent span initial vortex separation=',F8.3)
        WRITE (11,140) CORCOF
 140    FORMAT ('The CORCOF = core radius/span(Sprieter & Sacks,1951)',
     +        F8.5)
        WRITE (11,150) LFTCOF
 150    FORMAT ('Lift coefficient = ',F8.3)
        WRITE (11,160) VTHFTE
```

```
160       FORMAT ('Vortex height from trailing edge (pos. up)m',F8.4)
          WRITE (11,170) POSIZ,POSIY
170       FORMAT ('Nozzle location w.r.t. trailing edge(z,y)',2(F8.4))
          WRITE (11,180) NOZX,NOZY,NOZZ
180       FORMAT ('Nozzle direction (NOZX,NOZY,NOZZ)',3(F8.4))
          WRITE (11,190) NZPRES
190       FORMAT ('Nozzle pressure (Pa) = ',F12.0)
          WRITE (11,200) DELT
200       FORMAT ('Initial Time increment = ',F10.7)
          WRITE (11,210) DIA
210       FORMAT ('Initial drop diameter (m) = ',E11.4)
          WRITE (11,215) DENDRP
215       FORMAT ('Drop density (kg/m**3) = ',F8.2)
          WRITE (11,220) WTBDEP
220       FORMAT ('WET BULB DEPression = ',F8.2)
          WRITE (11,225) DRPLIF
225       FORMAT ('Expected DRoP LIFe  (sec) = ',E11.4)
          WRITE (11,230)
          WRITE (11,240) MSXWND
230       FORMAT ('Crosswind comp at designated height (m)')
240       FORMAT ('                        (pos from left wing, m/s) = ',F8.3)
          WRITE (11,250) MSXWHT
250       FORMAT ('HEIGHT at which the crosswind is measured (m)',F8.3)
          WRITE (11,252)
          WRITE (11,253) HTCOLA
252       FORMAT ('Height of collector array (final height) at the')
253       FORMAT ('  centerline where the drop hits =',F8.3)
          WRITE (11,254)
          WRITE (11,255) SLCOLA
254     FORMAT ('Percentage slope of collector array, ')
255     FORMAT (' where positive is down off right wing =',F8.3)
         WRITE (11,260) SPRWID
260      FORMAT ('Width of spray collection strip (m) = ',F8.2)
             IF (ADJSWA.EQ.'o'.OR.ADJSWA.EQ.'O') GOTO 510
         WRITE (11,500)
500      FORMAT ('Adjacent passes are in the SAME directions.')
             GOTO 530
510     WRITE (11,520)
520      FORMAT ('Adjacent passes are in the OPPOSITE directions.')
530     WRITE (11,540)
540      FORMAT ('          ')
*
550    RETURN
        END
*      =================================================================
*
*
*
*
*         ******* INDVEL ****** 6000 ***********************************
*         Velocities : induced, air, droplet, relative.
*         ==============================================================
*
          SUBROUTINE INDVEL(ACELX,ACELY,ACELZ,CD,CONST1,CORRAD,CRCLAT,
     +        DIA,HEIGHT,KINVSC,PRPDIA,RE,SEP,SLPANG,SWRROT,TIME,VLPLAN,
     +        VELXVT,VELYVT,VELZVT,VLXWND,
     +        VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +        VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +        VLXREL,VLYREL,VLZREL,VELREL,
     +        VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
     +        VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
     +        VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +        VX,VY,VZ,X,Y,Z)
*
          REAL ACELX,ACELY,ACELZ,CD,CONST1,CONST5,CORRAD,CRCLAT,DIA,
```

```
     +          GRAV,HEIGHT,KINVSC,PI,PRPDIA,RE,SEP,SLPANG,SWRROT,TIME,
     +          R1,R2,R3,R4,R5,R6,R7,R8,VLPLAN,
     +          VELXVT,VELYVT,VELZVT,VLXWND,
     +          VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +          VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +          VLXREL,VLYREL,VLZREL,VELREL,
     +          VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
     +          VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
     +          VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +          VX,VY,VZ,
     +          X,X1,X2,X3,X4,X5,X6,X7,X8,
     +          Y,Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,
     +          Z,Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8
*
      GRAV = 9.80665
      PI = 3.141593
*
*     ------ For vortex 1 ------ RIGHT WING VORTEX
      X1 = X - VRTLX1
      Y1 = VLPLAN * TIME - Y
      Z1 = Z - VRTLZ1
      R1 = (X1*X1 + Z1*Z1)**0.5
      IF (X1.EQ.0.AND.Z1.GE.0) THEN
          ALPHA1=PI/2.0
          GOTO 110
       ENDIF
      IF (X1.EQ.0.AND.Z1.LT.0) THEN
          ALPHA1=-PI/2.0
          GOTO 110
       ENDIF
      ALPHA1 = ATAN(Z1/X1)
      IF (X1.LT.0) ALPHA1 = ALPHA1 + PI
110   IF (Y1.EQ.0) THEN
         BETA1=PI/2.0
       ELSE
         BETA1 = ATAN(R1/Y1)
       ENDIF
      IF (R1.EQ.0) THEN
          VELVT1=0
          GOTO 180
       ENDIF
      IF (R1.GT.CORRAD) GOTO 170
      VELVT1 = CRCLAT * (1.0 + COS(BETA1)) / (4.0* PI * CORRAD)
      VELVT1 = VELVT1 *(R1 / CORRAD)
      GOTO 180
170   VELVT1 = CRCLAT * (1.0 + COS(BETA1)) / (4.0 * PI *R1)
180   VLXVT1 = VELVT1 * SIN(ALPHA1 + PI)
      VLZVT1 = VELVT1 * COS(ALPHA1)
*
*     ------ For vortex 2 ------ LEFT WING VORTEX
      X2 = X - VRTLX2
      Y2 = Y1
      Z2 = Z - VRTLZ2
      R2 = (X2*X2 + Z2*Z2)**0.5
      IF (X2.EQ.0.AND.Z2.GE.0) THEN
          ALPHA2=PI/2.0
          GOTO 290
       ENDIF
      IF (X2.EQ.0.AND.Z2.LT.0) THEN
          ALPHA2=-PI/2.0
          GOTO 290
       ENDIF
      ALPHA2 = ATAN(Z2/X2)
       IF (X2.LT.0) ALPHA2 = ALPHA2 + PI
290   IF (Y2.EQ.0) THEN
```

```
            BETA2=PI/2.0
          ELSE
            BETA2  =  ATAN(R2/Y2)
        ENDIF
       IF  (R2.EQ.0)  THEN
            VELVT2=0
            GOTO  360
        ENDIF
       IF  (R2.GT.CORRAD)  GOTO  350
       VELVT2  =  CRCLAT  *  (1.0  +  COS(BETA2))  /  (4.0  *  PI  *  CORRAD)
       VELVT2  =  VELVT2  *(R2  /  CORRAD)
       GOTO  360
 350  VELVT2  =  CRCLAT  *  (1.0  +  COS(BETA2))  /  (4.0  *  PI  *  R2)
 360  VLXVT2  =  VELVT2  *  SIN(ALPHA2)
       VLZVT2  =  VELVT2  *  COS(ALPHA2  +  PI)
*
*      ------  For  vortex  3  ------  RIGHT  WING  VORTEX  IMAGE
       X3  =  X  -  VRTLX3
       Y3  =  Y1
       Z3  =  Z  -  VRTLZ3
       R3  =  SQRT(X3*X3  +  Z3*Z3)
       IF  (X3.EQ.0)  THEN
          ALPHA3=PI/2.0
          GOTO  460
        ENDIF
       ALPHA3  =  ATAN(Z3/X3)
       IF  (X3.LT.0)  ALPHA3  =  ALPHA3  +  PI
 460  IF  (Y3.EQ.0)  THEN
          BETA3=PI/2.0
         ELSE
          BETA3  =  ATAN(R3/Y3)
        ENDIF
       IF  (R3.EQ.0)  THEN
          VELVT3=0
          GOTO  530
        ENDIF
       IF  (R3.GT.CORRAD)  GOTO  520
       VELVT3  =  CRCLAT  *  (1.0  +  COS(BETA3))  /  (4.0  *  PI  *  CORRAD)
       VELVT3  =  VELVT3  *  (R3  /  CORRAD)
       GOTO  530
 520  VELVT3  =  CRCLAT  *  (1.0  +  COS(BETA3))  /  (4.0  *  PI  *  R3)
 530  VLXVT3  =  VELVT3  *  SIN(ALPHA3)
       VLZVT3  =  VELVT3  *  COS(ALPHA3  +  PI)
*
*      ------  For  vortex  4  ------  LEFT  WING  VORTEX  IMAGE
       X4  =  X  -  VRTLX4
       Y4  =  Y1
       Z4  =  Z  -  VRTLZ4
       R4  =  SQRT(X4*X4  +  Z4*Z4)
       IF  (X4.EQ.0)  THEN
          ALPHA4=PI/2.0
          GOTO  630
        ENDIF
       ALPHA4  =  ATAN(Z4/X4)
       IF  (X4.LT.0)  ALPHA4  =  ALPHA4  +  PI
 630  IF  (Y4.EQ.0)  THEN
          BETA4=PI/2.0
         ELSE
          BETA4  =  ATAN(R4/Y4)
        ENDIF
       IF  (R4.EQ.0)  THEN
          VELVT4=0
          GOTO  700
        ENDIF
       IF  (R4.GT.CORRAD)  GOTO  690
```

```
        VELVT4 = CRCLAT * (1.0 + COS(BETA4)) / (4.0 * PI * CORRAD)
        VELVT4 = VELVT4 * (R4 / CORRAD)
        GOTO 700
 690    VELVT4 = CRCLAT * (1.0 + COS(BETA4)) / (4.0 * PI * R4)
 700    VLXVT4 = VELVT4 * SIN(ALPHA4 + PI)
        VLZVT4 = VELVT4 * COS(ALPHA4)
*
*       ------ For vortex 5 ------ BOUND VORTEX
        X5 = X
        Y5 = -Y1
        Z5 = Z - VRTLZ5
        R5 = (Y5*Y5 + Z5*Z5)**0.5
*                 note ===> SEP=INISEP/2
        IF ((SEP-X5).EQ.0) THEN
            BETA5A=PI/2.0
          ELSE
            BETA5A = ATAN(R5/(SEP - X5))
         ENDIF
        IF ((SEP-X5).LT.0) BETA5A = BETA5A+PI
        IF ((SEP+X5).EQ.0) THEN
            BETA5B=PI/2.0
          ELSE
            BETA5B = ATAN(R5/(SEP+X5))
         ENDIF
        IF ((SEP+X5).LT.0) BETA5B = BETA5B + PI
        IF (Y5.EQ.0.AND.Z5.GE.0) THEN
             ALPHA5=PI/2.0
             GOTO 850
         ENDIF
        IF (Y5.EQ.0.AND.Z5.LT.0) THEN
             ALPHA5=-PI/2.0
             GOTO 850
         ENDIF
        ALPHA5 = ATAN(Z5/Y5) + PI
 850    IF (R5.EQ.0) THEN
            VELVT5=0
            GOTO 910
         ENDIF
        IF (R5.GT.CORRAD) GOTO 900
        VELVT5 = CRCLAT * (COS(BETA5A) + COS(BETA5B))/(4.0*PI*CORRAD)
        VELVT5 = VELVT5 *(R5 / CORRAD)
        GOTO 910
 900    VELVT5 = CRCLAT * (COS(BETA5A) + COS(BETA5B)) /(4.0* PI * R5)
 910    VLYVT5 = VELVT5 * SIN(ALPHA5 + PI)
        VLZVT5 = VELVT5 * COS(ALPHA5)
*
*       ------ For vortex 6 ------ BOUND VORTEX IMAGE
        X6 = X
        Y6 = -Y1
        Z6 = Z - VRTLZ6
        R6 = SQRT(Y6*Y6 + Z6*Z6)
        IF (Y6.EQ.0.AND.Z6.GE.0) THEN
            ALPHA6=PI/2.0
            GOTO 702
         ENDIF
        IF (Y6.EQ.0.AND.Z6.LT.0) THEN
            ALPHA6=-PI/2.0
            GOTO 702
         ENDIF
        ALPHA6 = ATAN(Z6/Y6) + PI
 702    IF ((SEP-X6).EQ.0) THEN
            BETA6A=PI/2.0
          ELSE
            BETA6A = ATAN(R6/(SEP - X6))
         ENDIF
```

```
        IF ((SEP-X6).LT.0) BETA6A = BETA6A + PI
        IF ((SEP+X6).EQ.0) THEN
            BETA6B=PI/2.0
          ELSE
            BETA6B = ATAN(R6/(SEP+X6))
         ENDIF
        IF ((SEP+X6).LT.0) BETA6B = BETA6B + PI
        IF (R6.EQ.0) THEN
            VELVT6=0
            GOTO 712
         ENDIF
        IF (R6.GT.CORRAD) GOTO 711
        VELVT6 = CRCLAT * (COS(BETA6A) + COS(BETA6B))/(4.0*PI*CORRAD)
        VELVT6 = VELVT6 *(R6 / CORRAD)
        GOTO 712
 711    VELVT6 = CRCLAT * (COS(BETA6A) + COS(BETA6B)) / (4.0*PI*R6)
 712    VLYVT6 = VELVT6 * SIN(ALPHA6)
        VLZVT6 = VELVT6 * COS(ALPHA6 + PI)
*
*       ------------- Vortex to simulate the propeller
        X7 = X - VRTLX7
        Z7 = Z - VRTLZ7
        R7 = (X7*X7 + Z7*Z7)**0.5
        IF (X7.EQ.0.AND.Z7.GE.0) THEN
            ALPHA7=PI/2.0
            GOTO 722
         ENDIF
        IF (X7.EQ.0.AND.Z7.LT.0) THEN
            ALPHA7=-PI/2.0
            GOTO 722
         ENDIF
        ALPHA7 = ATAN(Z7/X7)
         IF (X7.LT.0) ALPHA7 = ALPHA7 + PI
 722    IF (R7.EQ.0) THEN
            VELVT7=0
            GOTO 728
         ENDIF
        VELVT7 = SWRROT
*              positive SWiRl ROTation =clockwise in pilot's view
        IF (R7.GT.PRPDIA) GOTO 727
        VELVT7 = VELVT7 * R7 / PRPDIA
        GOTO 728
 727    VELVT7 = VELVT7*PRPDIA/R7
 728    VLXVT7 = VELVT7*SIN(ALPHA7)
        VLZVT7 = VELVT7*COS(ALPHA7+PI)
*
*       ------------- Vortex to simulate the propeller IMAGE
        X8 = X - VRTLX8
        Z8 = Z - VRTLZ8
        R8 = (X8*X8 + Z8*Z8)**0.5
        IF (X8.EQ.0) THEN
            ALPHA8 = PI / 2
            GOTO 730
         ENDIF
        ALPHA8 = ATAN(Z8/X8)
        IF (X8.LT.0) ALPHA8 = ALPHA8 + PI
 730    VELVT8 = -SWRROT*PRPDIA/R8
        VLXVT8 = VELVT8*SIN(ALPHA8)
        VLZVT8 = VELVT8*COS(ALPHA8+PI)
*
*       ======== Velocity Sums ==============
      VELXVT =VLXVT1+VLXVT2+VLXVT3+VLXVT4+VLXVT7+VLXVT8
     +            + VLXWND * COS(SLPANG)
      VELYVT =VLYVT5+VLYVT6
      VELZVT =VLZVT1+VLZVT2+VLZVT3+VLZVT4+VLZVT5+VLZVT6+VLZVT7+VLZVT8
```

```
     +           - VLXWND * SIN(FLPANG)
*          Note that the crosswind is assumed to be parallel to the
*                uniformly sloped ground plane
*
        VLXREL = VX - VELXVT
        VLYREL = VY - VELYVT
        VLZREL = VZ - VELZVT
        VELREL = SQRT(VLXREL*VLXREL + VLYREL*VLYREL + VLZREL*VLZREL)
*
        RE = VELREL * DIA / KINVSC
*         PRINT*, 'REYNOLDS NO. = ',RE
        CALL DRAGCF(RE,CD)
*                     Subroutine to get CD as a function of Re
*
*
*
*        ************************
*        Acceleration of droplet
*        =======================
        CONST5 = CONST1 * CD / DIA
        SVLXRL = 1
        SVLYRL = 1
        SVLZRL = 1
        ACELX = CONST5 * VLXREL * VLXREL * (-SIGN(SVLXRL,VLXREL))
        ACELY = CONST5 * VLYREL * VLYREL * (-SIGN(SVLYRL,VLYREL))
        ACELZ = (CONST5*VLZREL*VLZREL*(-SIGN(SVLZRL,VLZREL)))-GRAV
*
        RETURN
        END
*       ===============================================================
*
*
*       ******* NZVELS ******** 8000 **********************************
*       Subroutine for nozzle velocities in coordinate directions
*       ===============================================================
*
        SUBROUTINE NZVELS(DENDRP,ENZVEL,NOZTYP,NOZX,NOZY,NOZZ,
     +        NZPRES,NZVELX,NZVELY,NZVELZ)
*
        CHARACTER*1 NOZTYP
        REAL DENDRP,ENZVEL,NOZLEN,NOZVEL,NOZX,NOZY,NOZZ,NZANGX,
     +        NZANGY,NZANGZ,NZCOEF,NZPRES,NZVELX,NZVELY,NZVELZ
*
        NZCOEF = 0.8
*       NZCOEF = 0.8    is the recommended value by Goering 1972
        NOZVEL = (2.0 * NZCOEF * NZPRES / DENDRP)**0.5
        IF (NOZTYP.EQ.'r'.OR.NOZTYP.EQ.'R') NOZVEL = ENZVEL
*                    see Subroutine NOZZLE
        NOZLEN = (NOZX*NOZX + NOZY*NOZY + NOZZ*NOZZ)**0.5
*            nozzle angle direction cosines
        NZANGX = NOZX / NOZLEN
        NZANGY = NOZY / NOZLEN
        NZANGZ = NOZZ / NOZLEN
        NZVELX = NZANGX * NOZVEL
        NZVELY = NZANGY * NOZVEL
        NZVELZ = NZANGZ * NOZVEL
        RETURN
        END
*
*       ============================================================
*
*
*
*       ******* DRPLYF **** 9000 ************************************
*       Subroutine for initial calculation of DRoPLIFe of droplet
```

```
*         ============================================================
*
          SUBROUTINE  DRPLYF(CD,DENDRP,DENAIR,DIA,DIAMIC,DRPLIF,INIDIA,
     +           KINVSC,PINDIA,PREVRE,RE,WTBDEP)
*
          REAL  CD,DENAIR,DENDRP,DIA,DIAMIC,DRPLIF,EVPCF1,INIDIA,KINVSC,
     +           PINDIA,PREVRE,PRTTRD,RE,WTBDEP
*
          IF  (DIA.EQ.PINDIA)  THEN
                RE  =  PREVRE
                GOTO  50
           ENDIF
*                 see  540  and  9210  for  PINDIA  and  PREVRE
          DIA  =  INIDIA
*              Prandtl  number  =  0.72,  PRTTRD  =  Prandtl**1/3
          PRTTRD  =  0.9
         CALL  TRMVEL(CD,DIA,DENDRP,DENAIR,KINVSC,RE,PREVRE)
   50    EVPCF1  =  84.76  *  (1.0  +  0.3  *  PRTTRD  *  SQRT(RE))
          IF  (WTBDEP.EQ.0)  THEN
                DRPLIF  =  1.0E+15
                RETURN
           ENDIF
          DRPLIF  =  (DIAMIC  *  DIAMIC)  /  (EVPCF1  *  WTBDEP)
          RETURN
          END
*         ============================================================
*
*
*
*         *******  TRMVEL  *********  9100  **********************************
*         Subroutine  to  find  the  TERMinal  VELocity  for  droplet
*         ============================================================
*
          SUBROUTINE  TRMVEL(CD,DIA,DENDRP,DENAIR,KINVSC,RE,PREVRE)
*
          REAL  CD,DIA,DENDRP,DENAIR,GRAV,KINVSC,PREVRE,RE,RECOEF,
     +          VELT,VELTRM,VTRMCF
           GRAV  =  9.80665
*
          RECOEF  =  DIA  /  KINVSC
*                first  guess  of  CD
          CD  =  1.0
          VTRMCF  =  SQRT(4.0  *  DENDRP  *  GRAV  *  DIA  /  (3.0  *  DENAIR))
          VELTRM  =  VTRMCF  /  SQRT(CD)
  170     RE  =  RECOEF  *  VELTRM
*               Now  to  get  new  CD  from  subroutine  DRAGCF
          CALL  DRAGCF(RE,CD)
           VELT  =  VTRMCF  /  SQRT(CD)
  200    IF  (((VELT/VELTRM).GT.(0.999)).AND.((VELT/VELTRM).LT.(1.001)))
     +                                                                  THEN
               VELTRM  =  (VELT  +  VELTRM)/2.0
             ELSE
               VELTRM  =  VELT
               GOTO  170
           ENDIF
  210   RE  =  RECOEF  *  VELTRM
           PREVRE  =  RE
  220   RETURN
         END
*        ============================================================
*
*
*
*        *******  EVPDIA  ****  9300  **************************************
*        Subroutine  to  determine  evaporated  DIAmeter  of  droplet
```

