Effects of wind on rainfall simulation

Spatial distribution of water application, kinetic energy, and overland flow under single full-cone nozzle sprays

J.L.M.P. de Lima[1] and P.J.J.F. Torfs[2]

(1) Department of Civil Engineering, Faculty of Science and Technology, University of Coimbra, 3049 Coimbra Codex, Portugal

(2) Department of Water Resources, Wageningen Agricultural University, Wageningen, The Netherlands

RAPPORT 40

Augustus 1993

Vakgroep Waterhuishouding
Nieuwe Kanaal 11, 6709 PA Wageningen

ISSN 0926-230X

CONTENTS

ABSTRACT

Portable rainfall simulators used in small plots give sufficient flexibility to study a wide variety of processes (e.g. infiltration, interrill erosion, water quality) on different soils and slopes for a wide range of land uses. They also allow a number of repetitions in a short period of time.

In conducting these studies, it is important to measure the characteristics of the simulated rainfall applied to the test surface. It is unquestionable that wind affects field experiments that make use of rainfall simulators. Water-drop trajectories and velocities are altered, affecting water application, kinetic energy distributions, and the hydraulics of underlying overland flow, namely water depths and velocities.

A three-dimensional numerical model was developed from the movement and physical conditions of individual drops, in order to estimate the distances and times which are necessary for them to reach the ground after their release from the nozzle of a simulator. The original momentum of the drop is affected by drag forces, wind and gravity. A logarithmic wind profile above the soil surface was assumed in all situations. Water application and kinetic energy distributions were estimated from the coupling of the hydrodynamic model for drop movement, a stochastic drop generator representing a single full-cone spray nozzle, and an appropriate interception algorithm at the soil surface. The hydraulic characteristics of overland flow are strongly related to the properties and spatial variability of the simulated rainfall. The spatial distribution of rainfall intensity, mean drop size, and drop incidence angles have been combined with an analysis of wind shear stress at the water-air boundary, in order to study the influence of wind on the mechanics of overland flow under spraying systems. This was done by analysis of the distribution of the overland flow driving force on the ground plane.

The mathematical model presented in this paper should facilitate the selection of single full-cone spray-nozzles (spray angles, ejection velocities, and drop-size distribution) and of the size and configuration of the spray area for expected field situations (rainfall characteristics, wind conditions, slope of terrain, and plot size). It provides a simple way of visualizing spray patterns and overland flow on different sloping surfaces and for different wind conditions.

1. INTRODUCTION

1.1. WIND/RAINFALL INTERACTION IN EROSION

The erosivity of wind–driven rain can differ drastically from that of rain falling under windless conditions (LAL et al., 1980) or that of wind under rainless conditions (LIMA et al., 1992). Highly erosive rains are generally those in which peak rainfall intensity and peak wind velocity coincide (AINA et al., 1977). The magnitude of the difference depends on wind and rainfall characteristics, on other meteorological factors like air and soil temperature and relative humidity, on soil characteristics, and on land use. In erosion, the effects of wind on rain and vice–versa are complex and influenced by many variables.

1.1.1. Effect of wind on water erosion

Wind is an unavoidable natural phenomenon that has long been recognized as an important factor in water erosion. Wind can cause rain to fall at a considerable inclination (UMBACK and LEMBKE, 1966; STRUZER, 1972; SHARON, 1980). Falling through a logarithmic wind profile (expected profile under storm conditions, ROSENBERG et al., 1983), raindrops arrive at the surface with most of the horizontal momentum they possessed at higher altitudes (CALDWELL and ELLIOTT, 1972). Raindrop shape is altered to an oblate spheroid with a flattened area between the bottom and upwind side (DISRUD et al., 1969). Wind alters the angle of raindrop impact and, by adding a horizontal component to the drop velocity (thus increasing the kinetic energy of rainfall), wind increases its detaching capacity (LYLES, 1977). Many researchers have studied the role of raindrops in soil erosion, especially how raindrops influence soil detachment (e.g. YOUNG and WIERSMA, 1973; MOSS and GREEN, 1983). Soil detachment is known to increase in wind–driven rainfalls. LYLES et al. (1969) and DISRUD and KRAUSS (1971) reported that soil detachment from clods exposed to wind–driven rain was greater than that caused by similar rainfall intensities without wind. Wind affects the relative proportion of upslope versus downslope splash (LYLES et al., 1974). Highly erosive rains are generally those in which peak rainfall intensity and peak wind velocity coincide (AINA et al., 1977).

Wind influences the distribution and the amount of precipitation (SHARON, 1980 and LIMA, 1990) and the size of raindrops by breaking large drops into smaller droplets (LYLES et al., 1969 and DISRUD et al., 1969). Wind also influences the mechanics of the overland flow process (LIMA, 1989a, b and c).