```
*          ============================================================
*
           SUBROUTINE  EVPDIA(DRPLIF,INIDIA,DIA,TIME)
*
           REAL  DRPLIF,EVPCF2,INIDIA,DIA,TIME
*
           EVPCF2  =  (1.0  -  TIME/DRPLIF)**0.5
           DIA  =  EVPCF2  *  INIDIA
           RETURN
           END
*
*          ==========================================================
*
*
*
*
*
*            ****** RUNKUT *** 10000 *************************************
*            Runge-Kutta  Fourth  Order  Algorithm
*            ============================================================
*
*        Output  Files:  (UNIT=10)   -DROPDATA     individual  drop  trajectories
*               (see  Main  Program)
*
         SUBROUTINE  RUNKUT(ACELX,ACELY,ACELZ,CD,CFXWND,CHK,CONST1,
     +      CORRAD,CRCLAT,DELT,DIA,DRPLIF,DRPNUM,DRPOUT,FHIGHT,
     +      FRPRNT,HEIGHT,HTCOLA,INIDIA,KINVSC,LCCNTR,LOCX,LOCY,LOCZ,
     +      PRPDIA,RE,RELAST,
     +      SEP,SLCOLA,SLPANG,SRFHIT,SWACNT,SWRROT,TIMCHK,TIME,
     +      VELREL,VELX,VELY,VELZ,VELXVT,VELYVT,VELZVT,VLXWND,VLPLAN,
     +      VLXREL,VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLXVT7,VLXVT8,
     +      VLYREL,VLYVT5,VLYVT6,VLZREL,
     +      VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +      VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +      VRTLY1,VRTLY2,VRTLY3,VRTLY4,
     +      VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8,
     +      VX,VY,VZ,X,Y,Z)
*
           CHARACTER*1  DRPOUT
           INTEGER  CHK,LCCNTR,SWACNT
           REAL  ACELX,ACELY,ACELZ,CD,CFXWND,CONST1,CORRAD,CRCLAT,
     +           DELT,DIA,DRPLIF,DRPNUM,
     +           FHIGHT,FRPRNT,HEIGHT,HTCOLA,INIDIA,INTCOF,KINVSC,
     +           K11X,K11Y,K11Z,K12X,K12Y,K12Z,K21X,K21Y,K21Z,
     +           K22X,K22Y,K22Z,K31X,K31Y,K31Z,K32X,K32Y,K32Z,
     +           K41X,K41Y,K41Z,K42X,K42Y,K42Z,K1X,K1Y,K1Z,K2X,K2Y,K2Z,
     +           LOCX,LOCY,LOCZ,LSLOCX,LSLOCY,LSLOCZ,LSTIME,
     +           PI,PRPDIA,RE,RELAST,
     +           SEP,SLCOLA,SLPANG,SRFHIT,SWRROT,TIMCHK,TIME,
     +           VELREL,VELX,VELY,VELZ,VELXVT,VELYVT,VELZVT,VLXWND,VLPLAN,
     +           VLXREL,VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLXVT7,VLXVT8,
     +           VLYREL,VLYVT5,VLYVT6,VLZREL,
     +           VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +           VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +           VRTLY1,VRTLY2,VRTLY3,VRTLY4,
     +           VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8,
     +           VX,VY,VZ,X,Y,Z
*
            PI  =  3.141593
*
*          -----  step  1
            TIME  =  TIME
            X  =  LOCX
            Y  =  LOCY
            Z  =  LOCZ
```

```
      VX  =  VELX
      VY  =  VELY
      VZ  =  VELZ
*
      CALL  INDVEL(ACELX,ACELY,ACELZ,CD,CONST1,CORRAD,CRCLAT,
     +       DIA,HEIGHT,KINVSC,PRPDIA,RE,SEP,SLPANG,SWRROT,TIME,VLPLAN,
     +       VELXVT,VELYVT,VELZVT,VLXWND,
     +       VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +       VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +       VLXREL,VLYREL,VLZREL,VELREL,
     +       VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
     +       VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
     +       VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +       VX,VY,VZ,X,Y,Z)
*                 vortex, air and relative velocities and accelerations
      IF  (RE.GT.(1.5*RELAST))  THEN
          DELT  =  DELT/2.0
          CHK  =  CHK-1
      ENDIF
       RELAST  =  RE
      K11X  =  DELT  *  VX
      K11Y  =  DELT  *  VY
      K11Z  =  DELT  *  VZ
      K12X  =  DELT  *  ACELX
      K12Y  =  DELT  *  ACELY
      K12Z  =  DELT  *  ACELZ
*
*        -----  step  2
      TIME  =  TIME  +  DELT/2
      X  =  LOCX  +  K11X/2.0
      Y  =  LOCY  +  K11Y/2.0
      Z  =  LOCZ  +  K11Z/2.0
      VX  =  VELX  +  K12X/2.0
      VY  =  VELY  +  K12Y/2.0
      VZ  =  VELZ  +  K12Z/2.0
*
      CALL  INDVEL(ACELX,ACELY,ACELZ,CD,CONST1,CORRAD,CRCLAT,
     +       DIA,HEIGHT,KINVSC,PRPDIA,RE,SEP,SLPANG,SWRROT,TIME,VLPLAN,
     +       VELXVT,VELYVT,VELZVT,VLXWND,
     +       VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +       VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +       VLXREL,VLYREL,VLZREL,VELREL,
     +       VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
     +       VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
     +       VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +       VX,VY,VZ,X,Y,Z)
*                 vortex, air and relative velocities and accelerations
      K21X  =  DELT  *  VX
      K21Y  =  DELT  *  VY
      K21Z  =  DELT  *  VZ
      K22X  =  DELT  *  ACELX
      K22Y  =  DELT  *  ACELY
      K22Z  =  DELT  *  ACELZ
*
*        -----  step  3
      TIME  =  TIME
      X  =  LOCX  +  K21X/2.0
      Y  =  LOCY  +  K21Y/2.0
      Z  =  LOCZ  +  K21Z/2.0
      VX  =  VELX  +  K22X/2.0
      VY  =  VELY  +  K22Y/2.0
      VZ  =  VELZ  +  K22Z/2.0
*
      CALL  INDVEL(ACELX,ACELY,ACELZ,CD,CONST1,CORRAD,CRCLAT,
     +       DIA,HEIGHT,KINVSC,PRPDIA,RE,SEP,SLPANG,SWRROT,TIME,VLPLAN,
```

```
     +       VELXVT,VELYVT,VELZVT,VLXWND,
     +       VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +       VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +       VLXREL,VLYREL,VLZREL,VELREL,
     +       VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
     +       VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
     +       VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +       VX,VY,VZ,X,Y,Z)
*                  vortex, air and relative velocities and accelerations
        K31X  =  DELT * VX
        K31Y  =  DELT * VY
        K31Z  =  DELT * VZ
        K32X  =  DELT * ACELX
        K32Y  =  DELT * ACELY
        K32Z  =  DELT * ACELZ
*
*            ----- step 4
        TIME = TIME + DELT/2.0
        X = LOCX + K31X
        Y = LOCY + K31Y
        Z = LOCZ + K31Z
        VX = VELX + K32X
        VY = VELY + K32Y
        VZ = VELZ + K32Z
*
      CALL INDVEL(ACELX,ACELY,ACELZ,CD,CONST1,CORRAD,CRCLAT,
     +       DIA,HEIGHT,KINVSC,PRPDIA,RE,SEP,SLPANG,SWRROT,TIME,VLPLAN,
     +       VELXVT,VELYVT,VELZVT,VLXWND,
     +       VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +       VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +       VLXREL,VLYREL,VLZREL,VELREL,
     +       VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
     +       VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
     +       VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +       VX,VY,VZ,X,Y,Z)
*                  vortex, air and relative velocities and accelerations
        K41X  =  DELT * VX
        K41Y  =  DELT * VY
        K41Z  =  DELT * VZ
        K42X  =  DELT * ACELX
        K42Y  =  DELT * ACELY
        K42Z  =  DELT * ACELZ
*
*            ----- step 5
        K1X  =  (K11X + 2.0*K21X + 2.0*K31X + K41X)/6.0
        K1Y  =  (K11Y + 2.0*K21Y + 2.0*K31Y + K41Y)/6.0
        K1Z  =  (K11Z + 2.0*K21Z + 2.0*K31Z + K41Z)/6.0
        K2X  =  (K12X + 2.0*K22X + 2.0*K32X + K42X)/6.0
        K2Y  =  (K12Y + 2.0*K22Y + 2.0*K32Y + K42Y)/6.0
        K2Z  =  (K12Z + 2.0*K22Z + 2.0*K32Z + K42Z)/6.0
*
*            ----- step 6
        TIME  =  TIME
        LOCX  =  LOCX + K1X
        LOCY  =  LOCY + K1Y
        LOCZ  =  LOCZ + K1Z
        VELX  =  VELX + K2X
        VELY  =  VELY + K2Y
        VELZ  =  VELZ + K2Z
*
        FHIGHT = HTCOLA - (LOCX * SLCOLA/100)
       IF (LOCZ.GT.FHIGHT) GOTO 990
         INTCOF = (FHIGHT-LSLOCZ)/(LOCZ-LSLOCZ)
         LOCX = (LOCX-LSLOCX) * INTCOF  +   LSLOCX
         LOCY = (LOCY-LSLOCY) * INTCOF  +   LSLOCY
```

```
            LOCZ  =  FHIGHT
            TIME  =  (TIME-LSTIME) * INTCOF   +    LSTIME
*
*         ------- Print intermediate results to screen
*         PRINT*, 'TIME,RE,SWACNT,LCCNTR,DRPNUM',TIME,RE,SWACNT,LCCNTR,
*     +                DRPNUM
*         PRINT*, 'DROPLET LOCATIONS   X,Y,Z   ',LOCX,LOCY,LOCZ
*         PRINT*, '#1 Vortex locations X,Y,Z ',VRTLX1,VRTLY1,VRTLZ1
*         PRINT*, '#3 Vortex locations X,Y,Z ',VRTLX3,VRTLY3,VRTLZ3
*         PRINT*, '#2 Vortex locations X,Y,Z ',VRTLX2,VRTLY2,VRTLZ2
*         PRINT*, '#4 Vortex locations X,Y,Z ',VRTLX4,VRTLY4,VRTLZ4
*         PRINT*, 'TIME,DELT,CD,RE   ',TIME,DELT,CD,RE
*         PRINT*, 'VELXVT     ',VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLXVT7,VLXVT8
*         PRINT*, 'VELYVT     ',VLYVT5,VLYVT6
*         PRINT*, 'VELZVT     ',VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,
*     +            VLZVT7,VLZVT8
*         PRINT*, 'TOTAL VELVT    X,Y,Z   ',VELXVT,VELYVT,VELZVT
*         PRINT*, 'DROP VEL   : VELX,VELY,VELZ ',VELX,VELY,VELZ
*         PRINT*, 'VELREL S X,Y,Z,TOT   ',VLXREL,VLYREL,VLZREL,VELREL
*         PRINT*, 'ACEL X,Y,Z   ',ACELX,ACELY,ACELZ
*         PRINT*, '--------------------------------'
  990     IF (LOCZ.LE.FHIGHT) GOTO 185
*
        CALL VRTCOR(CRCLAT,DELT,PRPDIA,SLPANG,SWRROT,VLXWND,
     +    VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +    VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8)
*                       Subroutine to get new VoRTex core LOCations
        CALL EVPDIA(DRPLIF,INIDIA,DIA,TIME)
*                    Subroutine to get new evaporated drop DIAmeter
*
        VLXWND = CFXWND * LOG(LOCZ/SRFHIT)
*                                                       see 5140
          IF (TIME.LT.TIMCHK) RETURN
            TIMCHK = TIMCHK + FRPRNT
           LSLOCX=LOCX
           LSLOCY=LOCY
           LSLOCZ=LOCZ
           LSTIME=TIME
 185    IF (DRPOUT.EQ.'n'.OR.DRPOUT.EQ.'N') GOTO 400
*                                                        See 4044
          WRITE (10,390) LOCX,LOCZ,LOCY,TIME,VRTLX1,VRTLZ1
 390      FORMAT (6(F8.4,2X))
*
 400      RETURN
          END
*
*         ========================================================
*
*
*         ******* VRTCOR ********* 11400 ********************************
*         Subroutine to calculate VoRTex CORE  LOCations and velocities
*         ==========================================================
*
      SUBROUTINE VRTCOR(CRCLAT,DELT,PRPDIA,SLPANG,SWRROT,VLXWND,
     +    VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +    VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8)
*
        INTEGER N
        REAL VX(8),VZ(8),VR1(8),VR2(8),VR3(8),VR4(8),VR7(8),VR8(8),
     +  VRVLX(8),VRVLZ(8),
     +  VVLX1(8),VVLX2(8),VVLX3(8),VVLX4(8),VVLX7(8),VVLX8(8),
     +  VVLZ1(8),VVLZ2(8),VVLZ3(8),VVLZ4(8),VVLZ7(8),VVLZ8(8)
        REAL CRCLAT,CRCPI,DELT,PI,SLPANG,VLXWND,
     +  VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +  VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8
```

```
*
        PI = 3.141593
         VX(1)  =  VRTLX1
         VX(2)  =  VRTLX2
         VX(3)  =  VRTLX3
         VX(4)  =  VRTLX4
         VX(7)  =  VRTLX7
         VX(8)  =  VRTLX8
         VZ(1)  =  VRTLZ1
         VZ(2)  =  VRTLZ2
         VZ(3)  =  VRTLZ3
         VZ(4)  =  VRTLZ4
         VZ(7)  =  VRTLZ7
         VZ(8)  =  VRTLZ8
*
*
*            velocities for potential flow and sloping ground
*                induced velocity at core #N center location
       DO 10 N=1,8
          IF (N.EQ.5.OR.N.EQ.6) GOTO 10
          CRCPI = CRCLAT / (2.0*PI)
*              distances between vortex cores
         VR1(N)= (VX(N) - VX(1))**2 + (VZ(N) - VZ(1))**2
         VR2(N)= (VX(N) - VX(2))**2 + (VZ(N) - VZ(2))**2
         VR3(N)= (VX(N) - VX(3))**2 + (VZ(N) - VZ(3))**2
         VR4(N)= (VX(N) - VX(4))**2 + (VZ(N) - VZ(4))**2
         VR7(N)= (VX(N) - VX(7))**2 + (VZ(N) - VZ(7))**2
         VR8(N)= (VX(N) - VX(8))**2 + (VZ(N) - VZ(8))**2
*            X and Z induced velocities at core locations
        IF (N.EQ.1) GOTO 2
         VVLX1(N) = -CRCPI * (VZ(N) - VZ(1)) / VR1(N)
         VVLZ1(N) = CRCPI * (VX(N) - VX(1)) / VR1(N)
  2     IF (N.EQ.2) GOTO 3
         VVLX2(N) = CRCPI * (VZ(N) - VZ(2)) / VR2(N)
         VVLZ2(N) = -CRCPI * (VX(N) - VX(2)) / VR2(N)
  3     IF (N.EQ.3) GOTO 4
         VVLX3(N) = CRCPI * (VZ(N) - VZ(3)) / VR3(N)
         VVLZ3(N) = -CRCPI * (VX(N) - VX(3)) / VR3(N)
  4     IF (N.EQ.4) GOTO 7
         VVLX4(N) = -CRCPI * (VZ(N) - VZ(4)) / VR4(N)
         VVLZ4(N) = CRCPI * (VX(N) - VX(4)) / VR4(N)
  7     IF (N.EQ.7) GOTO 8
         VVLX7(N) = (SWRROT*PRPDIA) * (VZ(N) - VZ(7)) / VR7(N)
         VVLZ7(N) = -(SWRROT*PRPDIA) * (VX(N) - VX(7)) / VR7(N)
  8     IF (N.EQ.8) GOTO 9
         VVLX8(N) = -(SWRROT*PRPDIA) * (VZ(N) - VZ(8)) / VR8(N)
         VVLZ8(N) = (SWRROT*PRPDIA) * (VX(N) - VX(8)) / VR8(N)
*            X and Z induced velocity sums at core location N
  9      VRVLX(N) = VVLX1(N) + VVLX2(N) + VVLX3(N) + VVLX4(N) +
     +         VVLX7(N) + VVLX8(N) + VLXWND * COS(SLPANG)
         VRVLZ(N) = VVLZ1(N) + VVLZ2(N) + VVLZ3(N) + VVLZ4(N) +
     +         VVLZ7(N) + VVLZ8(N) - VLXWND * SIN(SLPANG)
*       Note that the crosswind is assumed to be parallel to the
*              uniformly sloped ground plane
 10    CONTINUE
*            vortex core locations
      DO 20 N=1,8
          IF (N.EQ.5.OR.N.EQ.6) GOTO 20
         VX(N) = VX(N) + VRVLX(N) * DELT
         VZ(N) = VZ(N) + VRVLZ(N) * DELT
 20    CONTINUE
        VRTLX1 = VX(1)
        VRTLX2 = VX(2)
        VRTLX3 = VX(3)
        VRTLX4 = VX(4)
```

```
      VRTLX7  =  VX(7)
      VRTLX8  =  VX(8)
      VRTLZ1  =  VZ(1)
      VRTLZ2  =  VZ(2)
      VRTLZ3  =  VZ(3)
      VRTLZ4  =  VZ(4)
      VRTLZ7  =  VZ(7)
      VRTLZ8  =  VZ(8)
*
      RETURN
      END
*    ============================================================
*
*
*      ****** DRAGCF ***** 11700 ***********************************
*      Subroutine to define CD in terms of Reynolds number (RE)
*      ==========================================================
*
      SUBROUTINE DRAGCF(RE,CD)
*
      REAL RE,CD,CDBP,CDLB,CDSTOK
*
      CDSTOK = 24.0 / RE
*               drag coefficient of droplet according to Stoke's law
      IF (RE.LT.(0.01)) THEN
          CD = CDSTOK
          RETURN
        ENDIF
*   (Beard & Pruppacher (1969) p1069-1070;Mason(1971);Pruppacher(1970))
      IF (RE.LT.2) THEN
           CD = CDSTOK * (1.0 +0.102 * RE **0.955)
           RETURN
        ENDIF
      IF (RE.LT.21) THEN
           CD = CDSTOK * (1.0 +0.115 * RE **0.802)
           RETURN
        ENDIF
      IF (RE.LT.200) THEN
           CD = CDSTOK * (1.0 +0.189 * RE **0.632)
           RETURN
        ENDIF
*          Now is a linear interpolation between the theory of
*         Beard and Pruppacher, and that of Lamguir and Blogdett
      IF (RE.LT.400) THEN
          CDBP = CDSTOK * (1.0 +0.189 * RE **0.632)
          CDLB = CDSTOK * (1.0 +0.197 * RE **0.63+0.00026*RE **1.38)
          CD = CDBP + ((RE-200.0)/200.0) * (CDLB - CDBP)
          RETURN
        ENDIF
      IF (RE.GT.50000) THEN
          CD =0.5
          RETURN
        ENDIF
*                now for (400.LT.RE and RE.LT.50000)
      CD = CDSTOK * (1.0 +0.197 * RE **0.63 +0.00026 * RE ** 1.38)
*                               (from Langmuir and Blodgett, 1946)
      RETURN
      END
*    ==============================================================
*
*
*
*
*   ********* PRNFIL ******* 12000    ******************************
*   Subroutine to print intermediate results to file
```

```
*         ============================================================
*
*         Output file: (UNIT=11) -SPRAYINFO    for information for spraying
*            (see Main Program)       conditions, etc and final drop locations
*
          SUBROUTINE PRNFIL(ACELX,ACELY,ACELZ,CD,DELT,
     +         PRNT,RE,TIME,VELXVT,VELYVT,VELZVT,
     +         VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +         VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +         VLXREL,VLYREL,VLZREL,VELREL,
     +         VRTLX1,VRTLZ1,VRTLX2,VRTLZ2,
     +         VRTLX3,VRTLZ3,VRTLX4,VRTLZ4,
     +         VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +         VX,VY,VZ,X,Y,Z)
*
          REAL ACELX,ACELY,ACELZ,CD,DELT,
     +         PRNT,RE,TIME,VELXVT,VELYVT,VELZVT,
     +         VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +         VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +         VLXREL,VLYREL,VLZREL,VELREL,
     +         VRTLX1,VRTLZ1,VRTLX2,VRTLZ2,
     +         VRTLX3,VRTLZ3,VRTLX4,VRTLZ4,
     +         VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +         VX,VY,VZ,X,Y,Z
*
          WRITE (11,10) X,Y,Z
 10       FORMAT ('DROPLET LOCATIONS   X,Y,Z =',3(F8.3))
          WRITE (11,20) VRTLX1,VRTLZ1
 20       FORMAT ('#1 Vortex locations X,Z ',2(F8.3))
          WRITE (11,30) VRTLX3,VRTLZ3
 30       FORMAT ('#3 Vortex locations X,Z ',2(F8.3))
          WRITE (11,40) VRTLX2,VRTLZ2
 40       FORMAT ('#2 Vortex locations X,Z ',2(F8.3))
          WRITE (11,50) VRTLX4,VRTLZ4
 50       FORMAT ('#4 Vortex locations X,Z ',2(F8.3))
          WRITE (11,60) VRTLX7,VRTLZ7
 60       FORMAT ('#7 Vortex locations X,Z ',2(F8.3))
          WRITE (11,65) VRTLX8,VRTLZ8
 65       FORMAT ('#8 Vortex locations X,Z ',2(F8.3))
          WRITE (11,70) TIME,DELT,CD,RE
 70       FORMAT ('TIME,DELT,CD,RE   ',4(F12.6))
          WRITE (11,80) VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLXVT7,VLXVT8
 80       FORMAT ('VELXVT    ',6(F8.3))
          WRITE (11,90) VLYVT5,VLYVT6
 90       FORMAT ('VELYVT    ',2(F8.3))
          WRITE (11,100) VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,
     +           VLZVT8
 100      FORMAT ('VELZVT    ',8(F8.3))
          WRITE (11,110) VELXVT,VELYVT,VELZVT
 110      FORMAT ('TOTAL VELVT    X,Y,Z   ',3(F8.3))
          WRITE (11,120) VX,VY,VZ
 120      FORMAT ('DROP VEL    VX,VY,VZ   ',3(F8.3))
          WRITE (11,130) VLXREL,VLYREL,VLZREL,VELREL
 130      FORMAT ('VELREL    X,Y,Z,TOT   ',4(F8.3))
          WRITE (11,140) ACELX,ACELY,ACELZ
 140      FORMAT ('ACEL X,Y,Z   ',3(F10.2))
          WRITE (11,150)
 150      FORMAT ('--------------------------------')
*
           PRNT = PRNT + 0.25
           RETURN
           END
*          ====================================================
*
*
```

```
*          ******* PNTOUT **** 12500 *****************************
*          Subroutine to PRINT Results and Output for the Droplet
*          ====================================================
*
*       Output file: (UNIT=11) -SPRAYINFO    for information for sprayirg
*          (see Main Program)       conditions, etc and final drop locations
*
           SUBROUTINE PNTOUT(CD,DELT,DIA,DRPNUM,DRPRUN,
     +                 INILOX,INILOY,INILOZ,INIVLX,INIVLY,INIVLZ,
     +                 LCCNTR,LOCX,LOCY,LOCZ,RE,SWACNT,TIME)
*
           INTEGER DRPRUN,LCCNTR,SWACNT
           REAL CD,DELT,DIA,DRPNUM,LOCX,LOCY,LOCZ,RE,TIME,
     +          INILOX,INILOY,INILOZ,INIVLX,INIVLY,INIVLZ
*
*       PRINT*, 'Droplet simulation ended at time (sec) = ',TIME
*       PRINT*, 'The final DELT =   ',DELT,' Final Reynolds no = ',RE
*       PRINT*, '            '
*
           IF (SWACNT.EQ.2.OR.SWACNT.EQ.4) LOCX = -LOCX
*                                                     see 15080
         IF (DRPRUN.GT.0) GOTO 35
          WRITE (11,10)
   10    FORMAT ('  SWACNT LCCNTR DRPNUM')
          WRITE (11,20)
   20    FORMAT (' INIVLX      INIVLY      INIVLZ      INILOX      INILOY      INI
     +LOZ')
          WRITE (11,30)
   30    FORMAT ('  TIME          DELT          CD            RE          LOCX
LOCY
     +     LOCZ           DIA')
*
   35    WRITE (11,40) SWACNT,LCCNTR,DRPNUM
   40    FORMAT (2(I4,3X),F8.2)
   50    WRITE (11,60) INIVLX,INIVLY,INIVLZ,INILOX,INILOY,INILOZ
   60    FORMAT (6(F9.4,1X))
          WRITE (11,70) TIME,DELT,CD,RE,LOCX,LOCY,LOCZ,DIA
   70    FORMAT (F9.6,1X,F9.7,F8.2,1X,F9.4,2X,3(F7.3,2X),1X,E11.2)
*
          IF (SWACNT.EQ.2.OR.SWACNT.EQ.4) LOCX = -LOCX
*                                                     see 15080
          RETURN
          END
*
*         ==========================================================
*
*
*
*        ********  CHDELT  *******  13000  *****************************
*         Subroutine to change size of DELT  to reduce calculation time
*        ============================================================
*
          SUBROUTINE CHDELT(CHK,DELT,DIA,RE)
*
          INTEGER CHK
          REAL DIA,RE,DELT
*
          IF (DIA.GT.(0.0003)) GOTO 30
          IF (DIA.LT.(0.0001)) GOTO 10
           IF (RE.LT.1000.AND.CHK.LT.1) THEN
                DELT=DELT*2.0
                CHK=CHK+1
                GOTO 100
             ENDIF
           IF (RE.LT.500.AND.CHK.LT.2) THEN
```

```
                DELT=DELT*2.0
                CHK=CHK+1
                GOTO  100
           ENDIF
          IF  (RE.LT.300.AND.CHK.LT.3)  THEN
                DELT=DELT*2.0
                CHK=CHK+1
                GOTO  100
           ENDIF
          IF  (RE.LT.200.AND.CHK.LT.4)  THEN
                DELT=DELT*2.0
                CHK=CHK+1
                GOTO  100
           ENDIF
          IF  (RE.LT.100.AND.CHK.LT.5)  THEN
                DELT=DELT*2.0
                CHK=CHK+1
                GOTO  100
           ENDIF
          IF  (RE.LT.50.AND.CHK.LT.6)  THEN
                DELT=DELT*2.0
                CHK=CHK+1
                GOTO  100
           ENDIF
*         IF  (RE.LT.30.AND.CHK.LT.7)  THEN
*               DELT=DELT*2.0
*               CHK=CHK+1
*               GOTO  100
*          ENDIF
        GOTO  100
*
 10     IF  (DIA.LT.(0.00003))  GOTO  20
          IF  (RE.LT.200.AND.CHK.LT.1)  THEN
                DELT=DELT*2.0
                CHK=CHK+1
                GOTO  100
           ENDIF
          IF  (RE.LT.100.AND.CHK.LT.2)  THEN
                DELT=DELT*2.0
                CHK=CHK+1
                GOTO  100
           ENDIF
          IF  (RE.LT.30.AND.CHK.LT.3)  THEN
                DELT=DELT*2.0
                CHK=CHK+1
                GOTO  100
           ENDIF
          IF  (RE.LT.10.AND.CHK.LT.4)  THEN
                DELT=DELT*2.0
                CHK=CHK+1
                GOTO  100
           ENDIF
          IF  (RE.LT.4.AND.CHK.LT.5)  THEN
                DELT=DELT*2.0
                CHK=CHK+1
                GOTO  100
           ENDIF
          IF  (RE.LT.1.AND.CHK.LT.6)  THEN
                DELT=DELT*2.0
                CHK=CHK+1
                GOTO  100
           ENDIF
          IF  (RE.LT.(0.5).AND.CHK.LT.7)  THEN
                DELT=DELT*2.0
                CHK=CHK+1
```

```
              GOTO  100
           ENDIF
        GOTO  100
*
   20   IF  (DIA.LT.(0.00001))  GOTO  40
           IF  (RE.LT.5.AND.CHK.LT.1)  THEN
              DELT=DELT*2.0
              CHK=CHK+1
              GOTO  100
           ENDIF
           IF  (RE.LT.1.AND.CHK.LT.2)  THEN
              DELT=DELT*2.0
              CHK=CHK+1
              GOTO  100
           ENDIF
           IF  (RE.LT.(0.5).AND.CHK.LT.3)  THEN
              DELT=DELT*2.0
              CHK=CHK+1
              GOTO  100
           ENDIF
           IF  (RE.LT.(0.1).AND.CHK.LT.4)  THEN
              DELT=DELT*2.0
              CHK=CHK+1
              GOTO  100
           ENDIF
           IF  (RE.LT.(0.05).AND.CHK.LT.5)  THEN
              DELT=DELT*2.0
              CHK=CHK+1
              GOTO  100
           ENDIF
           IF  (RE.LT.(0.025).AND.CHK.LT.6)  THEN
              DELT=DELT*2.0
              CHK=CHK+1
              GOTO  100
           ENDIF
           IF  (RE.LT.(0.02).AND.CHK.LT.7)  THEN
              DELT=DELT*2.0
              CHK=CHK+1
              GOTO  100
           ENDIF
           IF  (RE.LT.(0.019).AND.CHK.LT.8)  THEN
              DELT=DELT*5.0
              CHK=CHK+1
              GOTO  100
           ENDIF
        GOTO  100
*
   30   IF  (RE.LT.5000.AND.CHK.LT.1)  THEN
              DELT=DELT*2.0
              CHK=CHK+1
              GOTO  100
           ENDIF
           IF  (RE.LT.4000.AND.CHK.LT.2)  THEN
              DELT=DELT*2.0
              CHK=CHK+1
              GOTO  100
           ENDIF
           IF  (RE.LT.3000.AND.CHK.LT.3)  THEN
              DELT=DELT*2.0
              CHK=CHK+1
              GOTO  100
           ENDIF
           IF  (RE.LT.2000.AND.CHK.LT.4)  THEN
              DELT=DELT*2.0
              CHK=CHK+1
```

```
                GOTO  100
             ENDIF
            IF  (RE.LT.1000.AND.CHK.LT.5)  THEN
                DELT=DELT*2.0
                CHK=CHK+1
                GOTO  100
             ENDIF
            IF  (RE.LT.500.AND.CHK.LT.6)  THEN
                DELT=DELT*2.0
                CHK=CHK+1
                GOTO  100
             ENDIF
          GOTO  100
*
  40      PRINT*,  '******   Diameter  is  less  than  10  microns!   ******'
*
 100      RETURN
           END
*
*          ==========================================================
*
*
*
*
*       *********  NUMVOL  ********  15000  *************************************
*       Subroutine  to  store  arrays  for  NUMBer  and  VOLume  lateral  distribution
*       ===================================================================
*       Note  that  this  subroutine  is  for  up  to  75  m  spray  coverage  width
*                                   with  1/2  m   divisions  for  spray  collection
*
          SUBROUTINE  NUMVOL(ADJNUM,ADJVOL,DIA,DIVS,DIV1S,DRPRUN,INIDIA,
     +          INIVOL,LOCX,LOCXX,LOCZ,NUMB,SPRWID,SWACNT,TIME,UEPVTL,VOL)
*
          INTEGER  DIVS,DIV1S,DRPRUN,I,SWACNT
          REAL  ADJLCX,ADJNUM(152),ADJVOL(152),DIA,INIDIA,INIVOL,LOCX,
     +     LOCXX(152),LOCZ,NUMB(152),PI,SPRWID,TIME,UEPVTL,VOL(152),VOLUM
*
           PI  =  3.141593
*
           DIVS  =  SPRWID  *  2  +  2
*                      #  of  division  for  catching  spray  pattern
           DIV1S  =  DIVS  -  1
           VOLUM  =  PI  *  DIA  *  DIA  *  DIA  /  6.0
           UEPVTL  =  UEPVTL  +  VOLUM
           INIVOL  =  INIVOL  +  INIDIA*INIDIA*INIDIA  *  PI/6
*
          IF  (SWACNT.EQ.2)  THEN
              LOCX  =  -LOCX
              GOTO  100
           ENDIF
          IF  (SWACNT.EQ.3)  THEN
              ADJLCX  =  LOCX
              GOTO  200
           ENDIF
          IF  (SWACNT.EQ.4)  THEN
              ADJLCX  =  -LOCX
              GOTO  200
           ENDIF
*
 100     IF  (DRPRUN.GT.1)  GOTO  120
         DO  110  I=1,DIVS
              LOCXX(I)=-(SPRWID/2.0+0.75)+I/2.0
              NUMB(I)  =  0.0
              VOL(I)  =  0.0
              ADJNUM(I)  =  0.0
```

```
              ADJVOL(I) = 0.0
 110   CONTINUE
 120      IF ((TIME.GT.20).AND.(LOCZ.GT.0)) RETURN
              IF (DIA.LT.(0.5*INIDIA)) RETURN
          IF (LOCX.LE.(-SPRWID/2.0)) THEN
              VOL(1)=VOL(1)+VOLUM
              NUMB(1) = NUMB(1) + 1
              RETURN
          ENDIF
        DO 150 I=2,DIV1S
            IF ((LOCX.LE.(-(SPRWID/2.0+0.5)+FLOAT(I)/2.0)).AND.
     +                  (LOCX.GT.(-(SPRWID/2.0+1.0)+FLOAT(I)/2.0))) THEN
                VOL(I) = VOL(I)+ VOLUM
                NUMB(I) = NUMB(I) + 1
                RETURN
            ENDIF
 150    CONTINUE
        IF (LOCX.GT.(SPRWID/2.0)) THEN
            VOL(DIVS) = VOL(DIVS) + VOLUM
            NUMB(DIVS) = NUMB(DIVS) + 1
        ENDIF
        GOTO 400
*
 200   IF (ADJLCX.LE.(-SPRWID/2.0)) THEN
            ADJVOL(1) = ADJVOL(1) + VOLUM
            ADJNUM(1) = ADJNUM(1) + 1
            RETURN
        ENDIF
       DO 230 I=2,DIV1S
          IF ((ADJLCX.LE.(-(SPRWID/2.0+0.5)+FLOAT(I)/2.0)).AND.
     +            (ADJLCX.GT.(-(SPRWID/2.0+1.0) + FLOAT(I)/2.0))) THEN
              ADJVOL(I) = ADJVOL(I) + VOLUM
              ADJNUM(I) = ADJNUM(I) + 1
              RETURN
          ENDIF
 230   CONTINUE
       IF (ADJLCX.GT.(SPRWID/2.0)) THEN
            ADJVOL(DIVS) = VOL(DIVS) + VOLUM
            ADJNUM(DIVS) = ADJNUM(DIVS) + 1.0
       ENDIF
*
 400   RETURN
          END
*       ===============================================================
*
*
*
*         ***** VOLDST ********** 16000 *********************************
*         Subroutine to print VOLume DISTribution to datafile
*         ===============================================================
*           to use to plot the volume deposited vs. location
*              or    to plot the number of drops deposited vs. location
*                                along the swath
*      Output files: (UNIT=8) -SWATHDIST spray distribution for one pass
*      ------------  (UNIT=9) -UNIFORMTY gives uniformity measure of
*                                        combined passes for each swath width
*                                       ========================================
         SUBROUTINE VOLDST(ADJNUM,ADJSWA,ADJVOL,DIVS,DIV1S,INIVOL,LOCXX,
     +             NOZTYP,NUMB,SPRWID,UEPVTL,VOL)
*
         CHARACTER*1 ADJSWA,NOZTYP
         CHARACTER*3  NAME3
         CHARACTER*10 NAME7,NAME8
         INTEGER DIVS,DIV1S,I,J,K,KLESS,K2LESS,KMORE,K2MORE,
     +              MAX,MIN,N,NSWID(150),
```

```
     +              SPRWII,SWIDTH,SWMIN9,SWTHM9,TEMPS,VSWID(150)
         REAL  ADJNUM(DIVS),ADJVOL(DIVS),DNUMBR,DVOLUM,INIVOL,
     +          LOCXX(DIVS),NCOV(150),NMCFVR,NMSQSM,NMSTDV,NUMB(152),
     +          NUMAVE,NUMSUM,SPRWID,SWIDT2,SWVLSM,
     +          TEMP,UEPVPC,UEPVTL,UVLNLS,VLSTDV,VLCFVR,VLSQSM,
     +          VOL(152),VOLAVE,VOLNLS,VOLSUM,VCOV(150)
*
          NAME3='NEW'
          NAME7='-SWATHDIST'
          NAME8='-UNIFORMTY'
         OPEN  (UNIT=8,FILE=NAME7,STATUS=NAME3)
         OPEN  (UNIT=9,FILE=NAME8,STATUS=NAME3)
*
          VOLSUM  =  0
          VLSQSM  =  0
          SWVLSM  =  0
*                   Set  volume  sums  to  zero
*                      SWVLSM  =  Total  volume  sum  of  spray  deposited
          NUMSUM  =  0
          NMSQSM  =  0
*                   Set  number  sums  to  zero
*
       DO  110  I=1,DIVS
          WRITE  (8,150)  LOCXX(I),NUMB(I),VOL(I)
          SWVLSM  =  SWVLSM  +  VOL(I)
 110  CONTINUE
        IF  (ADJSWA.EQ.'S'.OR.ADJSWA.EQ.'s')  GOTO  190
          WRITE  (8,140)
 140      FORMAT  ('ADJACENT  SWATH')
*
       DO  160  I=1,DIVS
          WRITE  (8,150)  LOCXX(I),ADJNUM(I),ADJVOL(I)
 150      FORMAT  (3(E11.4,2X))
          SWVLSM  =  SWVLSM  +  ADJVOL(I)
 160  CONTINUE
*
*
 190     SPRWII  =  SPRWID
       DO  650  SWIDTH=10,SPRWII
*                               minimum  10  m  swath  width
          WRITE  (9,450)
          WRITE  (9,200)  SWIDTH
 200     FORMAT  (I5)
           MIN  =  (SPRWID  -  SWIDTH)  +  2
            SWIDT2  =  SWIDTH  *  2
           MAX  =  MIN  +  SWIDT2  -  1
*
        DO  430  K=MIN,MAX
              KLESS  =  K  +  SWIDT2
*                      swath  on  left  side  of  pass
              K2LESS  =  K  +  2*SWIDT2
*                      2nd  swath  on  left  side  of  pass
              KMORE  =  K  -  SWIDT2
*                      swath  on  right  side  of  pass
              K2MORE  =  K  -  2*SWIDT2
*                      2nd  swath  on  right  side  of  pass
              DVOLUM  =  VOL(K)
              DNUMBR  =  NUMB(K)
           IF  (ADJSWA.EQ.'o'.OR.ADJSWA.EQ.'O')  GOTO  360
              IF  (K2LESS.LE.DIV1S)  THEN
                 DVOLUM  =  DVOLUM  +  VOL(KLESS)  +  VOL(K2LESS)
                 DNUMBR  =  DNUMBR  +  NUMB(KLESS)  +  NUMB(K2LESS)
                 GOTO  340
               ENDIF
              IF  (KLESS.LT.DIV1S)  THEN
```

```
                    DVOLUM  =  DVOLUM  +  VOL(KLESS)
                    DNUMBR  =  DNUMBR  +  NUMB(KLESS)
                    GOTO  340
                 ENDIF
 340           IF  (K2MORE.GE.2)  THEN
                    DVOLUM  =  DVOLUM  +  VOL(KMORE)  +  VOL(K2MORE)
                    DNUMBR  =  DNUMBR  +  NUMB(KMORE)  +  NUMB(K2MORE)
                    GOTO  390
                 ENDIF
                IF  (KMORE.GE.2)  THEN
                    DVOLUM  =  DVOLUM  +  VOL(KMORE)
                    DNUMBR  =  DNUMBR  +  NUMB(KMORE)
                    GOTO  390
                 ENDIF
              GOTO  390
  360         IF  (K2LESS.LE.DIV1S)  THEN
                   DVOLUM  =  DVOLUM  +  ADJVOL(KLESS)  +  ADJVOL(K2LESS)
                   DNUMBR  =  DNUMBR  +  ADJNUM(KLESS)  +  ADJNUM(K2LESS)
                   GOTO  375
                ENDIF
               IF  (KLESS.LE.DIV1S)  THEN
                   DVOLUM  =  DVOLUM  +  ADJVOL(KLESS)
                   DNUMBR  =  DNUMBR  +  ADJNUM(KLESS)
                   GOTO  375
                 ENDIF
  375          IF  (K2MORE.GE.2)  THEN
                   DVOLUM  =  DVOLUM  +  ADJVOL(KMORE)  +  ADJVOL(K2MORE)
                   DNUMBR  =  DNUMBR  +  ADJNUM(KMORE)  +  ADJNUM(K2MORE)
                   GOTO  390
                 ENDIF
                IF  (KMORE.GE.2)  THEN
                   DVOLUM  =  DVOLUM  +  ADJVOL(KMORE)
                   DNUMBR  =  DNUMBR  +  ADJNUM(KMORE)
                   GOTO  390
                 ENDIF
*
 390      IF  (DNUMBR.LT.(1.0E-30))  DNUMBR=0.0
           IF  (DVOLUM.LT.(1.0E-30))  DVOLUM=0.0
             WRITE  (9,400)  LOCXX(K),DNUMBR,DVOLUM
  400       FORMAT  (3(E11.4,2X))
               VOLSUM  =  VOLSUM  +  DVOLUM
                VLSQSM  =VLSQSM+DVOLUM*DVOLUM
               NUMSUM  =  NUMSUM  +  DNUMBR
                NMSQSM  =NMSQSM+DNUMBR*DNUMBR
               DNUMBR  =  0
               DVOLUM  =  0
*                   Reset  to  zeros
  430     CONTINUE
*
            UEPVPC  =  100.0  *   UEPVTL  /  INIVOL
            UVLNLS  =  100.0  *   SWVLSM  /  UEPVTL
            VOLNLS  =  100.0  *   SWVLSM  /  INIVOL
            NUMAVE  =  NUMSUM  /  SWIDT2
*                                        number  mean  deposit
             IF  (NUMAVE.EQ.0)  THEN
                 WRITE  (9,440)
 440              FORMAT  ('---  NO  drops  in  swath  yet  ---')
                 GOTO  650
             ENDIF
            NMSTDV=((NMSQSM-NUMSUM*NUMSUM/SWIDT2)/(SWIDT2-1.0))**0.5
*                                         STANDARD  DEVIATION  of  drop  numbers
            NMCFVR  =  100.0  *  NMSTDV  /  NUMAVE
*                                         COEFFICIENT  OF  VARIATION  OF  drop  numbers
            VOLAVE  =  VOLSUM  /  SWIDT2
*                                          volume  mean  deposit
```