1.1.2. Effect of rainfall on wind erosion

The main factor in wind erosion is the velocity of moving air. The initiation of sediment transport by wind is strongly influenced by the relative air humidity (KNOTTNERUS, 1980a and 1980b), by precipitation (PLOEY, 1980), and by soil moisture (AZIZOV, 1977). Although many strong wind events are accompanied by rainfall (LYLES et al., 1974; LYLES, 1977), only limited knowledge is available on the effect of rain in increasing (or decreasing) the ability of the wind to cause movement and abrasion of soil. JUNGERIUS et al. (1981) and LIMA et al. (1992) recorded appreciable wind erosion on beach sands and dunes during rainy days. This can be attributed to the transport of soil particles lifted into the air by raindrop splash and their higher transport trajectories as compared to particles that are lifted primarily by wind (JUNGERIUS and DEKKER, 1990; LIMA et al., 1992).

1.2. WIND AND SPRINKLER IRRIGATION

CHRISTIANSEN (1942) studied the effect of wind on single sprinkler patterns and found that the effect on the distribution was very significant. Many other researchers (VORIES and BERNUTH, 1986; etc.) have studied the effect of wind on sprinkler performance.

Sprinkler manufacturers offer a variety of sprinkler products designed to counteract the effects of wind. They are described as reducing wind drift and evaporation, with lower trajectory angles while still maintaining reasonable coverage (BERNUTH, 1988).

1.3. EFFECTS OF WIND ON RAINFALL SIMULATIONS

To collect sufficient information from natural rainfall events, from field plots requires a long time (10 to 20 years) and considerable financial outlay. Where erosion plots already exist, they should be used. Otherwise, it is more cost–effective to use rainfall simulators to apply a controlled rainstorm to a small plot and then collect the runoff and sediment arising from it. Portable rainfall simulators used in small plots will yield data in a relatively short time, and will give sufficient flexibility to study a variety of processes, on different soils and slopes, for a wide range of land uses. They will also allow a number of replicates in a short period of time. Normally, single nozzle rainfall simulators are more suitable for rugged terrain.

Nowadays rainfall simulators are commonly used for infiltration, erosion, water quality and hydrologic (rainfall–runoff) studies.

Knowledge of the mechanics of soil erosion and of the hydraulic properties of shallow overland flow has increased in recent years. For these studies, it is important to measure the characteristics of simulated rainfall applied to the test surface.

Two major types of rainfall simulators are found in the literature: simulators employing nozzles (mobile or stationary and single or multiple) and drop formers. This report deals only with single full–cone nozzle sprays (Fig. 1.1).

Rainfall simulations have become quite common, especially in water erosion studies. It is unquestionable that wind effects field experiments that use rainfall simulators. Researchers try to avoid the problem by waiting for windless periods or by isolating the boundary of their runoff plots with windscreens. On the other hand, if they conduct their experiments under windy conditions, the drop trajectories and velocities will be affected together with the hydraulic characteristics of overland flow, namely water depths and velocities, introducing more variables into their experimental data and making it more difficult to draw conclusions from the data set.

Raindrops vary in size from very tiny droplets barely larger than fog to a maximum diameter of slightly over 7 mm. Most rainwater comes down in drops of between 1 and 4 mm in diameter. In every instant of a simulated rainfall, a large variety of drop sizes can occur. The wind effect is greater on smaller drops falling slowly than on large drops with higher velocities.

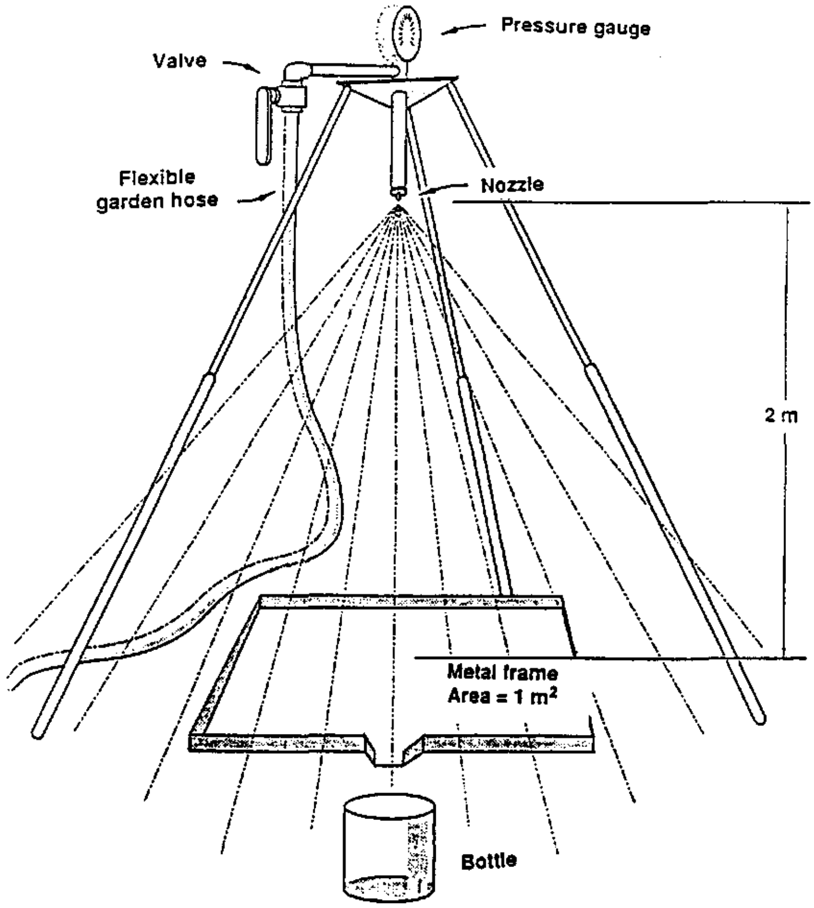


Fig. 1.1 – Simple setup with a single full–cone nozzle.

1.4. QUESTIONS THAT NEED TO BE ANSWERED

QUESTION 1 – How important is the wind effect on rainfall simulations in the field? What are the variables involved?

QUESTION 2 - What type of rainfall simulator (using nozzles or drop formers) is more susceptible to wind effects?

QUESTION 3 – What are the wind conditions above which field experiments are not to be undertaken without windscreens?

QUESTION 4 – How realistic/natural are simulations results if wind effects are neglected?

This report deals mainly with Questions 1 and 2.

2. NOZZLES AND SPRAY-NOZZLE DROP SIZES

2.1. NOZZLES

2.1.1. Introduction

Spray nozzles are vital components in rainfall simulations. Different spray nozzles produce different spray patterns. Full cone sprays (solid circular pattern of drops) are commonly used in rainfall simulations. Their performance is critical to the efficiency of the system.

Most spray nozzles can be set to a variety of spray angles, to determine the size and configuration of the spray area (Fig. 2.1). In actual operation, due to gravity and wind,spray drops do not follow straight lines after leaving the nozzle, (Fig. 2.1).

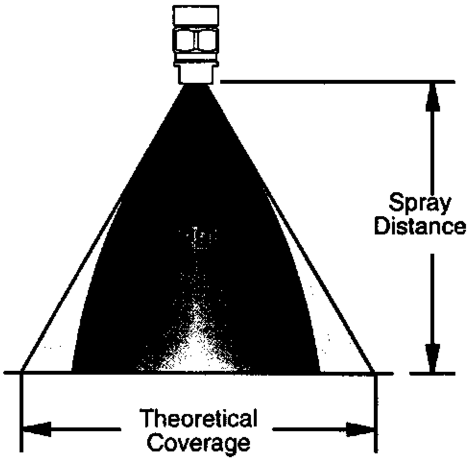


Fig. 2.1 – Spray angle, theoretical coverage, and reduced coverage in actual operation (Spraying Systems, 1991).

2.1.2. Types of nozzles

There are five major types of nozzles. They are:

– *Whirljet nozzles*:

They produce a spatial distribution of small to medium drops over a wide range of flow rates and pressures (Fig. 2.2). Whirljet nozzles also have the advantage of a large, unobstructed flow passage, which minimizes the chance of clogging.

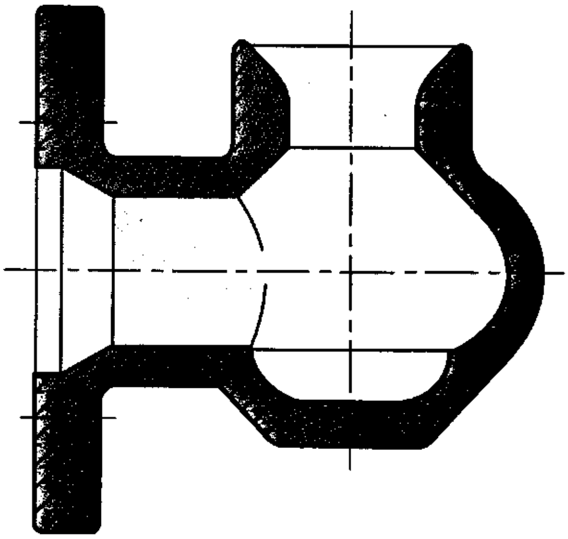


Fig. 2.2 – Whirljet nozzle (Spraying Systems, 1992a).