```
          VLSTDV=((VLSQSM-VOLSUM*VOLSUM/SWIDT2)/(SWIDT2-1.0))**0.5
*                                         STANDARD DEVIATION of drop volumes
          VLCFVR = 100.0 * VLSTDV / VOLAVE
*                                         COEFFICIENT OF VARIATION OF drop volumes
         WRITE (9,450)
 450     FORMAT ('          ')
         WRITE (9,460)
 460     FORMAT ('  NUMAVE        NMSTDV          NMCFVR      SWIDTH')
         WRITE (9,470) NUMAVE,NMSTDV,NMCFVR,SWIDTH
 470     FORMAT (3(E11.4,2X),I5)
         WRITE (9,480)
 480     FORMAT ('  VOLAVE        VLSTDV          VLCFVR')
         WRITE (9,490) VOLAVE,VLSTDV,VLCFVR
 490     FORMAT (3(E11.4,2X))
          VOLSUM = 0
          VLSQSM = 0
*               Reset volume sums to zero
          NUMSUM = 0
          NMSQSM = 0
*               Reset number sums to zero
*
          SWMIN9 = SWIDTH - 9
          NCOV(SWMIN9) = NMCFVR
          VCOV(SWMIN9) = VLCFVR
          NSWID(SWMIN9) = SWIDTH
          VSWID(SWMIN9) = SWIDTH
*
  650  CONTINUE
*
*        -----SORTING for Best SwathWIDTH according to NUMBER of drops
          N = SPRWID - 10
  690  DO 740 J=1,N
            IF (NCOV(J+1).GT.NCOV(J)) GOTO 740
            TEMP = NCOV(J)
            NCOV(J) = NCOV(J+1)
            NCOV(J+1) = TEMP
            TEMPS = NSWID(J)
            NSWID(J) = NSWID(J+1)
            NSWID(J+1) = TEMPS
  740  CONTINUE
         N = N - 1
         IF (N.GE.1) GOTO 690
*
*        -----SORTING for Best SwathWIDTH% according to VOLUME of drops
           N = SPRWID - 10
  800  DO 850 J=1,N
            IF (VCOV(J+1).GT.VCOV(J)) GOTO 850
            TEMP = VCOV(J)
            VCOV(J) = VCOV(J+1)
            VCOV(J+1) = TEMP
            TEMPS = VSWID(J)
            VSWID(J) = VSWID(J+1)
            VSWID(J+1) = TEMPS
  850  CONTINUE
         N = N - 1
         IF (N.GE.1) GOTO 800
*        =======
         WRITE (9,860)
 860     FORMAT ('          ')
         WRITE (9,870)
 870     FORMAT ('NMCFVR        SWIDTH      VLCFVR      SWIDTH')
           SWTHM9 = SPRWID - 9
        DO 890 J=1,SWTHM9
           WRITE (9,880) NCOV(J),NSWID(J),VCOV(J),VSWID(J)
 880      FORMAT (2(F8.3,2X,I5,2X))
```

```
 890    CONTINUE
*         =======
          WRITE (9,860)
          WRITE (9,900) INIVOL
  900     FORMAT ('Total initial spray volume (m**3) is ',E12.5)
          WRITE (9,910) UEPVTL
  910     FORMAT ('Total unevaporated volume of spray (m**3) is ',E12.5)
          WRITE (9,920) UEPVPC
  920     FORMAT ('Spray volume percentage unevaporated is ',F8.3)
          WRITE (9,930) UVLNLS
  930     FORMAT ('Percentage of unevaporated spray not lost',F8.3)
          WRITE (9,940) VOLNLS
  940     FORMAT ('Useful spray percentage of Initial volume ',F8.3)
          WRITE (9,950) NSWID(1)
  950     FORMAT ('Most uniform width (m) drop NUMBER-wise is ',I5)
          WRITE (9,960) VSWID(1)
  960     FORMAT ('Most uniform width (m) drop VOLUME-wise is ',I5)
*
          RETURN
          END
*        ============================================================
*
*
*
*
*          ******** NOZLOC ***** 17000 *********************************
*          SUBroutine to determine nozzle locations on span
*          ============================================================
*
*        Output file: (UNIT=11) -SPRAYINFO  for information for spraying
*          (see Main Program)    conditions, etc and final drop locations
*
           SUBROUTINE NOZLOC(DRPNUM,DRPRUN,LCCNTR,LFTNOZ,LFTWNG,
     +                               NOZLCN,NOZNUM,PCNSPN,RITNOZ,SWACNT)
*
           CHARACTER*1 NOZLCN
           INTEGER DRPRUN,LCCNTR,LFTNOZ,LFTWNG,M,RITNOZ,SWACNT
           REAL DRPNUM,LLNOZ(100),LOCINC,LRNOZ(100),
     +         MAXLOC,MINLOC,NOZNUM,PCNSPN
*
          IF (DRPRUN.GT.0) GOTO 150
          PRINT*, 'Even spacing of nozzles       =  E'
          PRINT*, 'Uneven spacing of nozzles     =  U'
 120     PRINT*, 'SPECIFY NOZZLE LOCATION TYPE (E,U) = '
          PRINT*, ' Be sure to enclose letter with apostrophes!'
          READ*, NOZLCN
*
 150    IF (NOZLCN.EQ.'E'.OR.NOZLCN.EQ.'e') GOTO 300
         IF (NOZLCN.EQ.'U'.OR.NOZLCN.EQ.'u') GOTO 500
         PRINT*, 'ERROR in specifying nozzle location type'
          GOTO 120
*        ======================================
*        Even spacing of nozzles along semi-span
*        ======================================
 300    IF (DRPRUN.EQ.0) GOTO 350
           GOTO 420
 350    PRINT*, 'How many nozzles to use per semi-span?'
          READ*, NOZNUM
           IF (NOZNUM.LT.2) GOTO 350
          PRINT*, 'Give MINimum percentage semi-span for nozzles '
          READ*, MINLOC
          PRINT*, 'Give MAXimum percentage semi-span for end nozzle '
          READ*, MAXLOC
          LOCINC = (MAXLOC-MINLOC)/(NOZNUM-1.0)
          WRITE (11,400) NOZNUM
```

```
  400     FORMAT ('There are',F4.0,' nozzles, EVENLY spaced, from')
            WRITE (11,405) MINLOC,MAXLOC
  405     FORMAT ('            ',F4.0,' to ',F4.0,' Percent semispan.')
  420     PCNSPN = MINLOC + LCCNTR * LOCINC
            LCCNTR = LCCNTR + 1
            DRPNUM = 0
*                             reset # of drops already done by nozzle
            RETURN
*          =========================================
*           Uneven spacing of nozzles along semi-span
*          =========================================
 500    IF (DRPRUN.EQ.0) GOTO 540
            GOTO 700
 540    PRINT*, 'How many NOZZLES on the RIGHT wing '
            READ*, RITNOZ
           PRINT*, 'How many NOZZLES on the LEFT wing   '
            READ*, LFTNOZ
           WRITE (11,550) RITNOZ
 550       FORMAT ('There are',I4,' nozzles on the right wing.')
           WRITE (11,560) LFTNOZ
 560       FORMAT ('There are',I4,' nozzles on the left wing.')
           IF (RITNOZ.EQ.0) THEN
               LFTWNG = 1
               SWACNT = SWACNT + 1
               GOTO 625
            ENDIF
          PRINT*, 'Now give the location of the nozzles on the RIGHT wing.'
 590   DO 620 M=1,RITNOZ
            PRINT*, 'RIGHT wing. What is the PERCENT SEMISPAN of nozzle num
      +ber ',M
            READ*, LRNOZ(M)
            WRITE (11,600) M,LRNOZ(M)
 600       FORMAT ('RIGHT wing. Nozzle number',I4,' is at ',F8.3,' PERCENT
      + SEMIspan.')
 620   CONTINUE
 625      IF (LFTNOZ.EQ.0) GOTO 700
           PRINT*, 'Now give the location of the nozzles on the LEFT wing.'
        DO 670 M=1,LFTNOZ
 650      PRINT*, 'LEFT wing. What is the PERCENT SEMISPAN of nozzle
      +     number   ',M
             READ*, LLNOZ(M)
           WRITE (11,660) M,LLNOZ(M)
 660       FORMAT ('LEFT wing. Nozzle number',I4,' is at ',F8.3,' PERCENT
      +SEMIspan.')
 670   CONTINUE
*
 700    LCCNTR = LCCNTR + 1
          IF (LFTWNG.EQ.1) GOTO 750
           IF (LCCNTR.GT.RITNOZ) RETURN
           PCNSPN = LRNOZ(LCCNTR)
*                    RIGHT wing nozzles
           DRPNUM = 0
*                    reset # of drops already done by nozzle
           RETURN
*
 750    IF (LCCNTR.GT.LFTNOZ) RETURN
          PCNSPN = LLNOZ(LCCNTR)
*                    LEFT wing nozzles
           DRPNUM = 0
*                    reset # of drops already done by nozzle
*
           RETURN
           END
*          ==============================================================
*
```

```
*
*
*
*          ****** NOZZLE **** 18000 ***************************************
*          SUBroutine, direction of initial velocity of drop w.r.t nozzle
*          ==============================================================
*
*       Output file: (UNIT=11) -SPRAYINFO    for information for spraying
*          (see Main Program)       conditions, etc and final drop locations
*
         SUBROUTINE NOZZLE(DNPRNZ,DRPNUM,DRPRUN,ENZVEL,NOZTYP,
     +                                   NOZX,NOZY,NOZZ)
*
         CHARACTER*1 CHGNOZ,NOZTYP
         INTEGER DRPRUN
         REAL CONANG,DNANGL,DNPRNZ,DRPNUM,ENZVEL,FANANG,HORANG,
     +         NOZX,NOZY,NOZZ,PI
*
         PI = 3.141593
*
   50    IF (DRPRUN.EQ.0) GOTO 90
           GOTO 190
   90    PRINT*, 'Nozzle types:  Single Drop        = S '
          PRINT*, '                    Hollow Cone        = H '
          PRINT*, '                    Flat Fan            = F '
          PRINT*, '                    Rotary              = R '
          PRINT*, 'Specify nozzle type desired   (S,H,F,R) '
          PRINT*, ' Be sure to enclose letter with apostrophes!'
          READ*, NOZTYP
  190    IF (NOZTYP.EQ.'S'.OR.NOZTYP.EQ.'s') GOTO 300
  200    IF (DRPRUN.GT.0) THEN
            GOTO 220
          ELSE
            PRINT*, 'How many drops per nozzle? '
            READ*, DNPRNZ
          ENDIF
  205     IF (DNPRNZ.LT.2)  GOTO 200
          WRITE (11,210) DNPRNZ
  210    FORMAT ('The number of drops per nozzle = ',F7.2)
  220    IF (NOZTYP.EQ.'H'.OR.NOZTYP.EQ.'h') GOTO 400
          IF (NOZTYP.EQ.'F'.OR.NOZTYP.EQ.'f') GOTO 500
          IF (NOZTYP.EQ.'R'.OR.NOZTYP.EQ.'r') GOTO 600
  250    PRINT*, 'ERROR specifying nozzle type.    REDO '
          GOTO 90
*
*       Single Drop Nozzle
*       ==================
  300    IF (DRPRUN.GT.0) THEN
            GOTO 380
          ELSE
            WRITE (11,320)
  320       FORMAT ('The nozzle is a SINGLE DROP nozzle.')
          ENDIF
          PRINT*, 'Default values:   NOZX=0,   NOZY=-1,   NOZZ=0 '
          PRINT*, '                     (ie. straight back)'
           NOZX=0.0
           NOZY=-1.0
           NOZZ=0.0
  330    PRINT*, '( NOZX +ve to right wing                    )'
          PRINT*, '( NOZY +ve in direction of flight path )'
          PRINT*, '( NOZZ +ve upwards                              )'
  340    PRINT*, 'Change nozzle direction (Y/N) '
          PRINT*, ' Be sure to enclose letter with apostrophes!'
          READ*, CHGNOZ
  350    IF (CHGNOZ.EQ.'Y'.OR.CHGNOZ.EQ.'y') GOTO 370
```

```
  360    IF (CHGNOZ.EQ.'N'.OR.CHGNOZ.EQ.'n') THEN
             GOTO 380
           ELSE
             GOTO 340
          ENDIF
  370    PRINT*, 'Desired nozzle direction    NOZX,NOZY,NOZZ'
          READ*, NOZX,NOZY,NOZZ
  380    DNPRNZ = 1
          DRPNUM = DRPNUM + 1
  390    RETURN
*
*
*        Hollow Cone Nozzle  (aiming straight back)
*        ==================
  400    IF (DRPRUN.GT.0) THEN
             GOTO 450
           ELSE
             PRINT*, 'What is the Cone Angle in degrees ?'
             READ*, CONANG
          ENDIF
          WRITE(11,430) CONANG
  430    FORMAT ('HOLLOW CONE nozzle with CONANG = ',F7.2,' degrees')
           CONANG = CONANG * PI /180.0
          PRINT*, 'What is nozzle angle w.r.t. horizontal  (0 to 90) ?'
          PRINT*, '      Straight back = 0           Down = 90 '
          READ*, HORANG
          WRITE (11,440) HORANG
  440    FORMAT ('The nozzle angle w.r.t. horizontal=',F7.2,' degrees')
           HORANG = HORANG * PI /180.0
  450    DNANGL = (2.0 * PI * DRPNUM/DNPRNZ)
          NOZX =   SIN(CONANG/2.0) * COS(DNANGL)
          NOZY =  -(COS(CONANG/2.0) * COS(HORANG) +
     +                     SIN(CONANG/2.0) * SIN(DNANGL) * SIN(HORANG))
          NOZZ =   -COS(CONANG/2.0) * SIN(HORANG) +
     +                     SIN(CONANG/2.0) * SIN(DNANGL) * COS(HORANG)
           DRPNUM = DRPNUM + 1
          RETURN
*
*        Flat Fan Nozzle
*        ===============
  500    IF (DRPRUN.GT.0) GOTO 550
             PRINT*, 'What is the Flat Fan Angle, degrees ?'
             READ*, FANANG
             WRITE (11,530)  FANANG
  530    FORMAT ('FLAT FAN nozzle with FAN ANGLE = ',F7.2,' degrees')
              FANANG = FANANG * PI / 180.0
          PRINT*, 'What is nozzle angle w.r.t. horizontal  (0 to 90) ? '
          PRINT*, '      Straight back = 0        Down = 90 '
          READ*, HORANG
          WRITE (11,540) HORANG
  540    FORMAT ('The nozzle angle w.r.t. horizontal=',F7.2,' degrees')
             HORANG = HORANG * PI /180.0
  550 NOZX= SIN((1.0-(DRPNUM*2.0/(DNPRNZ-1.0)))*FANANG/2.0)
        NOZY=-COS((1.0-(DRPNUM*2.0/(DNPRNZ-1.0)))*FANANG/2.0)*COS(HORANG)
        NOZZ=-COS((1.0-(DRPNUM*2.0/(DNPRNZ-1.0)))*FANANG/2.0)*SIN(HORANG)
          DRPNUM = DRPNUM + 1
        RETURN
*
*        Rotary Nozzle
*        =============
  600    IF (DRPRUN.GT.0) GOTO 650
             WRITE (11,620)
  620    FORMAT ('This is a ROTARY nozzle.')
          PRINT*, 'What is the angle of the axis of the spinner w.r.t.'
          PRINT*, '    the horizontal, in degrees (0 to 90) '
```

```
         READ*, HORANG
         WRITE (11,630)
  630   FORMAT ('The angle of the axis of the spinner w.r.t.')
         WRITE (11,640) HORANG
  640   FORMAT ('              the horizontal = ',F7.2,' degrees')
            HORANG = HORANG * PI/180.0
         PRINT*, 'What is velocity of the drop off of the spinner (m/s)'
          READ*, ENZVEL
         WRITE (11,645) ENZVEL
  645   FORMAT ('The velocity of the drop off of the spinner (m/s) =',
     +        F7.2)
  650   NOZX = COS(2.0 * PI * DRPNUM/DNPRNZ)
         NOZY = SIN(2.0 * PI * DRPNUM/DNPRNZ) * SIN(HORANG)
         NOZZ = SIN(2.0 * PI * DRPNUM/DNPRNZ) * COS(HORANG)
          DRPNUM = DRPNUM + 1
*
        RETURN
        END
```

10. APPENDIX II. NASA DATA ANALYSIS MODEL

The program was tested using the data from Morris *et al*. (1984). In order to do so, the model was modified to accept the input conditions for each run through two data files. One file, called NASAPLAN, contains the information which remains constant regarding the aircraft during all test runs. The other file, called NASADATA, contains the information which is run dependent, such as the crosswind. Using this information for each run the final droplet position could be determined, to be later compared with the positions of measured deposition from Morris *et al*. (1984).

A listing of the program NASA77 is included after the explanation of the program input and output.

Program Input

The file NASAPLAN contains the following information, which remains constant throughout the flight tests. For the case when the left wing of the aircraft is being tested, only the sign of the swirl coefficient is changed. The explanation of terms which is given below is not part of the file.

<u>NASAPLAN</u>

12.625	Span
2.286	Chord
3.5	Dihedral per wing, degrees
0.0775	Core coefficient
2.7432	Propeller diameter, m
0.6096	Propeller height from t.e. of wing, m
1300	Propeller RPM
0.0040	Swirl coefficient for propeller
-0.3048	Y Position of nozzle from t.e. of wing, m
-0.4572	Z Position of nozzle from t.e. of wing, m
0	X direction of nozzle emission
-1	Y direction of nozzle emission
0	Z direction of nozzle emission
0	Nozzle pressure, Pa
650	Droplet density, kg/m^3
3.6576	Measured crosswind height, m
0.6096	Physical roughness height, m
0.6096	Height of collector array at centerline, m
0	Slope of collector array, %
-2	Slope of ground, m/100m

The file NASADATA contains the following information for each individual pass of the aircraft over the collector arrays. This information for each pass of the aircraft is given on one horizontal line in the file. The actual listing of the file is given in Appendix III.

NASADATA

50	Percent spanwise location of nozzle
5851	Weight of aircraft, lb
113.2	Velocity of aircraft, knots
14	Height of aircraft wing t.e., ft
-2.23	Measured crosswind, ft/s
650	Droplet diameter, micrions

Program Output

The output from NASA77 is in the temporary file -NASAINFO. Included in the output is the final lateral location for the drop movement for each drop of the flight test data set. This is done for each of the initial trailing vortex separations from 82% to 100% span.

Program Listing

A listing of the program NASA77 starts on the next page.

```
*        ******************************************************************
*        SIMULATION OF AIRCRAFT WAKE - PARTICLE INTERACTION
*        for M.Sc. Thesis                    Filename = NASA77
*        CRAIG S. MERKL, B.Sc.                          Sept 19, 1988
*        University of Alberta
*           This model is used to test the data from the NASA flight
*              tests done by Morris, et. al. (1984).
*           The model presents the vortex system describing the aircraft
*             and the vortex images for the ground plane, the crosswind,
*             the spray nozzle emission velocity, and the aircraft velocity.
*             A uniformly sloping ground is allowed for in this model.
*           The spray droplet size, etc. is entered, then the program
*             determines the resulting motion of the droplet in the wake,
*             and the final position of the droplet, hence the distribution
*             of spray.
*       **  Notice that the first DO loop gives the range of initial
*              spanwise vortex separations which will be used.
*       **  The second DO loop gives the number of data points in the
*              flight test data file NASADATA
*
*        Language:   FORTRAN 77
*
*        SUBROUTINEs included:       CONSTS
*                                              INPUTS
*                                              INITAL
*                                              INDVEL
*                                              NZVELS
*                                              RUNKUT
*                                              VRTCOR
*                                              DRAGCF
*                                              CHDELT
*
*        Input Files:           NASADATA      span location of nozzle, weight,
*                                                  speed, height, crosswind, drop size
*                                                        (UNIT=12)   see INPUTS
*                                    NASAPLAN      aircraft and operation data
*                                                        (UNIT=13)   see Main Program
*        Output Files:          -NASAINFO     location and time for drop for
*           (temporary files)                      each pass of the aircraft
*                                                        (UNIT=11)
*
*        ******************************************************************
         PROGRAM   NASA77
*        ------------------------------------------------------------------
*
         INTEGER CHK,DRPRUN,IISPSP
         REAL ACELX,ACELY,ACELZ,
     +         CD,CFXWND,CHORD,CONST1,CORCOF,CORRAD,CRCLAT,
     +         DELT,DENAIR,DENDRP,DIA,DIAMIC,DIHDRL,
     +         FHIGHT,HEIGHT,HT,HTCOLA,INIDIA,INISEP,INIVOL,
     +         INIVLX,INIVLY,INIVLZ,INILOX,INILOY,INILOZ,
     +         KINVSC,LOCX,LOCY,LOCZ,MSXWND,MSXWHT,
     +         NOZX,NOZY,NOZZ,NZPRES,NZVELX,NZVELY,NZVELZ,
     +         PCNSPN,PI,POSIZ,POSIY,PREVRE,
     +         PRPDIA,PRPHIT,PRPRPM,PSVSEP,RE,RELAST,
     +         SLCOLA,SLPANG,SLPGND,SPAN,SRFHIT,SWRCOF,SWRROT,TIME,
     +         VELREL,VELX,VELY,VELZ,VELXVT,VELYVT,VELZVT,VLPLAN,VLXWND,
     +         VLXREL,VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLXVT7,VLXVT8,
     +         VLYREL,VLYVT5,VLYVT6,VLZREL,VLZVT1,VLZVT2,
     +         VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +         VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +         VRTLY1,VRTLY2,VRTLY3,VRTLY4,
     +         VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8,
     +         VX,VY,VZ,WEIGHT,X,Y,Z
*
```

```
        OPEN  (UNIT=11,FILE='-NASAINFO',STATUS='NEW')
        OPEN  (UNIT=12,FILE='NASADATA',STATUS='OLD')
        OPEN  (UNIT=13,FILE='NASAPLAN',STATUS='OLD')
*
       READ  (13,*)  SPAN,CHORD,DIHDRL,CORCOF,PRPDIA,PRPHIT,PRPRPM,
     +        SWRCOF,POSIY,POSIZ,NOZX,NOZY,NOZZ,NZPRES,DENDRP,
     +        MSXWHT,PSHGHT,HTCOLA,SLCOLA,SLPGND
*
       SWRCF2  =  -SWRCOF
       SLPGN2  =  -SLPGND
       SLCOL2  =  -SLCOLA
          DRPRUN  =  0
      DO  500  IISPSP  =  82,100
*              range  of  initial  percentage  span  vortex  separations
        PSVSEP  =  IISPSP
*      IF  (DRPRUN.EQ.0)  GOTO  345
       WRITE  (11,340)  IISPSP
 340  FORMAT  (I3)
*
 345  DO  495  NUMPTS  =  1,136
*                        number  of  data  points  in  NASADATA
*      first  68  points  right  wing,  next  68  left  wing  data
       IF  (NUMPTS.GE.69)  THEN
         SWRCOF  =  SWRCF2
         SLPGND  =  SLPGN2
         SLCOLA  =  SLCOL2
        ENDIF
 350  CALL  CONSTS(CD,CHK,CL,DENAIR,DENDRP,GRAV,KINVSC,
     +        PI,RE,TIME)
*                        Constants  and  Coefficients
*
       CALL  INPUTS(CHORD,CORCOF,DELT,DENDRP,DIA,DIAMIC,
     +              DIHDRL,HEIGHT,HT,HTCOLA,
     +              INIDIA,INISEP,MSXWND,MSXWHT,
     +              NOZX,NOZY,NOZZ,NZPRES,NZVELX,NZVELY,NZVELZ,
     +              PCNSPN,POSIY,POSIZ,PRPDIA,PRPHIT,PRPRPM,PSHGHT,PSVSEP,
     +              SLCOLA,SLPGND,SPAN,SRFHIT,SWRCOF,SWRROT,
     +              VLPLAN,WEIGHT)
*                        Input  values
*
       CALL  INITAL(CFXWND,CHORD,CONST1,CORCOF,CORRAD,
     +        CRCLAT,DELT,DENAIR,DENDRP,DIA,DIAMIC,DIHDRL,DRPRUN,
     +        HEIGHT,HT,HTCOLA,IISPSP,INIDIA,INISEP,INILOX,INILOY,INILOZ,
     +        INIVLX,INIVLY,INIVLZ,KINVSC,LOCX,LOCY,LOCZ,
     +        MSXWHT,MSXWND,NOZX,NOZY,NOZZ,NZPRES,NZVELX,NZVELY,NZVELZ,
     +        PCNSPN,POSIY,POSIZ,PREVRE,PRPDIA,PRPHIT,PRPRPM,PSVSEP,
     +        RE,RELAST,SEP,SLCOLA,SLPANG,SLPGND,SPAN,SRFHIT,
     +        SWRCOF,SWRROT,TIME,VELX,VELY,VELZ,VX,VY,VZ,
     +        VLPLAN,VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +        VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8,WEIGHT)
*                        Initial  conditions
*
        IF  (LOCX.LT.0)  GOTO  495
*
 410  CALL  RUNKUT(ACELX,ACELY,ACELZ,CD,CFXWND,CHK,CONST1,
     +    CORRAD,CRCLAT,DELT,DIA,FHIGHT,
     +    HEIGHT,HTCOLA,INIDIA,KINVSC,LOCX,LOCY,LOCZ,
     +    PRPDIA,RE,RELAST,SEP,SLCOLA,SLPANG,SRFHIT,SWRROT,TIME,
     +    VELREL,VELX,VELY,VELZ,VELXVT,VELYVT,VELZVT,VLXWND,VLPLAN,
     +    VLXREL,VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLXVT7,VLXVT8,
     +    VLYREL,VLYVT5,VLYVT6,VLZREL,
     +    VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +    VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,VRTLY1,VRTLY2,
     +    VRTLY3,VRTLY4,VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8,
     +    VX,VY,VZ,X,Y,Z)
```

```
*                    Simulation routine (Runge-Kutta fourth order)
*
*
 440     CALL CHDELT(CHK,DELT,DIA,RE)
*                    Subroutine to Modify time increment, DELT
              IF (TIME.GE.(20.0)) GOTO 475
*                           End droplet simulation at 20 sec
 470     IF (LOCZ.GT.FHIGHT) GOTO 410
*         -----------------------------
*
 475     WRITE (11,480) LOCX,LOCZ,TIME
 480      FORMAT (2(F8.3,2X),F8.3)
 495     CONTINUE
            REWIND (UNIT=12)
 500    CONTINUE
*
          END
*       ****************************************************************
*       --------End of Main Program-------------------------------------
*       ****************************************************************
*
*
*
*          ****** CONSTS ********* 1000 *********************************
*          Constants and Coefficients
*          -------------------------------------------------------------
*
          SUBROUTINE CONSTS(CD,CHK,CL,DENAIR,DENDRP,GRAV,KINVSC,
     +         PI,RE,TIME)
*
          INTEGER CHK
          REAL CD,CL,DENAIR,DENDRP,DYVISC,GRAV,KINVSC,
     +         PI,RE,TIME
*
           CHK=0
*                   used as counters, see CHDELT subroutine
           CD = 0
*                   drag coefficient of droplet
*         CHORD      chord of aircraft wing, m
           CL = 0
*                   lift coefficient of wing
*         CRCLAT                circulation around wing, m**2/s
           DENAIR = 1.2256
*                             density of air, kg/m**3   (Sea Level)
*         DENDRP = 650.0
*          density of SIMULATED droplet (Morris,et.al.(1984)), kg/m**3
*         DIA                   diameter of droplet, m
*         DIAMIC              Initial DIAmeter of droplet in microns
           DYVISC = 0.0000178
*                             dynamic viscosity of air, N*s/m**2 (Sea Level)
*         FHIGHT                Final Height where drop hits ground
           GRAV = 9.80665
*                             acceleration of gravity, m/s**2
*         HEIGHT                height of Trailing Edge of wing ROOT AGL, m
*         HT = HEIGHT + POSIZ   height of nozzle above ground, m
           KINVSC = DYVISC / DENAIR
*                             kinematic viscosity of air
*         LIFT               lift of wing, N
*         M                  mass of droplet, kg
*         NZPRES      Nozzle pressure, Pa  (but entered in kPa)
*         PCNSPN     percent span location of nozzle
           PI = 3.141593
*         POSIZ      POSIY       position of nozzle below and behind wing, m
*         PSHGHT     Physical height of surface covering (m) (nonzer0)
*
```

```
          RE = 0
*                   Reynolds number of droplet
*         RELAST    Reynolds number of droplet previous loop
*         SEP        half separation between trailing vortices, m
*         SPAN       wingspan of aircraft wing, m
          TIME = 0
*                   time from emission of droplet, sec
*         WEIGHT             weight of airplane, N
*
*
*         Distance definitions and Initialization of variables
*         ------------------------------------------------
*         LOCX                        location of drop on X-axis, m
*         LOCY                        location of drop on Y-axis, m
*         LOCZ                        location of drop on Z-axis, m
*
*         R1                           distance from drop to vortex core #1, m
*         R2                           distance from drop to vortex core #2, m
*         R3                           distance from drop to vortex core #3, m
*         R4                           distance from drop to vortex core #4, m
*         R5                           distance from drop to vortex core #5, m
*         R6                           distance from drop to vortex core #6, m
*         R7                           distance from drop to slipstream center, m
*
*         X1                           local x component for vortex 1, m
*         X2                           local x component for vortex 2, m
*         X3                           local x component for vortex 3, m
*         X4                           local x component for vortex 4, m
*         X5                           local x component for vortex 5, m
*         X6                           local x component for vortex 6, m
*         X7                           local x component for slipstream, m
*         X8                           local x component for slipstream image, m
*         Y1                           local y component for vortex 1, m
*         Y2                           local y component for vortex 2, m
*         Y3                           local y component for vortex 3, m
*         Y4                           local y component for vortex 4, m
*         Y5                           local y component for vortex 5, m
*         Y6                           local y component for vortex 6, m
*         Y7                           local y component for slipstream, m
*         Z1                           local z component for vortex 1, m
*         Z2                           local z component for vortex 2, m
*         Z3                           local z component for vortex 3, m
*         Z4                           local z component for vortex 4, m
*         Z5                           local z component for vortex 5, m
*         Z6                           local z component for vortex 6, m
*         Z7                           local z component for slipstream, m
*         Z8                           local z component for slipstream image, m
*
*         velocity definitions and initialization
*         --------------------------------------
*         VELX                 actual x velocity of drop, m/s
*         VELY                 actual y velocity of drop, m/s
*         VELZ                 actual z velocity of drop, m/s
*
*         NZVELX              initial x vel of fluid relative to nozzle, m/s
*         NZVELY              initial y vel of fluid relative to nozzle, m/
*         NZVELZ              initial z vel of fluid relative to nozzle, m/s
*         VLPLAN              velocity of plane (along y axis), m/s
*         VLXWND              x component of wind (crosswind), m/s
*         VLYWND              y component of wind, m/s
*
*         VLXVT1              x component of induced velocity of vortex 1, m/s
*         VLXVT2              x component of induced velocity of vortex 2, m/s
*         VLXVT3              x component of induced velocity of vortex 3, m/s
*         VLXVT4              x component of induced velocity of vortex 4, m/s
```

```
*        VLYVT5               y component of induced velocity of vortex 5, m/s
*        VLYVT6               y component of induced velocity of vortex 6, m/s
*        VLZVT1               z component of induced velocity of vortex 1, m/s
*        VLZVT2               z component of induced velocity of vortex 2, m/s
*        VLZVT3               z component of induced velocity of vortex 3, m/s
*        VLZVT4               z component of induced velocity of vortex 4, m/s
*        VLZVT5               z component of induced velocity of vortex 5, m/s
*        VLZVT6               z component of induced velocity of vortex 6, m/s
*
*        VELXVT               resultant induced x velocity, m/s
*        VELYVT               resultant induced y velocity, m/s
*        VELZVT               resultant induced z velocity, m/s
*
*        VLXREL               relative x velocity, drop to air, m/s
*        VLYREL               relative y velocity, drop to air, m/s
*        VLZREL               relative z velocity, drop to air, m/s
*        VELREL               resultant relative velocity, drop to air, m/s
*
*
*        acceleration definitions and initialization
*        ------------------------------------------
*        ACELX              x acceleration of drop, m/s**2
*        ACELY              y acceleration of drop, m/s**2
*        ACELZ              z acceleration of drop, m/s**2
*
*        angle definitions and initialization
*        -----------------------------------
*        BETA1              angle from drop to end of vortex 1
*        BETA2              angle from drop to end of vortex 2
*        BETA3              angle from drop to end of vortex 3
*        BETA4              angle from drop to end of vortex 4
*        BETA5A             angle from drop to one end of vortex 5
*        BETA5B             angle from drop to other end of vortex 5
*        BETA6A             angle from drop to one end of vortex 6
*        BETA6B             angle from drop to other end of vortex 6
*
*        ALPHA1             angle from horizontal plane of vortex 1 to drop
*        ALPHA2             angle from horizontal plane of vortex 2 to drop
*        ALPHA3             angle from horizontal plane of vortex 3 to drop
*        ALPHA4             angle from horizontal plane of vortex 4 to drop
*        ALPHA5             angle from horizontal plane of vortex 5 to drop
*        ALPHA6             angle from horizontal plane of vortex 6 to drop
*
         RETURN
         END
*        ------------------------------------------------------
*
*
*
*
*
*
*         ******* INPUTS ******* 4000 **********************************
*         SUBROUTINE for Input values
*         ---------------------------------------------------------------
*
*      Input Files:           NASADATA     span location of nozzle, weight,
*                                        speed, height, crosswind, drop size
*                                                   (UNIT=12)   see INPUTS
*
         SUBROUTINE INPUTS(CHORD,CORCOF,DELT,DENDRP,DIA,DIAMIC,
     +              DIHDRL,HEIGHT,HT,HTCOLA,
     +              INIDIA,INISEP,MSXWND,MSXWHT,
     +              NOZX,NOZY,NOZZ,NZPRES,NZVELX,NZVELY,NZVELZ,
     +              PCNSPN,POSIY,POSIZ,PRPDIA,PRPHIT,PRPRPM,PSHGHT,PSVSEP,
     +              SLCOLA,SLPGND,SPAN,SRFHIT,SWRCOF,SWRROT,
```

```
     +               VLPLAN,WEIGHT)
*
          CHARACTER*1 UNITS
          REAL  CHORD,CORCOF,DELT,DENDRP,DIA,DIAMIC,DIHDRL,
     +        GRAV,HEIGHT,HT,HTCOLA,INIDIA,INISEP,MSXWND,MSXWHT,
     +        NOZX,NOZY,NOZZ,NZCOEF,NZPRES,NZVELX,NZVELY,NZVELZ,PCNSPN,
     +        PI,POSIY,POSIZ,PRPDIA,PRPHIT,PRPROT,PRPRPM,PSHGHT,PSVSEP,
     +        SLCOLA,SLPGND,SPAN,SRFHIT,SWRCOF,SWRROT,
     +        VLPLAN,WEIGHT
*
           GRAV = 9.80665
           PI = 3.141593
*
*  90     PRINT*, 'SI units or Imperial --- (S or I) '
*          READ*, UNITS
*
*        ----------Input regarding the aircraft and operation
          UNITS = 'I'
         READ (12,*) PCNSPN,WEIGHT,VLPLAN,HEIGHT,MSXWND,DIAMIC
* 145    PRINT*, 'The weight of the plane (N or lb) is '
*           READ*, WEIGHT
         IF (UNITS.EQ.'i'.OR.UNITS.EQ.'I') WEIGHT=WEIGHT*0.453592*GRAV
*         PRINT*, 'The speed of the plane (m/s or knots) is '
*           READ*, VLPLAN
         IF (UNITS.EQ.'i'.OR.UNITS.EQ.'I') VLPLAN=
     +                               (VLPLAN*6080.0/3600.0)*0.3048
*         PRINT*, 'Height of wing root trailing edge (m or ft) AGL is '
*           READ*, HEIGHT
         IF (UNITS.EQ.'i'.OR.UNITS.EQ.'I')   HEIGHT=HEIGHT*0.3048
*         PRINT*, 'What is percentage slope of ground at centerline?'
*         PRINT*, '           (positive is down off right wing)'
*           READ*, SLPGND
*         PRINT*, 'The wingspan of the plane (m or ft) is '
*           READ*, SPAN
*        IF (UNITS.EQ.'i'.OR.UNITS.EQ.'I')   SPAN=SPAN*0.3048
*         PRINT*, 'The wing chord of the plane (m or ft) is '
*           READ*, CHORD
*        IF (UNITS.EQ.'i'.OR.UNITS.EQ.'I')   CHORD=CHORD*0.3048
*         PRINT*, 'The wing DIHeDRaL (degrees per wing) is '
*           READ*, DIHDRL
*         PRINT*, 'The percent span initial vortex separation ='
*           READ*, PSVSEP
*         PRINT*, 'The default value of CORCOF = 0.0775 '
*             CORCOF = 0.0775
*         PRINT*, 'Nozzle Z location w.r.t. trailing edge  (m) ='
*         PRINT*, '         (upwards is positive)'
*           READ*, POSIZ
*         PRINT*, 'Nozzle Y location w.r.t. trailing edge  (m) = '
*         PRINT*, '                   (forward is positive) '
*           READ*, POSIY
*
*
*         ----------Droplet and movement information
* 280    PRINT*, 'Strength of the crosswind at 3.048 m (m/s or ft/s)'
*           READ*, MSXWND
         IF (UNITS.EQ.'i'.OR.UNITS.EQ.'I') MSXWND=MSXWND *0.3048
*         PRINT*, 'HEIGHT at which the crosswind is measured (m or ft)'
*           READ*, MSXWHT
*        IF (UNITS.EQ.'i'.OR.UNITS.EQ.'I') MSXWHT = MSXWHT * 0.3048
*         PRINT*, 'What is height of collector array (final height) '
*         PRINT*, '                   at the centerline where the drop hits ?'
*           READ*, HTCOLA
*         PRINT*, 'What is percentage slope of collector array ?'
*         PRINT*, '         (positive is down off right wing)'
*           READ*, SLCOLA
```

```
* 320    PRINT*, 'Physical height of surface covering (m) (nonzer0) = ?'
*           READ*, PSHGHT
*        IF (PSHGHT.LE.0) GOTO 320
*                                 p 6  Morris et al.   1984
*          PRINT*, 'The initial size of the droplet (microns) is '
*           READ*, DIAMIC
*          PRINT*, 'Nozzle pressure in kPa =   ? '
*           READ*, NZPRES
*            NZPRES = NZPRES * 1000
*
*        ---------
*
  380    INIDIA = DIAMIC / 1000000.0
              DIA = INIDIA
         CALL NZVELS(DENDRP,NOZX,NOZY,NOZZ,
     +         NZPRES,NZVELX,NZVELY,NZVELZ)
*                Subroutine for nozzle velocities along coordinate axis
*
*
          HT = HEIGHT + POSIZ
          SRFHIT = PSHGHT / 30.0
*        'Notice 1/30 of physical height of surface covering
*                                           p6 Morris, et al.   1984
          DELT = 0.000625
           IF (DIA.LT.(0.0001)) DELT = DELT/10.0
           IF (DIA.LT.(0.00001)) DELT = DELT/10.0
*
*        ---- Propeller information
*         PRINT*, 'What is Propeller height from trailing edge (m) ?'
*        READ*, PRPHIT
*         PRINT*, ' What is Propeller diameter (m) = '
*        READ*, PRPDIA
*         PRINT*, 'What is Propeller RPM =   '
*         PRINT*, '    positive is clockwise from pilots view   '
*         PRINT*, '          negative if counterclockwise '
*        READ*, PRPRPM
*         PRINT*, '    What is the SWiRlCOeF =   '
*         PRINT*, '(= slipstream angular velocity divided by the'
*         PRINT*, '(  angular velocity of the propeller, 0.015 suggested)'
*        READ*, SWRCOF
 740     PRPROT = PRPRPM*2.0*PI/60.0
*                   angular velocity of the propeller, rad/s
          SWRROT =(PRPDIA/2)* PRPROT * SWRCOF
*                angular velocity of the slipstream, rad/s.
*                  (positive is clockwise from pilot'view)
*
*        ---- Vortex information
*            CORCOF = 0.0775
*           (CORCOF =core radius/span=0.0775  Sprieter & Sacks,1951)
*800     INISEP=PSVSEP*SPAN/100.0
*                                 initial vortex separation horiz, m
*
*
         RETURN
         END
*        -------------------------------------------------------------
*
*
*
*
*         ******** INITAL ******** 5000 *********************************
*         Initial Conditions
*         -------------------------------------------------------------
*
*        Output File:        -NASAINFO    for location of drop for each
```