– *Spiraljet nozzles*:

These nozzles have a spiral or helical design (Fig. 2.3). For any given pipe size, a spiraljet nozzle offers the highest flow rate (Spraying Systems, 1991). Their structural strength is limited however.

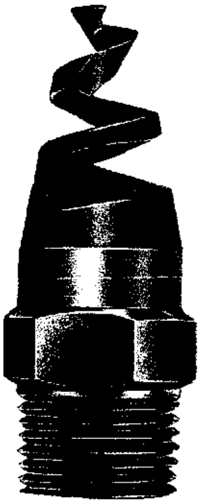


Fig. 2.3 – Spiraljet nozzle (Spraying Systems, 1992a).

– *Fulljet nozzles*:

These nozzles produce a spray of medium to large drops that are uniformly distributed across the solid pattern (Fig. 2.4). Some have an internal vane to control the spraying. The vaneless design allows for unrestricted internal flow. Vaneless fulljet nozzles are best suited to applications requiring a full–cone spray pattern with coarser spray characteristics and greater free–passage dimensions through the nozzle.

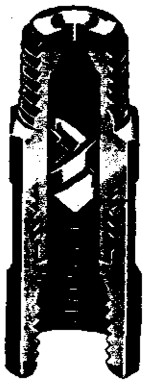


Fig. 2.4 – Fulljet nozzle (Spraying Systems, 1992a).

– *Atomizing nozzles*:

Atomizing nozzles are engineered to produce extremely fine drop sizes. There are two basic types: hydraulic atomizing and air atomizing. Hydraulic atomizing nozzles use the liquid pressure alone to produce extremely fine drops in a hollow cone pattern. Air atomizing nozzles mix liquid and air to produce a completely atomized spray.

– *Flowback nozzles*:

These nozzles allow a portion of the liquid fed to be returned (rather than continuing through the nozzles) to stabilize pressure while the flow rate fluctuates (Fig. 2.5). As a result, optimum drop size is always maintained.

2.2. SPRAY NOZZLE DROP SIZES

2.2.1. Introduction

Accurate assessment of spatial drop size distribution is an important factor in the overall effectiveness of spray performance, affecting both water applications and kinetic energy distributions (see Chapter 4). The sensitivity of the system is strongly affected by the distribution of drop sizes (see Chapter 5).