```
*          (temporary file)         pass  of  the  aircraft
*                                                 (UNIT=11)
*
         SUBROUTINE  INITAL(CFXWND,CHORD,CONST1,CORCOF,CORRAD,
     +        CRCLAT,DELT,DENAIR,DENDRP,DIA,DIAMIC,DIHDRL,DRPRUN,
     +        HEIGHT,HT,HTCOLA,IISPSP,INIDIA,INISEP,INILOX,INILOY,INILOZ,
     +        INIVLX,INIVLY,INIVLZ,KINVSC,LOCX,LOCY,LOCZ,
     +        MSXWHT,MSXWND,NOZX,NOZY,NOZZ,NZPRES,NZVELX,NZVELY,NZVELZ,
     +        PCNSPN,POSIY,POSIZ,PREVRE,PRPDIA,PRPHIT,PRPRPM,PSVSEP,
     +        RE,RELAST,SEP,SLCOLA,SLPANG,SLPGND,SPAN,SRFHIT,
     +        SWRCOF,SWRROT,TIME,VELX,VELY,VELZ,VX,VY,VZ,
     +        VLPLAN,VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +        VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8,WEIGHT)
*
         INTEGER  DRPRUN,IISPSP
         REAL  ACELX,ACELY,ACELZ,CD,CFXWND,CHORD,CONST1,CORCOF,CORRAD,
     +          CRCLAT,DELT,DENAIR,DENDRP,DIA,DIAMIC,DIHDRL,
     +          HEIGHT,HT,HTCOLA,
     +          INIDIA,INISEP,INILOX,INILOY,INILOZ,INIVLX,INIVLY,INIVLZ,
     +          KINVSC,LFTCOF,LFTCVS,LIFT,LOCX,LOCY,LOCZ,LWNGHT,MSXWND,
     +          MSXWHT,NNELCR,NOZX,NOZY,NOZZ,NZPRES,NZVELX,NZVELY,NZVELZ,
     +          PSVSEP,PCNSPN,PI,POSIY,POSIZ,PREVRE,
     +          PRPDIA,PRPHIT,PRPRPM,RE,RELAST,RWNGHT,SSPGND,
     +          SEP,SLCOLA,SLPANG,SLPGND,SPAN,SRFHIT,SWRCOF,SWRROT,TIME,
     +          VELX,VELY,VELZ,VELXVT,VELYVT,VELZVT,VLPLAN,VLXWND,
     +          VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +          VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +          VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
     +          VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
     +          VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +          VTHFTE,VTHTDI,VX,VY,VZ,WEIGHT,X,Y,Z
*
            PI  =  3.141593
*                 location  of  droplet  at  the  start
            LOCX  =  (SPAN/2.0)  *  PCNSPN/100.0
            IF  (LOCX.LT.0)  RETURN
            LOCY  =  -0.75  *  CHORD  +  POSIY
            LOCZ  =  HT  +  LOCX  *  TAN(DIHDRL*PI/180.0)
*                        initial  X  position  of  drop
*                             "      Y       "       "     "
*                             "      Z       "       "     "
            INILOX  =  LOCX
            INILOY  =  LOCY
            INILOZ  =  LOCZ
*            location  of  drop  at  the  start,  see  12620  and  12660
*            velocity  of  drop  at  the  start,  also  see  12620
            VELX  =  NZVELX
            VELY  =  NZVELY  +  VLPLAN
            VELZ  =  NZVELZ
             INIVLX  =  VELX
             INIVLY  =  VELY
             INIVLZ  =  VELZ
*
4          CFXWND  =  MSXWND  /  LOG(MSXWHT/SRFHIT)
           VLXWND  =  CFXWND  *  LOG(LOCZ/SRFHIT)
*
           LFTCOF=WEIGHT*2.0/(DENAIR  *  VLPLAN  *  VLPLAN  *  CHORD  *  SPAN)
           NNELCR  =  0.1
*                    Correction  for  nonelliptical  planform
           LFTCVS  =  (2.0*PI)/(1.0+2.0*(1.0+NNELCR)/(SPAN/CHORD))
*                      LFTCVS  =  lift  curve  slope
           VTHFTE  =  SIN(LFTCOF/LFTCVS)  *  CHORD*3/4
*                vortex  height  w.r.t.  the  trailing  edge  of  the  wing  tip
*                                                            due  to  angle  of  attack
           CORRAD  =  CORCOF  *  SPAN
```

```
*                                    vortex core radius
          LIFT = WEIGHT
          CRCLAT = ((1+4/PI)/2)*(LIFT/(DENAIR * VLPLAN * SPAN))
*                                    lift & circulation of wing
          IF (HEIGHT.GE.SPAN) THEN
             INISEP = 0.82 * SPAN
           ELSE
             INISEP = (PSVSEP-(PSVSEP-82)*(HEIGHT/SPAN))*(SPAN/100.0)
          ENDIF
          SEP = INISEP / 2.0
           VTHTDI = SEP * TAN((DIHDRL*PI)/180)
*         added vortex height w.r.t. the wing root, due to wing dihedral
           CONST1 = 0.75 * DENAIR / DENDRP
*         initial vortex locations
             VRTLX1 = SEP
             VRTLX2 = -VRTLX1
             VRTLX7= 0
             VRTLZ1 = HEIGHT + VTHFTE + VTHTDI
             VRTLZ2 = VRTLZ1
             VRTLZ7 = HEIGHT + PRPHIT
*         initial vortex image locations
          SSPGND = 1.0
          SSPGND = SIGN(SSPGND,SLPGND)
          SLPANG = ATAN(SLPGND/100)
          RWNGHT = VRTLZ1 + VRTLX1 * SLPGND/100
          LWNGHT = VRTLZ1 + VRTLX2 * SLPGND/100
             VRTLX3 = VRTLX1 - (2*RWNGHT*SIN(SLPANG)*SIN(SLPANG)*SSPGND)
             VRTLX4 = VRTLX2 - (2*LWNGHT*SIN(SLPANG)*SIN(SLPANG)*SSPGND)
             VRTLX8 = VRTLX7 - (2*VRTLZ7*SIN(SLPANG)*SIN(SLPANG)*SSPGND)
             VRTLZ3 = VRTLZ1 - (2*RWNGHT*COS(SLPANG)*COS(SLPANG))
             VRTLZ4 = VRTLZ2 - (2*LWNGHT*COS(SLPANG)*COS(SLPANG))
             VRTLZ8 = VRTLZ7 - (2*VRTLZ7*COS(SLPANG)*COS(SLPANG))
*
           X=LOCX
           Y=LOCY
           Z=LOCZ
           VX=VELX
           VY=VELY
           VZ=VELZ
*
*
         CALL INDVEL(ACELX,ACELY,ACELZ,CD,CONST1,CORRAD,CRCLAT,
     +        DIA,HEIGHT,KINVSC,PRPDIA,RE,SEP,SLPANG,SWRROT,TIME,
     +        VELXVT,VELYVT,VELZVT,VLPLAN,VLXWND,
     +        VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +        VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +        VLXREL,VLYREL,VLZREL,VELREL,
     +        VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
     +        VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
     +        VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +        VX,VY,VZ,X,Y,Z)
*                          to get initial Reynolds number
            RELAST = RE
*
*         This line skips print out of information
           IF (DRPRUN.GE.0) GOTO 550
         WRITE (11,10)
 10      FORMAT ('            ')
         WRITE (11,20) WEIGHT
 20      FORMAT ('The WEIGHT of the plane (N)=',F12.2)
         WRITE (11,30) VLPLAN
 30      FORMAT ('The SPEED of the plane (m/s)=',F8.3)
         WRITE (11,40) HEIGHT
 40      FORMAT ('HEIGHT of the wing root trailing edge AGL (m)=',F8.3)
         WRITE (11,45)
```

```
      WRITE (11,46) SLPGND
45     FORMAT ('Percentage slope of ground at centerline, ')
46     FORMAT (' where positive is down off right wing =',F8.2)
      WRITE (11,50) SPAN
50     FORMAT ('The wing SPAN of the plane (m)=',F8.3)
      WRITE (11,60) CHORD
60     FORMAT ('The wing CHORD of the plane (m)=',F8.3)
      WRITE (11,70) DIHDRL
70     FORMAT ('The wing DIHeDRaL (degrees per wing) is ',F8.3)
      WRITE (11,75) PRPHIT
75     FORMAT ('Propeller height from trailing edge (m) =',F8.3)
      WRITE (11,80) PRPDIA
80     FORMAT ('Propeller diameter (m) = ',F8.3)
      WRITE (11,90) PRPRPM
90     FORMAT ('Propeller RPM (revolutions/minute) = ',F8.2)
      WRITE (11,100) SWRCOF
      WRITE (11,110)
      WRITE (11,120)
100   FORMAT ('Swirl coefficient    ',F8.5)
110   FORMAT ('= slipstream angular velocity divided by the')
120   FORMAT ('                    angular velocity of the propeller')
      WRITE (11,130) PSVSEP
130   FORMAT ('The percent span initial vortex separation=',F8.3)
      WRITE (11,140) CORCOF
140   FORMAT ('The CORCOF = core radius/span(Sprieter & Sacks,1951)',
     +       F8.5)
      WRITE (11,150) LFTCOF
150   FORMAT ('Lift coefficient = ',F8.3)
      WRITE (11,160) VTHFTE
160   FORMAT ('Vortex height from trailing edge (pos. up)m',F8.4)
      WRITE (11,161)
161   FORMAT ('VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8')
      WRITE (11,162) VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8
162   FORMAT (6F8.3)
      WRITE (11,163)
163   FORMAT ('VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8')
      WRITE (11,164) VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8
164   FORMAT (6F8.3)
      WRITE (11,170) POSIZ,POSIY
170   FORMAT ('Nozzle location w.r.t. trailing edge(z,y)',2(F8.4))
      WRITE (11,180) NOZX,NOZY,NOZZ
180   FORMAT ('Nozzle direction (NOZX,NOZY,NOZZ)',3(F8.4))
      WRITE (11,190) NZPRES
190   FORMAT ('Nozzle pressure (Pa) = ',F12.0)
      WRITE (11,200) DELT
200   FORMAT ('Initial Time increment = ',F10.7)
      WRITE (11,210) DIA
210   FORMAT ('Initial drop diameter (m) = ',E11.4)
      WRITE (11,215) DENDRP
215   FORMAT ('Drop density (kg/m**3) = ',F8.2)
      WRITE (11,230)
      WRITE (11,240) MSXWND
230   FORMAT ('Crosswind comp at designated height (m)')
240   FORMAT ('                       (pos from left wing, m/s) = ',F8.3)
      WRITE (11,250) MSXWHT
250   FORMAT ('HEIGHT at which the crosswind is measured (m)',F8.3)
      WRITE (11,252)
      WRITE (11,253) HTCOLA
252   FORMAT ('Height of collector array (final height) at the')
253   FORMAT ('  centerline where the drop hits (m) =',F8.3)
      WRITE (11,254)
      WRITE (11,255) SLCOLA
254   FORMAT ('Percentage slope of collector array, ')
255   FORMAT (' where positive is down off right wing =',F8.3)
530   WRITE (11,540)
```

```
 540     FORMAT ('            ')
        WRITE (11,543)
 543    FORMAT ('PSVSEP = percent span initial vortex separation')
         WRITE (11,544)
 544     FORMAT ('    LOCX       LOCZ         TIME ')
         WRITE (11,545) PSVSEP
 545    FORMAT (F5.1)
              DRPRUN = 1
*
 550   RETURN
        END
*      ----------------------------------------------------------------
*
*
*
*
*          ******* INDVEL ****** 6000 ***********************************
*          Velocities : induced, air, droplet, relative.
*          --------------------------------------------------------------
*
         SUBROUTINE  INDVEL(ACELX,ACELY,ACELZ,CD,CONST1,CORRAD,CRCLAT,
     +        DIA,HEIGHT,KINVSC,PRPDIA,RE,SEP,SLPANG,SWRROT,TIME,
     +        VELXVT,VELYVT,VELZVT,VLPLAN,VLXWND,
     +        VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +        VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +        VLXREL,VLYREL,VLZREL,VELREL,
     +        VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
     +        VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
     +        VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +        VX,VY,VZ,X,Y,Z)
*
         REAL  ACELX,ACELY,ACELZ  D,CONST1,CONST5,CORRAD,CRCLAT,DIA,
     +           GRAV,HEIGHT,KINVSC,PI,PRPDIA,RE,SEP,SWRROT,TIME,
     +           R1,R2,R3,R4,R5,R6,R7,R8,SLPANG,
     +           VELXVT,VELYVT,VELZVT,VLPLAN,VLXWND,
     +           VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +           VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +           VLXREL,VLYREL,VLZREL,VELREL,
     +           VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
     +           VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
     +           VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +           VX,VY,VZ,
     +           X,X1,X2,X3,X4,X5,X6,X7,X8,
     +           Y,Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,
     +           Z,Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8
*
        GRAV = 9.80665
        PI = 3.141593
*
*       ------ For vortex 1 ------ RIGHT WING VORTEX
        X1 = X - VRTLX1
        Y1 = VLPLAN * TIME - Y
        Z1 = Z - VRTLZ1
        R1 = (X1*X1 + Z1*Z1)**0.5
        IF (X1.EQ.0.AND.Z1.GE.0) THEN
            ALPHA1=PI/2.0
            GOTO 110
         ENDIF
        IF (X1.EQ.0.AND.Z1.LT.0) THEN
            ALPHA1=-PI/2.0
            GOTO 110
         ENDIF
        ALPHA1 = ATAN(Z1/X1)
        IF (X1.LT.0) ALPHA1 = ALPHA1 + PI
 110   IF (Y1.EQ.0) THEN
```

```
           BETA1=PI/2.0
          ELSE
           BETA1 = ATAN(R1/Y1)
          ENDIF
         IF (R1.EQ.0) THEN
             VELVT1=0
             GOTO 180
          ENDIF
         IF (R1.GT.CORRAD) GOTO 170
         VELVT1 = CRCLAT * (1.0 + COS(BETA1)) / (4.0* PI * CORRAD)
         VELVT1 = VELVT1 *(R1 / CORRAD)
         GOTO 180
170     VELVT1 = CRCLAT * (1.0 + COS(BETA1)) / (4.0 * PI *R1)
180     VLXVT1 = VELVT1 * SIN(ALPHA1 + PI)
         VLZVT1 = VELVT1 * COS(ALPHA1)
*
*        ------ For vortex 2 ------ LEFT WING VORTEX
         X2 = X - VRTLX2
         Y2 = Y1
         Z2 = Z - VRTLZ2
         R2 = (X2*X2 + Z2*Z2)**0.5
         IF (X2.EQ.0.AND.Z2.GE.0) THEN
             ALPHA2=PI/2.0
             GOTO 290
          ENDIF
         IF (X2.EQ.0.AND.Z2.LT.0) THEN
             ALPHA2=-PI/2.0
             GOTO 290
          ENDIF
         ALPHA2 = ATAN(Z2/X2)
          IF (X2.LT.0) ALPHA2 = ALPHA2 + PI
290     IF (Y2.EQ.0) THEN
              BETA2=PI/2.0
             ELSE
              BETA2 = ATAN(R2/Y2)
          ENDIF
         IF (R2.EQ.0) THEN
              VELVT2=0
              GOTO 360
          ENDIF
         IF (R2.GT.CORRAD) GOTO 350
         VELVT2 = CRCLAT * (1.0 + COS(BETA2)) / (4.0 * PI * CORRAD)
         VELVT2 = VELVT2 *(R2 / CORRAD)
         GOTO 360
350     VELVT2 = CRCLAT * (1.0 + COS(BETA2)) / (4.0 * PI * R2)
360     VLXVT2 = VELVT2 * SIN(ALPHA2)
         VLZVT2 = VELVT2 * COS(ALPHA2 + PI)
*
*        ------ For vortex 3 ------ RIGHT WING VORTEX IMAGE
         X3 = X - VRTLX3
         Y3 = Y1
         Z3 = Z - VRTLZ3
         R3 = SQRT(X3*X3 + Z3*Z3)
         IF (X3.EQ.0) THEN
             ALPHA3=PI/2.0
             GOTO 460
          ENDIF
         ALPHA3 = ATAN(Z3/X3)
         IF (X3.LT.0) ALPHA3 = ALPHA3 + PI
460     IF (Y3.EQ.0) THEN
             BETA3=PI/2.0
            ELSE
             BETA3 = ATAN(R3/Y3)
          ENDIF
         IF (R3.EQ.0) THEN
```

```
            VELVT3=0
            GOTO 530
          ENDIF
         IF (R3.GT.CORRAD) GOTO 520
         VELVT3 = CRCLAT * (1.0 + COS(BETA3)) / (4.0 * PI * CORRAD)
         VELVT3 = VELVT3 * (R3 / CORRAD)
         GOTO 530
 520     VELVT3 = CRCLAT * (1.0 + COS(BETA3)) / (4.0 * PI * R3)
 530     VLXVT3 = VELVT3 * SIN(ALPHA3)
         VLZVT3 = VELVT3 * COS(ALPHA3 + PI)
*
*        ------ For vortex 4 ------ LEFT WING VORTEX IMAGE
         X4 = X - VRTLX4
         Y4 = Y1
         Z4 = Z - VRTLZ4
         R4 = SQRT(X4*X4 + Z4*Z4)
         IF (X4.EQ.0) THEN
            ALPHA4=PI/2.0
            GOTO 630
          ENDIF
         ALPHA4 = ATAN(Z4/X4)
         IF (X4.LT.0) ALPHA4 = ALPHA4 + PI
 630     IF (Y4.EQ.0) THEN
            BETA4=PI/2.0
           ELSE
            BETA4 = ATAN(R4/Y4)
          ENDIF
         IF (R4.EQ.0) THEN
            VELVT4=0
            GOTO 700
          ENDIF
         IF (R4.GT.CORRAD) GOTO 690
         VELVT4 = CRCLAT * (1.0 + COS(BETA4)) / (4.0 * PI * CORRAD)
         VELVT4 = VELVT4 * (R4 / CORRAD)
         GOTO 700
 690     VELVT4 = CRCLAT * (1.0 + COS(BETA4)) / (4.0 * PI * R4)
 700     VLXVT4 = VELVT4 * SIN(ALPHA4 + PI)
         VLZVT4 = VELVT4 * COS(ALPHA4)
*
*        ------ For vortex 5 ------ BOUND VORTEX
         X5 = X
         Y5 = -Y1
         Z5 = Z - VRTLZ5
         R5 = (Y5*Y5 + Z5*Z5)**0.5
*                   note ===> SEP=INISEP/2
         IF ((SEP-X5).EQ.0) THEN
            BETA5A=PI/2.0
           ELSE
            BETA5A = ATAN(R5/(SEP - X5))
          ENDIF
         IF ((SEP-X5).LT.0) BETA5A = BETA5A+PI
         IF ((SEP+X5).EQ.0) THEN
            BETA5B=PI/2.0
           ELSE
            BETA5B = ATAN(R5/(SEP+X5))
          ENDIF
         IF ((SEP+X5).LT.0) BETA5B = BETA5B + PI
         IF (Y5.EQ.0.AND.Z5.GE.0) THEN
             ALPHA5=PI/2.0
             GOTO 850
          ENDIF
         IF (Y5.EQ.0.AND.Z5.LT.0) THEN
             ALPHA5=-PI/2.0
             GOTO 850
          ENDIF
```

```
      ALPHA5 = ATAN(Z5/Y5) + PI
 850  IF (R5.EQ.0) THEN
         VELVT5=0
         GOTO 910
       ENDIF
      IF (R5.GT.CORRAD) GOTO 900
      VELVT5 = CRCLAT * (COS(BETA5A) + COS(BETA5B))/(4.0*PI*CORRAD)
      VELVT5 = VELVT5 *(R5 / CORRAD)
      GOTO 910
 900  VELVT5 = CRCLAT * (COS(BETA5A) + COS(BETA5B)) /(4.0* PI * R5)
 910  VLYVT5 = VELVT5 * SIN(ALPHA5 + PI)
      VLZVT5 = VELVT5 * COS(ALPHA5)
*
*     ------ For vortex 6 ------ BOUND VORTEX IMAGE
      X6 = X
      Y6 = -Y1
      Z6 = Z - VRTLZ6
      R6 = SQRT(Y6*Y6 + Z6*Z6)
      IF (Y6.EQ.0.AND.Z6.GE.0) THEN
         ALPHA6=PI/2.0
         GOTO 702
       ENDIF
      IF (Y6.EQ.0.AND.Z6.LT.0) THEN
         ALPHA6=-PI/2.0
         GOTO 702
       ENDIF
      ALPHA6 = ATAN(Z6/Y6) + PI
 702  IF ((SEP-X6).EQ.0) THEN
         BETA6A=PI/2.0
        ELSE
         BETA6A = ATAN(R6/(SEP - X6))
       ENDIF
      IF ((SEP-X6).LT.0) BETA6A = BETA6A + PI
      IF ((SEP+X6).EQ.0) THEN
         BETA6B=PI/2.0
        ELSE
         BETA6B = ATAN(R6/(SEP+X6))
       ENDIF
      IF ((SEP+X6).LT.0) BETA6B = BETA6B + PI
      IF (R6.EQ.0) THEN
         VELVT6=0
         GOTO 712
       ENDIF
      IF (R6.GT.CORRAD) GOTO 711
      VELVT6 = CRCLAT * (COS(BETA6A) + COS(BETA6B))/(4.0*PI*CORRAD)
      VELVT6 = VELVT6 *(R6 / CORRAD)
      GOTO 712
 711  VELVT6 = CRCLAT * (COS(BETA6A) + COS(BETA6B)) / (4.0*PI*R6)
 712  VLYVT6 = VELVT6 * SIN(ALPHA6)
      VLZVT6 = VELVT6 * COS(ALPHA6 + PI)
*
*     ------------- Vortex to simulate the propeller
      X7 = X - VRTLX7
      Z7 = Z - VRTLZ7
      R7 = (X7*X7 + Z7*Z7)**0.5
      IF (X7.EQ.0.AND.Z7.GE.0) THEN
         ALPHA7=PI/2.0
         GOTO 722
       ENDIF
      IF (X7.EQ.0.AND.Z7.LT.0) THEN
         ALPHA7=-PI/2.0
         GOTO 722
       ENDIF
      ALPHA7 = ATAN(Z7/X7)
       IF (X7.LT.0) ALPHA7 = ALPHA7 + PI
```