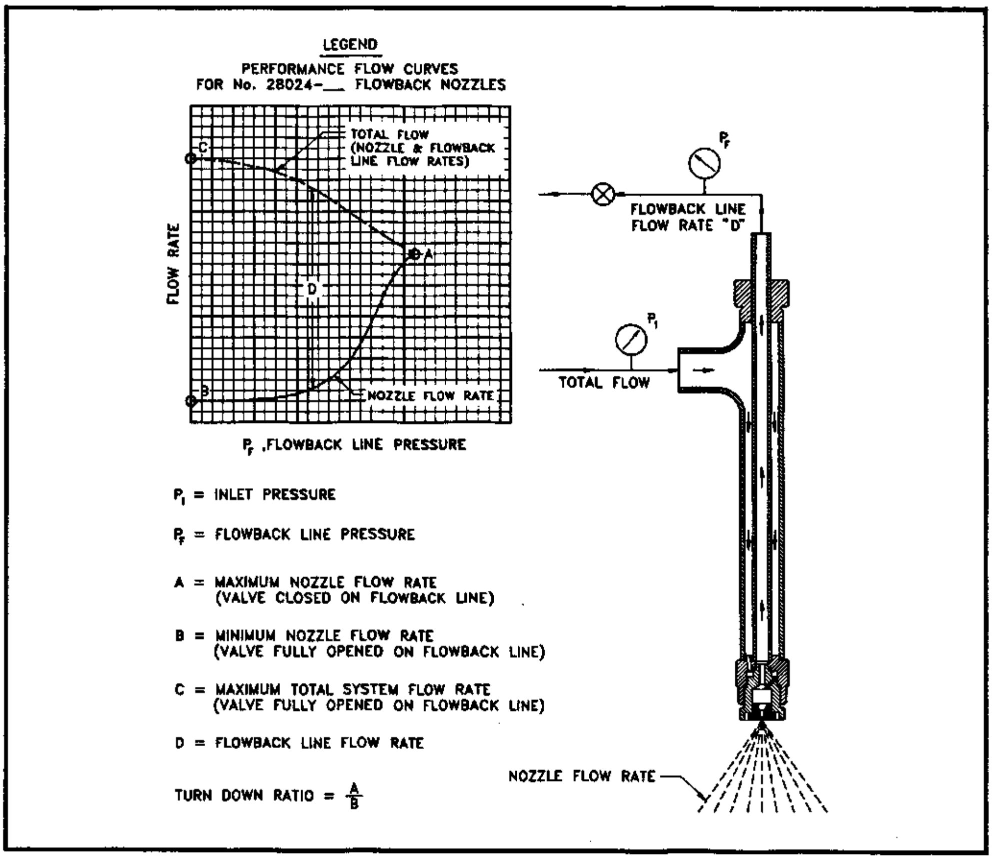
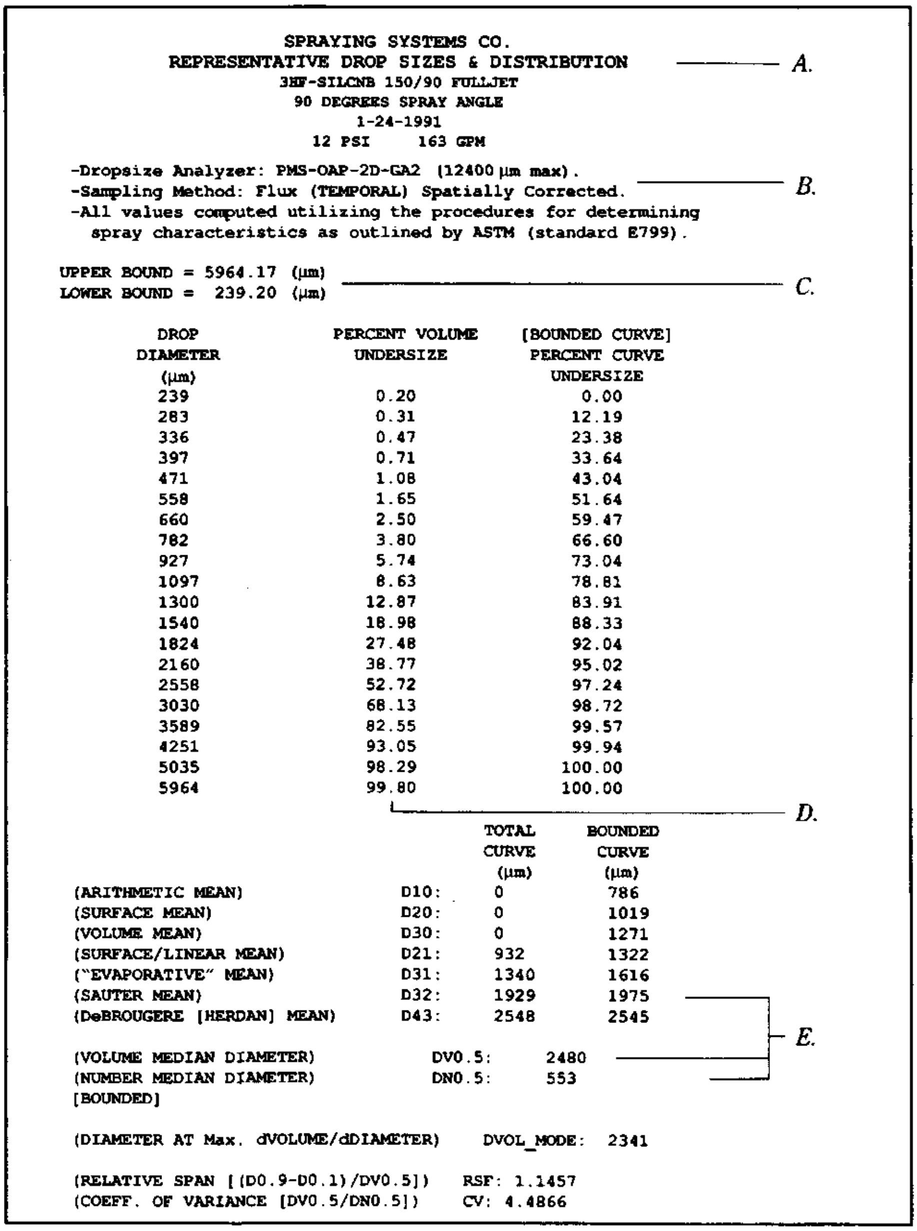

LEGEND
PERFORMANCE FLOW CURVES FOR No. 28024-__ FLOWBACK NOZZLES
TOTAL FLOW (NOZZLE & FLOWBACK LINE FLOW RATES)
C
A
D
B
NOZZLE FLOW RATE
FLOW RATE
P_F ,FLOWBACK LINE PRESSURE
P_I = INLET PRESSURE
P_F = FLOWBACK LINE PRESSURE
A = MAXIMUM NOZZLE FLOW RATE (VALVE CLOSED ON FLOWBACK LINE)
B = MINIMUM NOZZLE FLOW RATE (VALVE FULLY OPENED ON FLOWBACK LINE)
C = MAXIMUM TOTAL SYSTEM FLOW RATE (VALVE FULLY OPENED ON FLOWBACK LINE)
D = FLOWBACK LINE FLOW RATE
TURN DOWN RATIO = A/B
P_F
FLOWBACK LINE FLOW RATE "D"
P_I
TOTAL FLOW
NOZZLE FLOW RATE

Fig. 2.5 – Flowback nozzle (Spraying Systems, 1992b).