```
 722      IF (R7.EQ.0) THEN
              VELVT7=0
              GOTO 728
            ENDIF
           VELVT7 = SWRROT
*                positive SWiRl ROTation =clockwise in pilot's view
           IF (R7.GT.PRPDIA) GOTO 727
           VELVT7 = VELVT7 * R7 / PRPDIA
           GOTO 728
 727      VELVT7 = VELVT7*PRPDIA/R7
 728      VLXVT7 = VELVT7*SIN(ALPHA7)
           VLZVT7 = VELVT7*COS(ALPHA7+PI)
*
*          ------------- Vortex to simulate the propeller IMAGE
           X8 = X - VRTLX8
           Z8 = Z - VRTLZ8
           R8 = (X8*X8 + Z8*Z8)**0.5
           IF (X8.EQ.0) THEN
               ALPHA8 = PI / 2
               GOTO 730
            ENDIF
           ALPHA8 = ATAN(Z8/X8)
           IF (X8.LT.0) ALPHA8 = ALPHA8 + PI
 730      VELVT8 = -SWRROT*PRPDIA/R8
           VLXVT8 = VELVT8*SIN(ALPHA8)
           VLZVT8 = VELVT8*COS(ALPHA8+PI)
*
*         -------- Velocity Sums --------
       VELXVT =VLXVT1+VLXVT2+VLXVT3+VLXVT4+VLXVT7+VLXVT8
     +       + VLXWND * COS(SLPANG)
       VELYVT =VLYVT5+VLYVT6
       VELZVT =VLZVT1+VLZVT2+VLZVT3+VLZVT4+VLZVT5+VLZVT6+VLZVT7
     +       + VLZVT8 - VLXWND * SIN(SLPANG)
*         Note that the crosswind is assumed to be parallel to the
*                 uniformly sloped ground plane
*
          VLXREL = VX - VELXVT
          VLYREL = VY - VELYVT
          VLZREL = VZ - VELZVT
          VELREL = SQRT(VLXREL*VLXREL + VLYREL*VLYREL + VLZREL*VLZREL)
*
          RE = VELREL * DIA / KINVSC
*           PRINT*, 'REYNOLDS NO. = ',RE
          CALL DRAGCF(RE,CD)
*                       Subroutine to get CD as a function of Re
*
*
*
*          ************************
*          Accelerations of droplet
*          ------------------------
          CONST5 = CONST1 * CD / DIA
          SVLXRL = 1
          SVLYRL = 1
          SVLZRL = 1
          ACELX = CONST5 * VLXREL * VLXREL * (-SIGN(SVLXRL,VLXREL))
          ACELY = CONST5 * VLYREL * VLYREL * (-SIGN(SVLYRL,VLYREL))
          ACELZ = (CONST5*VLZREL*VLZREL*(-SIGN(SVLZRL,VLZREL)))-GRAV
*
          RETURN
          END
*         ------------------------------------------------------------
*
*
*         ******* NZVELS ******** 8000 *********************************
```

```
*          Subroutine for nozzle velocities in coordinate directions
*          -------------------------------------------------------------
*
           SUBROUTINE NZVELS(DENDRP,NOZX,NOZY,NOZZ,
     +           NZPRES,NZVELX,NZVELY,NZVELZ)
*
           REAL  DENDRP,NOZLEN,NOZVEL,NOZX,NOZY,NOZZ,NZANGX,
     +           NZANGY,NZANGZ,NZCOEF,NZPRES,NZVELX,NZVELY,NZVELZ
*
           NZCOEF  =  0.8
*          NZCOEF  =  0.8     is the recommended value by Goering 1972
           NOZVEL  =  (2.0 * NZCOEF * NZPRES / DENDRP)**0.5
           NOZLEN  =  (NOZX*NOZX + NOZY*NOZY + NOZZ*NOZZ)**0.5
*              nozzle angle direction cosines
           NZANGX  =  NOZX / NOZLEN
           NZANGY  =  NOZY / NOZLEN
           NZANGZ  =  NOZZ / NOZLEN
           NZVELX  =  NZANGX * NOZVEL
           NZVELY  =  NZANGY * NOZVEL
           NZVELZ  =  NZANGZ * NOZVEL
           RETURN
           END
*
*          ------------------------------------------------------------
*
*
*
*             ****** RUNKUT *** 10000 **************************************
*             Runge-Kutta Fourth Order Algorithm
*             -------------------------------------------------------------
*
          SUBROUTINE RUNKUT(ACELX,ACELY,ACELZ,CD,CFXWND,CHK,CONST1,
     +      CORRAD,CRCLAT,DELT,DIA,FHIGHT,
     +      HEIGHT,HTCOLA,INIDIA,KINVSC,LOCX,LOCY,LOCZ,
     +      PRPDIA,RE,RELAST,SEP,SLCOLA,SLPANG,SRFHIT,SWRROT,TIME,
     +      VELREL,VELX,VELY,VELZ,VELXVT,VELYVT,VELZVT,VLXWND,VLPLAN,
     +      VLXREL,VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLXVT7,VLXVT8,
     +      VLYREL,VLYVT5,VLYVT6,VLZREL,
     +      VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +      VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,VRTLY1,VRTLY2,
     +      VRTLY3,VRTLY4,VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8,
     +      VX,VY,VZ,X,Y,Z)
*
           INTEGER CHK
           REAL  ACELX,ACELY,ACELZ,CD,CFXWND,CONST1,CORRAD,CRCLAT,
     +           DELT,DIA,
     +           FHIGHT,HEIGHT,HTCOLA,INIDIA,INTCOF,KINVSC,
     +           K11X,K11Y,K11Z,K12X,K12Y,K12Z,K21X,K21Y,K21Z,
     +           K22X,K22Y,K22Z,K31X,K31Y,K31Z,K32X,K32Y,K32Z,
     +           K41X,K41Y,K41Z,K42X,K42Y,K42Z,K1X,K1Y,K1Z,K2X,K2Y,K2Z,
     +           LOCX,LOCY,LOCZ,LSLOCX,LSLOCY,LSLOCZ,LSTIME,PI,PRPDIA,
     +           RE,RELAST,SEP,SLCOLA,SLPANG,SLPGND,SRFHIT,SWRROT,TIME,
     +           VELREL,VELX,VELY,VELZ,VELXVT,VELYVT,VELZVT,VLXWND,VLPLAN,
     +           VLXREL,VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLXVT7,VLXVT8,
     +           VLYREL,VLYVT5,VLYVT6,VLZREL,
     +           VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +           VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +           VRTLY1,VRTLY2,VRTLY3,VRTLY4,
     +           VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8,
     +           VX,VY,VZ,X,Y,Z
*
             PI  =  3.141593
*
*            ----- step 1
             TIME  =  TIME
```

```
      X  =  LOCX
      Y  =  LOCY
      Z  =  LOCZ
      VX  =  VELX
      VY  =  VELY
      VZ  =  VELZ
*
   CALL  INDVEL(ACELX,ACELY,ACELZ,CD,CONST1,CORRAD,CRCLAT,
 +        DIA,HEIGHT,KINVSC,PRPDIA,RE,SEP,SLPANG,SWRROT,TIME,
 +        VELXVT,VELYVT,VELZVT,VLPLAN,VLXWND,
 +        VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
 +        VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
 +        VLXREL,VLYREL,VLZREL,VELREL,
 +        VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
 +        VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
 +        VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
 +        VX,VY,VZ,X,Y,Z)
*                vortex, air and relative velocities and accelerations
      IF (RE.GT.(1.5*RELAST)) THEN
          DELT  =  DELT/2.0
          CHK  =  CHK-1
      ENDIF
        RELAST  =  RE
      K11X  =  DELT * VX
      K11Y  =  DELT * VY
      K11Z  =  DELT * VZ
      K12X  =  DELT * ACELX
      K12Y  =  DELT * ACELY
      K12Z  =  DELT * ACELZ
*
*         ----- step 2
      TIME  =  TIME + DELT/2
      X  =  LOCX  +  K11X/2.0
      Y  =  LOCY  +  K11Y/2.0
      Z  =  LOCZ  +  K11Z/2.0
      VX  =  VELX  +  K12X/2.0
      VY  =  VELY  +  K12Y/2.0
      VZ  =  VELZ  +  K12Z/2.0
*
   CALL  INDVEL(ACELX,ACELY,ACELZ,CD,CONST1,CORRAD,CRCLAT,
 +        DIA,HEIGHT,KINVSC,PRPDIA,RE,SEP,SLPANG,SWRROT,TIME,
 +        VELXVT,VELYVT,VELZVT,VLPLAN,VLXWND,
 +        VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
 +        VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
 +        VLXREL,VLYREL,VLZREL,VELREL,
 +        VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
 +        VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
 +        VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
 +        VX,VY,VZ,X,Y,Z)
*                vortex, air and relative velocities and accelerations
      K21X  =  DELT * VX
      K21Y  =  DELT * VY
      K21Z  =  DELT * VZ
      K22X  =  DELT * ACELX
      K22Y  =  DELT * ACELY
      K22Z  =  DELT * ACELZ
*
*         ----- step 3
      TIME  =  TIME
      X  =  LOCX  +  K21X/2.0
      Y  =  LOCY  +  K21Y/2.0
      Z  =  LOCZ  +  K21Z/2.0
      VX  =  VELX  +  K22X/2.0
      VY  =  VELY  +  K22Y/2.0
      VZ  =  VELZ  +  K22Z/2.0
```

```
*
      CALL  INDVEL(ACELX,ACELY,ACELZ,CD,CONST1,CORRAD,CRCLAT,
     +      DIA,HEIGHT,KINVSC,PRPDIA,RE,SEP,SLPANG,SWRROT,TIME,
     +      VELXVT,VELYVT,VELZVT,VLPLAN,VLXWND,
     +      VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +      VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +      VLXREL,VLYREL,VLZREL,VELREL,
     +      VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
     +      VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
     +      VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +      VX,VY,VZ,X,Y,Z)
*                  vortex, air and relative velocities and accelerations
        K31X  =  DELT  *  VX
        K31Y  =  DELT  *  VY
        K31Z  =  DELT  *  VZ
        K32X  =  DELT  *  ACELX
        K32Y  =  DELT  *  ACELY
        K32Z  =  DELT  *  ACELZ
*
*         -----  step  4
        TIME  =  TIME  +  DELT/2.0
        X  =  LOCX  +  K31X
        Y  =  LOCY  +  K31Y
        Z  =  LOCZ  +  K31Z
        VX  =  VELX  +  K32X
        VY  =  VELY  +  K32Y
        VZ  =  VELZ  +  K32Z
*
      CALL  INDVEL(ACELX,ACELY,ACELZ,CD,CONST1,CORRAD,CRCLAT,
     +      DIA,HEIGHT,KINVSC,PRPDIA,RE,SEP,SLPANG,SWRROT,TIME,
     +      VELXVT,VELYVT,VELZVT,VLPLAN,VLXWND,
     +      VLXVT1,VLXVT2,VLXVT3,VLXVT4,VLYVT5,VLYVT6,VLXVT7,VLXVT8,
     +      VLZVT1,VLZVT2,VLZVT3,VLZVT4,VLZVT5,VLZVT6,VLZVT7,VLZVT8,
     +      VLXREL,VLYREL,VLZREL,VELREL,
     +      VRTLX1,VRTLY1,VRTLZ1,VRTLX2,VRTLY2,VRTLZ2,
     +      VRTLX3,VRTLY3,VRTLZ3,VRTLX4,VRTLY4,VRTLZ4,
     +      VRTLX7,VRTLZ7,VRTLX8,VRTLZ8,
     +      VX,VY,VZ,X,Y,Z)
*                  vortex, air and relative velocities and accelerations
        K41X  =  DELT  *  VX
        K41Y  =  DELT  *  VY
        K41Z  =  DELT  *  VZ
        K42X  =  DELT  *  ACELX
        K42Y  =  DELT  *  ACELY
        K42Z  =  DELT  *  ACELZ
*
*         -----  step  5
        K1X  =  (K11X  +  2.0*K21X  +  2.0*K31X  +  K41X)/6.0
        K1Y  =  (K11Y  +  2.0*K21Y  +  2.0*K31Y  +  K41Y)/6.0
        K1Z  =  (K11Z  +  2.0*K21Z  +  2.0*K31Z  +  K41Z)/6.0
        K2X  =  (K12X  +  2.0*K22X  +  2.0*K32X  +  K42X)/6.0
        K2Y  =  (K12Y  +  2.0*K22Y  +  2.0*K32Y  +  K42Y)/6.0
        K2Z  =  (K12Z  +  2.0*K22Z  +  2.0*K32Z  +  K42Z)/6.0
*
*         -----  step  6
        TIME  =  TIME
        LOCX  =  LOCX  +  K1X
        LOCY  =  LOCY  +  K1Y
        LOCZ  =  LOCZ  +  K1Z
        VELX  =  VELX  +  K2X
        VELY  =  VELY  +  K2Y
        VELZ  =  VELZ  +  K2Z
*
        FHIGHT  =  HTCOLA  -  (LOCX  *  SLCOLA/100)
       IF  (LOCZ.GT.FHIGHT)  GOTO  990
```

```
              INTCOF  =  (FHIGHT-LSLOCZ)/(LOCZ-LSLOCZ)
              LOCX  =  (LOCX-LSLOCX)  *  INTCOF   +    LSLOCX
              LOCY  =  (LOCY-LSLOCY)  *  INTCOF   +    LSLOCY
              LOCZ  =  FHIGHT
              TIME  =  (TIME-LSTIME)  *  INTCOF   +    LSTIME
*
  990       IF (LOCZ.LE.FHIGHT) GOTO 185
*
          CALL VRTCOR(CRCLAT,DELT,PRPDIA,SLPANG,SWRROT,VLXWND,
     +      VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +      VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8)
*                         Subroutine to get new VoRTex core LOCations
*
          VLXWND = CFXWND * LOG(LOCZ/SRFHIT)
*
 185        LSLOCX=LOCX
             LSLOCY=LOCY
             LSLOCZ=LOCZ
             LSTIME=TIME
*
 400      RETURN
           END
*
*          ------------------------------------------------------------
*
*
*          ******* VRTCOR ********* 11400 ********************************
*          Subroutine to calculate VoRTex CORE  LOCations and velocities
*          --------------------------------------------------------------
*
        SUBROUTINE VRTCOR(CRCLAT,DELT,PRPDIA,SLPANG,SWRROT,VLXWND,
     +      VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +      VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8)
*
         INTEGER N
         REAL VX(8),VZ(8),VR1(8),VR2(8),VR3(8),VR4(8),VR7(8),VR8(8),
     +    VRVLX(8),VRVLZ(8),
     +    VVLX1(8),VVLX2(8),VVLX3(8),VVLX4(8),VVLX7(8),VVLX8(8),
     +    VVLZ1(8),VVLZ2(8),VVLZ3(8),VVLZ4(8),VVLZ7(8),VVLZ8(8)
         REAL CRCLAT,CRCPI,DELT,PI,SLPANG,VLXWND,
     +    VRTLX1,VRTLX2,VRTLX3,VRTLX4,VRTLX7,VRTLX8,
     +    VRTLZ1,VRTLZ2,VRTLZ3,VRTLZ4,VRTLZ7,VRTLZ8
*
          PI  =  3.141593
            VX(1)  =  VRTLX1
            VX(2)  =  VRTLX2
            VX(3)  =  VRTLX3
            VX(4)  =  VRTLX4
            VX(7)  =  VRTLX7
            VX(8)  =  VRTLX8
            VZ(1)  =  VRTLZ1
            VZ(2)  =  VRTLZ2
            VZ(3)  =  VRTLZ3
            VZ(4)  =  VRTLZ4
            VZ(7)  =  VRTLZ7
            VZ(8)  =  VRTLZ8
*
*
*               velocities for potential flow and sloping ground
*                    induced velocity at core #N center location
         DO 10 N=1,8
             IF (N.EQ.5.OR.N.EQ.6) GOTO 10
             CRCPI = CRCLAT / (2.0*PI)
*                 distances between vortex cores
            VR1(N)= (VX(N) - VX(1))**2 + (VZ(N) - VZ(1))**2
```

```
          VR2(N)=  (VX(N)  -  VX(2))**2  +  (VZ(N)  -  VZ(2))**2
          VR3(N)=  (VX(N)  -  VX(3))**2  +  (VZ(N)  -  VZ(3))**2
          VR4(N)=  (VX(N)  -  VX(4))**2  +  (VZ(N)  -  VZ(4))**2
          VR7(N)=  (VX(N)  -  VX(7))**2  +  (VZ(N)  -  VZ(7))**2
          VR8(N)=  (VX(N)  -  VX(8))**2  +  (VZ(N)  -  VZ(8))**2
*            X  and  Z  induced  velocities  at  core  locations
         IF  (N.EQ.1)  GOTO  2
          VVLX1(N)  =  -CRCPI  *  (VZ(N)  -  VZ(1))  /  VR1(N)
          VVLZ1(N)  =  CRCPI  *  (VX(N)  -  VX(1))  /  VR1(N)
  2      IF  (N.EQ.2)  GOTO  3
          VVLX2(N)  =  CRCPI  *  (VZ(N)  -  VZ(2))  /  VR2(N)
          VVLZ2(N)  =  -CRCPI  *  (VX(N)  -  VX(2))  /  VR2(N)
  3      IF  (N.EQ.3)  GOTO  4
          VVLX3(N)  =  CRCPI  *  (VZ(N)  -  VZ(3))  /  VR3(N)
          VVLZ3(N)  =  -CRCPI  *  (VX(N)  -  VX(3))  /  VR3(N)
  4      IF  (N.EQ.4)  GOTO  7
          VVLX4(N)  =  -CRCPI  *  (VZ(N)  -  VZ(4))  /  VR4(N)
          VVLZ4(N)  =  CRCPI  *  (VX(N)  -  VX(4))  /  VR4(N)
  7      IF  (N.EQ.7)  GOTO  8
          VVLX7(N)  =  (SWRROT*PRPDIA)  *  (VZ(N)  -  VZ(7))  /  VR7(N)
          VVLZ7(N)  =  -(SWRROT*PRPDIA)  *  (VX(N)  -  VX(7))  /  VR7(N)
  8      IF  (N.EQ.8)  GOTO  9
          VVLX8(N)  =  -(SWRROT*PRPDIA)  *  (VZ(N)  -  VZ(8))  /  VR8(N)
          VVLZ8(N)  =  (SWRROT*PRPDIA)  *  (VX(N)  -  VX(8))  /  VR8(N)
*            X  and  Z  induced  velocity  sums  at  core  location  N
  9       VRVLX(N)  =  VVLX1(N)  +  VVLX2(N)  +  VVLX3(N)  +  VVLX4(N)  +
     +          VVLX7(N)  +  VVLX8(N)  +  VLXWND  *  COS(SLPANG)
          VRVLZ(N)  =  VVLZ1(N)  +  VVLZ2(N)  +  VVLZ3(N)  +  VVLZ4(N)  +
     +          VVLZ7(N)  +  VVLZ8(N)  -  VLXWND  *  SIN(SLPANG)
*        Note  that  the  crosswind  is  assumed  to  be  parallel  to  the
*                 uniformly  sloped  ground  plane
 10     CONTINUE
*             vortex  core  locations
        DO  20  N=1,8
           IF  (N.EQ.5.OR.N.EQ.6)  GOTO  20
          VX(N)  =  VX(N)  +  VRVLX(N)  *  DELT
          VZ(N)  =  VZ(N)  +  VRVLZ(N)  *  DELT
 20     CONTINUE
         VRTLX1  =  VX(1)
         VRTLX2  =  VX(2)
         VRTLX3  =  VX(3)
         VRTLX4  =  VX(4)
         VRTLX7  =  VX(7)
         VRTLX8  =  VX(8)
         VRTLZ1  =  VZ(1)
         VRTLZ2  =  VZ(2)
         VRTLZ3  =  VZ(3)
         VRTLZ4  =  VZ(4)
         VRTLZ7  =  VZ(7)
         VRTLZ8  =  VZ(8)
*
         RETURN
         END
*
*        ----------------------------------------------------
*
*
*         ******  DRAGCF  *****  11700  *************************************
*         Subroutine  to  define  CD  in  terms  of  Reynolds  number  (RE)
*         ----------------------------------------------------------------
*
         SUBROUTINE  DRAGCF(RE,CD)
*
         REAL  RE,CD,CDBP,CDLB,CDSTOK
*
```

```
      CDSTOK = 24.0 / RE
*               drag coefficient of droplet according to Stoke's law
      IF (RE.LT.(0.01)) THEN
           CD = CDSTOK
           RETURN
        ENDIF
*    (Beard & Pruppacher (1969) p1069-1070;Mason(1971);Pruppacher(1970))
      IF (RE.LT.2) THEN
            CD = CDSTOK * (1.0 +0.102 * RE **0.955)
            RETURN
        ENDIF
      IF (RE.LT.21) THEN
            CD = CDSTOK * (1.0 +0.115 * RE **0.802)
            RETURN
        ENDIF
      IF (RE.LT.200) THEN
            CD = CDSTOK * (1.0 +0.189 * RE **0.632)
            RETURN
        ENDIF
*            Now is a linear interpolation between the theory of
*           Beard and Pruppacher, that of Lamguir and Blogdett
      IF (RE.LT.400) THEN
          CDBP = CDSTOK * (1.0 +0.189 * RE **0.632)
          CDLB = CDSTOK * (1.0 +0.197 * RE **0.63+0.00026*RE **1.38)
          CD = CDBP + ((RE-200.0)/200.0) * (CDLB - CDBP)
          RETURN
        ENDIF
      IF (RE.GT.50000) THEN
           CD =0.5
           RETURN
        ENDIF
*                 now for (400.LT.RE and RE.LT.50000)
      CD = CDSTOK * (1.0 +0.197 * RE **0.63 +0.00026 * RE ** 1.38)
*                              (from Langmuir and Blodgett, 1946)
      RETURN
      END
*     ----------------------------------------------------------------
*
*
*     ********   CHDELT   *******   13000   ****************************
*      Subroutine to change size of DELT  to reduce calculation time
*     ----------------------------------------------------------------
*
       SUBROUTINE CHDELT(CHK,DELT,DIA,RE)
*
       INTEGER CHK
       REAL DIA,RE,DELT
*
       IF (DIA.GT.(0.0003)) GOTO 30
       IF (DIA.LT.(0.0001)) GOTO 10
        IF (RE.LT.1000.AND.CHK.LT.1) THEN
             DELT=DELT*2.0
             CHK=CHK+1
             GOTO 100
         ENDIF
        IF (RE.LT.500.AND.CHK.LT.2) THEN
             DELT=DELT*2.0
             CHK=CHK+1
             GOTO 100
         ENDIF
        IF (RE.LT.300.AND.CHK.LT.3) THEN
             DELT=DELT*2.0
             CHK=CHK+1
             GOTO 100
         ENDIF
```