2.2.2. How to measure drop size

Accurate drop–size analysis requires several sophisticated measuring aids, including photo–imaging, laser light diffraction, a linear diode and a phase–Doppler. With this state–of–the–art instrumentation, it is possible to characterize sprays and obtain valuable information such as drop diameter averages, drop distribution and velocity profiles. Table 2.1 gives a typical spray–drop analysis report. These drop–size analysis can also be made with classical techniques such as the oil method or pressure transducers (see Section 2.2.3).

Table 2.1 Spray characterization (Spraying Systems, 1991).



```
                    SPRAYING SYSTEMS CO.
          REPRESENTATIVE DROP SIZES & DISTRIBUTION          ———— A.
                  3HF-SILCNB 150/90 FULLJET
                   90 DEGREES SPRAY ANGLE
                         1-24-1991
                    12 PSI      163 GPM

 -Dropsize Analyzer: PMS-OAP-2D-GA2 (12400 µm max).
 -Sampling Method: Flux (TEMPORAL) Spatially Corrected.  ———— B.
 -All values computed utilizing the procedures for determining
   spray characteristics as outlined by ASTM (standard E799).

UPPER BOUND = 5964.17 (µm)
LOWER BOUND =  239.20 (µm)  —————————————————————————————— C.

      DROP          PERCENT VOLUME    [BOUNDED CURVE]
    DIAMETER          UNDERSIZE        PERCENT CURVE
      (µm)                               UNDERSIZE
      239               0.20               0.00
      283               0.31              12.19
      336               0.47              23.38
      397               0.71              33.64
      471               1.08              43.04
      558               1.65              51.64
      660               2.50              59.47
      782               3.80              66.60
      927               5.74              73.04
      1097              8.63              78.81
      1300             12.87              83.91
      1540             18.98              88.33
      1824             27.48              92.04
      2160             38.77              95.02
      2558             52.72              97.24
      3030             68.13              98.72
      3589             82.55              99.57
      4251             93.05              99.94
      5035             98.29             100.00
      5964             99.80             100.00
                        L———————————————————————————— D.
                              TOTAL      BOUNDED
                              CURVE      CURVE
                              (µm)       (µm)
 (ARITHMETIC MEAN)       D10:  0          786
 (SURFACE MEAN)          D20:  0          1019
 (VOLUME MEAN)           D30:  0          1271
 (SURFACE/LINEAR MEAN)   D21:  932        1322
 ("EVAPORATIVE" MEAN)    D31:  1340       1616
 (SAUTER MEAN)           D32:  1929       1975  ——————┐
 (DeBROUGERE [HERDAN] MEAN) D43: 2548     2545        │
                                                      ├ E.
 (VOLUME MEDIAN DIAMETER)   DV0.5:    2480  ——————————┤
 (NUMBER MEDIAN DIAMETER)   DN0.5:    553       ——————┘
 [BOUNDED]

 (DIAMETER AT Max. dVOLUME/dDIAMETER)    DVOL_MODE:  2341

 (RELATIVE SPAN [(D0.9-D0.1)/DV0.5])   RSF: 1.1457
 (COEFF. OF VARIANCE [DV0.5/DN0.5])    CV: 4.4866
```

A. Nozzle designation, nominal spray angle, discharge pressure and flow rate.

B. Sampling method and statement of conformance to ASTM-E799 (Revision 1987).

C. Minimum and maximum drop diameters of accumulated (test) volume.

D. Per ASTM-E799-87; section 4.5, less than 1.0% of the cumulative volume is contained in the largest drop measured.

E. Three principal means of reporting drop size are commonly requested. All other values shown (arithmetic mean, surface mean, etc.) are provided for conformance to the ASTM-E799-87 standard.

The American Society for Testing and Materials (ASTM, 1981) recognizes two different types of drop–size sampling techniques: spatial, and temporal or flux–sensitive.

The spatial technique is used when a collection of drops occupying a given volume is sampled instantaneously (Fig. 2.6). Generally, spatial measurements are collected by holographic means or with high–speed photography.