```
            IF (RE.LT.200.AND.CHK.LT.4) THEN
                 DELT=DELT*2.0
                 CHK=CHK+1
                 GOTO 100
             ENDIF
            IF (RE.LT.100.AND.CHK.LT.5) THEN
                 DELT=DELT*2.0
                 CHK=CHK+1
                 GOTO 100
             ENDIF
            IF (RE.LT.50.AND.CHK.LT.6) THEN
                 DELT=DELT*2.0
                 CHK=CHK+1
                 GOTO 100
             ENDIF
*           IF (RE.LT.30.AND.CHK.LT.7) THEN
*                DELT=DELT*2.0
*                CHK=CHK+1
*                GOTO 100
*            ENDIF
          GOTO 100
*
   10    IF (DIA.LT.(0.00003)) GOTO 20
            IF (RE.LT.200.AND.CHK.LT.1) THEN
                 DELT=DELT*2.0
                 CHK=CHK+1
                 GOTO 100
             ENDIF
            IF (RE.LT.100.AND.CHK.LT.2) THEN
                 DELT=DELT*2.0
                 CHK=CHK+1
                 GOTO 100
             ENDIF
            IF (RE.LT.30.AND.CHK.LT.3) THEN
                 DELT=DELT*2.0
                 CHK=CHK+1
                 GOTO 100
             ENDIF
            IF (RE.LT.10.AND.CHK.LT.4) THEN
                 DELT=DELT*2.0
                 CHK=CHK+1
                 GOTO 100
             ENDIF
            IF (RE.LT.4.AND.CHK.LT.5) THEN
                 DELT=DELT*2.0
                 CHK=CHK+1
                 GOTO 100
             ENDIF
            IF (RE.LT.1.AND.CHK.LT.6) THEN
                 DELT=DELT*2.0
                 CHK=CHK+1
                 GOTO 100
             ENDIF
            IF (RE.LT.(0.5).AND.CHK.LT.7) THEN
                 DELT=DELT*2.0
                 CHK=CHK+1
                 GOTO 100
             ENDIF
          GOTO 100
*
   20     IF (DIA.LT.(0.00001)) GOTO 40
            IF (RE.LT.5.AND.CHK.LT.1) THEN
                 DELT=DELT*2.0
                 CHK=CHK+1
                 GOTO 100
```

```
          ENDIF
         IF  (RE.LT.1.AND.CHK.LT.2)  THEN
               DELT=DELT*2.0
               CHK=CHK+1
               GOTO  100
          ENDIF
         IF  (RE.LT.(0.5).AND.CHK.LT.3)  THEN
               DELT=DELT*2.0
               CHK=CHK+1
               GOTO  100
          ENDIF
         IF  (RE.LT.(0.1).AND.CHK.LT.4)  THEN
               DELT=DELT*2.0
               CHK=CHK+1
               GOTO  100
          ENDIF
         IF  (RE.LT.(0.05).AND.CHK.LT.5)  THEN
               DELT=DELT*2.0
               CHK=CHK+1
               GOTO  100
          ENDIF
         IF  (RE.LT.(0.025).AND.CHK.LT.6)  THEN
               DELT=DELT*2.0
               CHK=CHK+1
               GOTO  100
          ENDIF
         IF  (RE.LT.(0.02).AND.CHK.LT.7)  THEN
               DELT=DELT*2.0
               CHK=CHK+1
               GOTO  100
          ENDIF
         IF  (RE.LT.(0.019).AND.CHK.LT.8)  THEN
               DELT=DELT*5.0
               CHK=CHK+1
               GOTO  100
          ENDIF
      GOTO  100
*
30     IF  (RE.LT.5000.AND.CHK.LT.1)  THEN
               DELT=DELT*2.0
               CHK=CHK+1
               GOTO  100
          ENDIF
         IF  (RE.LT.4000.AND.CHK.LT.2)  THEN
               DELT=DELT*2.0
               CHK=CHK+1
               GOTO  100
          ENDIF
         IF  (RE.LT.3000.AND.CHK.LT.3)  THEN
               DELT=DELT*2.0
               CHK=CHK+1
               GOTO  100
          ENDIF
         IF  (RE.LT.2000.AND.CHK.LT.4)  THEN
               DELT=DELT*2.0
               CHK=CHK+1
               GOTO  100
          ENDIF
         IF  (RE.LT.1000.AND.CHK.LT.5)  THEN
               DELT=DELT*2.0
               CHK=CHK+1
               GOTO  100
          ENDIF
         IF  (RE.LT.500.AND.CHK.LT.6)  THEN
               DELT=DELT*2.0
```

```
                  CHK=CHK+1
                  GOTO 100
               ENDIF
            GOTO 100
*
   40       PRINT*, '******  Diameter is less than 10 microns!  ******'
*
  100       RETURN
             END
```

11. APPENDIX III. PARTICLE DEPOSITION DATA AND RELEVANT FLIGHT TEST DATA

The data for the particle deposition along with relevant information pertaining to the flight test conditions, from Morris *et al.* (1984), is given in this section. The file NASARES contains the final lateral location of the droplets as found by Morris *et al.* (1984), shown in meters from the centerline. The first 68 points represent the data for the right wing, while the last 68 represent the data for the left wing. As noted earlier only one location is given for each pass of the aircraft, this being the average of the collector arrays over which the aircraft passed at the same height. Also, for the locations for which the deposition could not be determined with sufficient accurately, the value of exactly 0 is entered in the data file.

Second is given a listing of the file NASADATA, which as explained in Appendix II, contains the flight test information which changes for each pass of the aircraft. Again the first 68 points are for the right wing and the last 68 are for the left wing. It may be noted that they are the same except that the sign of the crosswind has been changed for the left wing.

NASARES

3.05
5.06

8.20
9.09
7.74
-.42
0.07
2.89
7.99
6.57
8.22
8.99
6.99
6.51
8.61
7.05
7.51
8.45
6.47
5.57
6.17
7.49
8.70
0
0
0
0
4.46
3.62
2.03
3.50
5.79
15.13
14.59
17.04
16.21
15.29
11.80
7.78
5.08
1.90
3.63
5.08
8.09
14.98
16.09
3.06
6.79
6.98
-.52
2.96
0.78
5.90
0
4.31
0.66
0
0
0
8.76
8.31
8.46
5.71
5.28
8.15
7.68
1.99

2.33
-9.04
0
-11.86
-9.32
-12.51
0
-3.18
-2.16
-11.52
-14.25
-12.30
-34.70
-11.56
-15.32
-10.93
-13.83
-14.11
-13.71
-16.64
-16.62
-16.87
-15.83
-14.15
-11.90
0
0
0
-2.85
-12.64
-12.40
-11.34
-13.20
-4.93
-8.34
-9.42
-5.49
-5.87
-10.28
-14.75
-14.42
-20.89
-14.26
-17.82
-8.20
-4.04
-4.64
-14.28
-7.50
-9.80
-15.61
-13.23
-16.37
--1.66
0
-0.87
-1.95
0
-8.97
0
-11.93
-14.16
-15.40
-14.96
-13.66

-12.32
-16.48
-6.10
-8.86

NASADATA

50	5851	113.2	14	-2.23	650
50	5794	86.9	14	-3.38	650
75	6166	113	16	-1.64	650
75	6118	111.3	11	-.34	650
75	6034	71.9	14	-1.67	650
25	5998	118.7	9	-4.46	650
25	5965	120.8	9	-2.17	650
25	6029	90.9	13	1.37	650
80	6247	110.3	11	-3.28	650
80	6176	113.3	12	-5.07	650
80	6235	90.9	12	-3.25	650
80	6188	93.2	12	-2.12	650
80	6147	118.1	11	-3.04	650
80	6106	114.9	12	-6.79	650
85	6178	113.6	13	-6.20	650
85	6229	117.7	13	-6.23	650
85	6176	91.2	12	-6.76	650
85	6129	88.7	11	-4.76	650
90	6235	112.3	15	-6.44	650
90	6165	114.4	14	-6.52	650
90	6079	87.3	11	-5.48	650
90	6026	88.0	10	-5.69	650
95	6115	118.6	15	-2.72	650
95	6019	86.7	14	-2.07	650
15	6273	117.2	10	3.08	650
15	6099	89.7	8	4.91	650
15	6050	89.7	11	1.01	650
40	6212	118.0	15	2.81	650
60	6148	116.7	13	-5.25	650
50	6235	123.3	14	-5.84	327
50	6094	88.5	12	-5.54	327
50	5956	89.1	12	-5.22	327
75	6154	115.7	12	3.85	327
75	6115	117.9	13	2.62	327
75	6079	86.3	13	2.20	327
80	6181	115.7	12	2.30	327
80	6139	117.4	11	2.47	327
80	6043	85.2	4	-2.89	327
80	6007	82.9	13	-2.36	327
85	6178	115.1	14	-3.87	327
85	6142	150.3	12	-6.04	327
85	6070	86.6	13	-5.50	327
85	6034	87.9	12	-4.53	327
90	6157	124.7	17	1.97	327
90	6009	88.7	8	3.35	327
90	5950	87.7	9	5.87	327
90	6100	118.3	15	-2.81	327
90	6028	86.5	10	-.62	327
90	5998	85.6	9	-2.43	327
95	6148	150.5	18	-5.18	327
95	6112	119.3	16	-3.73	327

95	6070	89.4	13	-3.90	327
25	6218	120.3	21	5.05	327
25	6182	118.0	19	2.01	327
25	6088	83.7	10	5.06	327
15	6016	119.1	14	2.07	327
15	5902	121.7	17	-4.72	327
15	5878	89.2	13	-4.72	327
15	5800	85.1	11	-5.02	327
70	6176	119.8	15	-4.04	327
70	5982	87.3	11	-2.65	327
70	6117	88.2	12	-4.10	327
60	5929	116.4	14	-3.00	327
60	6206	110.5	16	-2.45	327
60	6173	82.2	14	-.11	327
60	6147	79.1	13	-3.54	327
40	6150	113.5	13	-3.00	327
40	6094	80.5	10	-3.40	327
50	5851	113.2	14	2.23	650
50	5794	86.9	14	3.38	650
75	6166	113	16	1.64	650
75	6118	111.3	11	.34	650
75	6034	71.9	14	1.67	650
25	5998	118.7	9	4.46	650
25	5965	120.8	9	2.17	650
25	6029	90.9	13	-1.37	650
80	6247	110.3	11	3.28	650
80	6176	113.3	12	5.07	650
80	6235	90.9	12	3.25	650
80	6188	93.2	12	2.12	650
80	6147	118.1	11	3.04	650
80	6106	114.9	12	6.79	650
85	6178	113.6	13	6.20	650
85	6229	117.7	13	6.23	650
85	6176	91.2	12	6.76	650
85	6129	88.7	11	4.76	650
90	6235	112.3	15	6.44	650
90	6165	114.4	14	6.52	650
90	6079	87.3	11	5.48	650
90	6026	88.0	10	5.69	650
95	6115	118.6	15	2.72	650
95	6019	86.7	14	2.07	650
15	6273	117.2	10	-3.08	650
15	6099	89.7	8	-4.91	650
15	6050	89.7	11	-1.01	650
40	6212	118.0	15	-2.81	650
60	6148	116.7	13	5.25	650
50	6235	123.3	14	5.84	327
50	6094	88.5	12	5.54	327
50	5956	89.1	12	5.22	327
75	6154	115.7	12	-3.85	327
75	6115	117.9	13	-2.62	327
75	6079	86.3	13	-2.20	327
80	6181	115.7	12	-2.30	327
80	6139	117.4	11	-2.47	327
80	6043	85.2	4	2.89	327
80	6007	82.9	13	2.36	327
85	6178	115.1	14	3.87	327
85	6142	150.3	12	6.04	327
85	6070	86.6	13	5.50	327
85	6034	87.9	12	4.53	327
90	6157	124.7	17	-1.97	327
90	6009	88.7	8	-3.35	327
90	5950	87.7	9	-5.87	327
90	6100	118.3	15	2.81	327
90	6028	86.5	10	.62	327

90	5998	85.6	9	2.43	327
95	6148	150.5	18	5.18	327
95	6112	119.3	16	3.73	327
95	6070	89.4	13	3.90	327
25	6218	120.3	21	-5.05	327
25	6182	118.0	19	-2.01	327
25	6088	83.7	10	-5.06	327
15	6016	119.1	14	-2.07	327
15	5902	121.7	17	4.72	327
15	5878	89.2	13	4.72	327
15	5800	85.1	11	5.02	327
70	6176	119.8	15	4.04	327
70	5982	87.3	11	2.65	327
70	6117	88.2	12	4.10	327
60	5929	116.4	14	3.00	327
60	6206	110.5	16	2.45	327
60	6173	82.2	14	.11	327
60	6147	79.1	13	3.54	327
40	6150	113.5	13	3.00	327
40	6094	80.5	10	3.40	327

12. APPENDIX IV. DROPTIME PROGRAM LISTING

The following program is used to determine the diameter of an evaporating water droplet. The model for evaporation, terminal velocity, and drag coefficient is the same as in WAKE77 and NASA77. The user is prompted for the initial diameter of the drop, the time increment, and the wet bulb depression. The output file contains the size of the droplet, in microns, for the life of the droplet. The output file name is DROPXXXX.TIM, where XXXX is the size of the droplet in microns. If the drop diameter is less than 1000, then the last X is omitted; if less than 100, then the last two XX's are omitted; and if less than 10, then the last three XXX's are omitted.

```
180 '**************************************************************
190 '  DROPTIME --- PROGRAM WRITTEN BY CRAIG S. MERKL        12-3-87
200 ' This program calculates the drop life, then calculates the
205 '     diameter of the drop at a particular wet bulb depression
206 '     until the drop has evaporated completely.
210 ' Language:  Advanced Basic version 2.0
220 '     Note:  The drag coefficient is calculated by averaging that
225 '              predicted by Lamguir and Blodgett (1946)
226 '                   and Beard and Pruppacher (1969)
230 '    Input: The user is prompted for the required input.
250 '**************************************************************
260 '    Main Program
[illegible] ' ---------------
280 '
290 INPUT "The starting drop diameter in microns=";DIA.MIC
295    DIA=DIA.MIC/1000000!
300   DIA$=STR$(DIA.MIC) : STRLEN%=LEN(DIA$) :STRLN%=STRLEN%-1
303      INIT.DIA=DIA
305      DIA$=RIGHT$(DIA$,STRLN%) : FILE$="DROP"+DIA$ + ".TIM"
311   OPEN FILE$ FOR OUTPUT AS #1
330    GOSUB 1000                      'Constants and Coefficients
346   LPRINT DATE$,TIME$,FILE$
347   LPRINT "      DENAIR (kg/m^3) =";DENAIR;"      DYVISC = ";DYVISC
349 INPUT "TIME INCREMENT TO USE (SECONDS) = ";TIME.INCREMENT
350 INPUT "WET BULB DEPRESSION (degrees C) =  ";WET.BULB.DEP
351   LPRINT "Initial diameter (microns)= ";DIA.MIC;"
352   LPRINT "Wet Bulb Depression= ";WET.BULB.DEP
353   GOSUB 2000 : LPRINT "The DROP.LIFE (sec) = ";DROP.LIFE
354  FOR TIME = 0 TO 1000 STEP TIME.INCREMENT
355   GOSUB 2300
356   DIA.MIC=DIA*1000000!
357  LPRINT USING"####.## ";DIA.MIC,TIME
358   PRINT #1,USING"####.### ";DIA.MIC,TIME
359  NEXT TIME
360  PRINT " WELL SO MUCH FOR THAT DROP !" : CLOSE  : END
369   '*********************************************************
370   '
380   ' Subroutine to define CD in terms of Reynolds number (RE)
390   '  --------------------------------------------------------
400    CD.STOKE = 24 / RE   'drag coefficient according to Stoke's law
410    IF (RE < .01) THEN CD = CD.STOKE : RETURN
420         Beard & Pruppacher (1969) p1069-1070
425    '    Mason(1971)    and Pruppacher(1970)
430  IF (RE < 2) THEN CD = CD.STOKE * (1 + .102 * RE ^ .955) : RETURN
433  IF (RE < 21) THEN CD = CD.STOKE * (1 + .115 * RE ^ .802) : RETURN
435  IF (RE < 200) THEN CD = CD.STOKE * (1 + .189 * RE ^ .632) :RETURN
440  IF (RE > 50000!) THEN CD = .5 : RETURN
445  IF (RE < 400) THEN GOTO 470
450  CD = CD.STOKE * (1 + .197 * RE ^ .63 + .00026 * RE ^ 1.38) :RETURN
455   '      (from Lamguir and Blogdett, 1946)
460   ' Note: 470 is a linear interpolation between the theory of Beard
465   '  and Pruppacher (1969), and that of Lamguir and Blogdett (1946)
470    CD.B.P = CD.STOKE * (1 + .189 * RE ^ .632)
495     CD.L.B = CD.STOKE * (1 + .197 * RE ^ .63 + .00026 * RE ^ 1.38)
500     CD = CD.B.P + (( RE-200)/200) * (CD.L.B - CD.B.P)
510   RETURN
```

```
999   '**************************
1000  ' Constants and Coefficients
1020  ' -------------------------
1030  '
1050    DENAIR = 1.2256         ' density of air, kg/m^3
1060    DENDROP = 1000          ' density of droplet, kg/m^3
1070    PI = 3.141593
1080    M = M                   ' mass of droplet, kg
1090    GRAV = 9.80665          ' acceleration of gravity, m/s^2
1100    DYVISC = .0000178       ' dynamic viscosity of air, N*s/m^2
1120    DIA = DIA               ' diameter of droplet, m
1125   'DIA.MIC                   diameter of droplet in microns
1200    CD = 0                  ' drag coefficient of droplet
1210    CD.STOKE = 0            ' drag coefficient of droplet by Stoke's law
1400    RE = 0                  ' Reynolds number of droplet
1410    RELAST = 0              ' Reynolds number of droplet previous loop
1430    TIME = 0                ' time from emission of droplet, sec
1500  RETURN
1990  '*********************************************************
2000  ' SUBROUTINE TO DETERMINE THE DROP.LIFE
2010  ' --------------------------------------------------------
2020  '
2040    PRANDTL.TO.THIRD = .9   'Prandtl number = .72
2045    GOSUB 2100     : DIA.MIC=DIA*1000000!
2050    EVAP.COEF1 = 84.76 * (1 + .3 * PRANDTL.TO.THIRD * SQR(RE))
2060    IF WET.BULB.DEP = 0 THEN DROP.LIFE = 1E+15    :    RETURN
2070    DROP.LIFE = (DIA.MIC * DIA.MIC) / (EVAP.COEF1 * WET.BULB.DEP)
2080    RETURN
2085  '
2090  ' ****************************************************
2100  ' Subroutine to find the terminal velocity for droplet
2110    ----------------------------------------------------
2120  '
2130    KINVISC = DYVISC / DENAIR : REYNOLD.COEF = DIA / KINVISC
2140    CD = 1                          'first guess of CD
2150    VEL.TERM.COEF = SQR(4 * DENDROP * GRAV * DIA / (3 * DENAIR))
2160    VEL.TERM = VEL.TERM.COEF / SQR(CD)
2170     RE = REYNOLD.COEF * VEL.TERM
2180     GOSUB 370                      'To get new CD
2190     VEL.T = VEL.TERM.COEF / SQR(CD)
2200     IF ((VEL.T/VEL.TERM)<=.999) THEN VEL.TERM = VEL.T  : GOTO 2170
2205     IF ((VEL.T/VEL.TERM)=>1.001) THEN VEL.TERM = VEL.T : GOTO 2170
2208       VEL.TERM = (VEL.T + VEL.TERM)/2
2210    RE = REYNOLD.COEF * VEL.TERM    :   PREV.RE = RE
2220    RETURN
2230  '
2290  '*******************************************************
2300    Subroutine to determine evaporated DIAmeter of droplet
2310  ' ------------------------------------------------------
2320  '
2325    EVAP.VALUE=1-TIME/DROP.LIFE :IF EVAP.VALUE<0 THEN RETURN 359
2330    EVAP.COEF2 = SQR(EVAP.VALUE)
2340    DIA = EVAP.COEF2 * INIT.DIA
2350    RETURN
2360  '
2990  ' *******************************************************
```