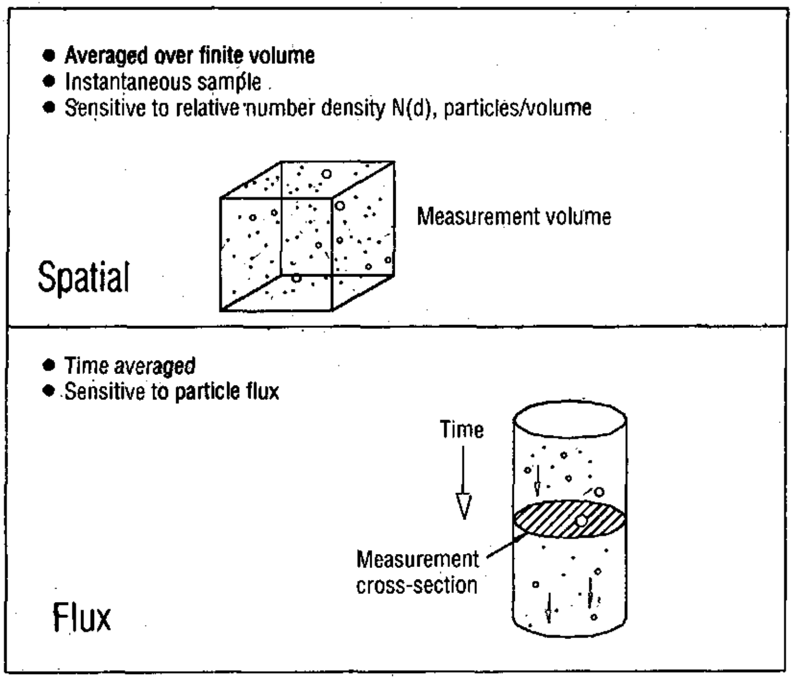

• Averaged over finite volume
• Instantaneous sample
• Sensitive to relative number density N(d), particles/volume
Measurement volume
Spatial
• Time averaged
• Sensitive to particle flux
Time
V
Measurement cross-section
Flux

Fig. 2.6 – Spatial and flux sampling (ASTM, 1981).

The flux technique is applied when individual drops passing through the cross section of the sampling region are examined during an interval of time (Fig. 2.6). Flux measurements are generally collected by optical or electro–mechanical sensing of individual drops.

If all drops in a spray are moving at the same velocity, the flux and spatial distribution are identical. The spray will generally exhibit differences in drop velocities that vary from class size to class size, depending on the type of nozzle, capacity, and spraying pressure. Therefore, it is important to combine measurement techniques and equipment for measuring drop diameter (FERRAZZA et al., 1992).

2.2.3. Methods and equipment for measuring of drop size

There are many methods and instruments available for drop–size data collection. The following is a brief description of some of these techniques.

By exposing a pan of oil to simulated rainfall, one can count and size individual drops using a microscope (oil method). Because of spray velocity problems with this method involve drop coalescence, inadequate sample size, and the fact that very small drops will be deflected away from the oil by air currents at the surface. Larger drops can and do break up on impact with the surface. This method is based on the premise that water drops suspended in a less dense but viscous fluid assume a near–perfect shape owing to the surface tension forces (EIGEL and MOORE, 1983).

A similar method is the spraying of water drops into liquid sensitive paper (stain method) or into a tray with flour (flour method). Again, the small drops might be deflected away from the target and the large drops can break up on impact.

Momentum methods that include pressure transducers and piezoelectric sensors have also been successfully used to measure raindrop size and energy. These instruments are subject to interference between drops and sensitivity (EIGEL and MOORE, 1983).

The first breakthrough in drop–sizing technology was the development of the automated imaging analyzer (Fig. 2.7). This incorporates the spatial measurement technique, which uses a strobe light to illuminate the spray and record the image with a vidicon tube. The image is scanned, and the drops are sized and separated into different classes (FERRAZZA et al., 1992).

Other spatial–sampling particle analyzers utilize the fact that a spray drop will scatter laser light through an angle dependent on the diameter of the drop (Fig. 2.8). The scattered light intensity is measured with a series of diodes. A curve–fitting program is used to convert the light intensity distribution into any of several empirical drop–size distribution functions (FERRAZZA et al., 1992).

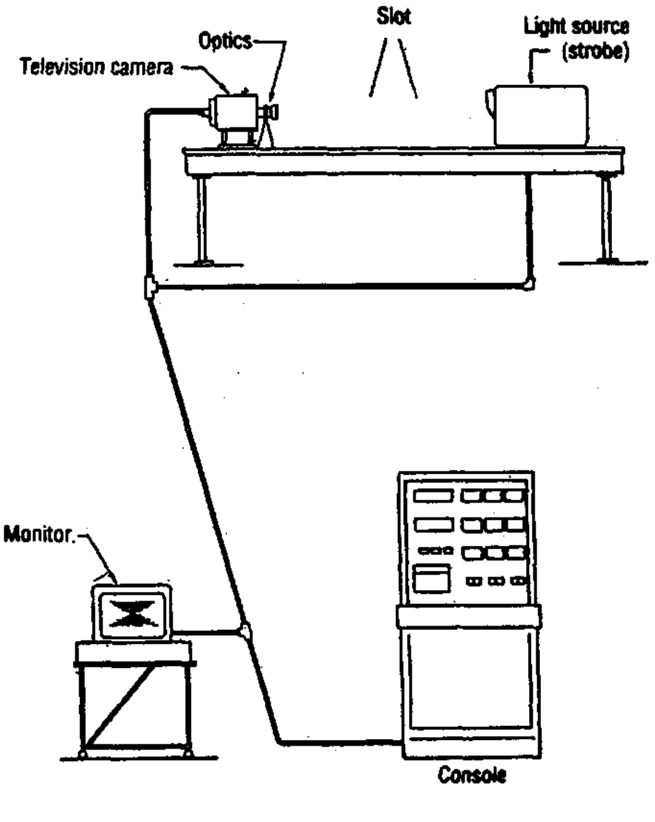


Fig. 2.7 – Automated imaging analyzer (FERRAZZA et al., 1992).

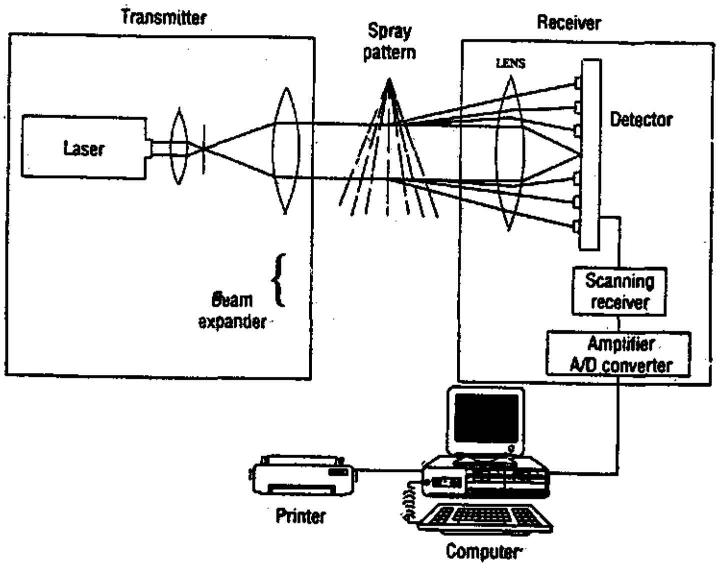


Fig. 2.8 – Laser particle analyzer (FERRAZZA et al., 1992).

The optical array spectrometer probes are flux–sampling instruments (Fig. 2.9). As the drops pass through the sampling plane, they are sized and counted, providing information which can also be used to determine drop velocity (FERRAZZA et al., 1992).

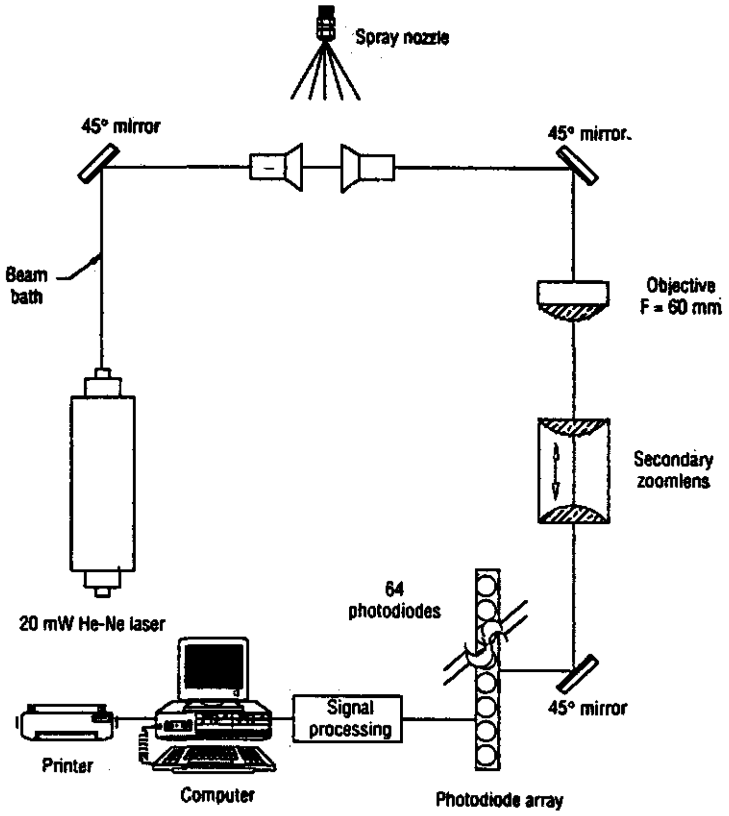


Fig. 2.9 – Optical array spectrometer probes (FERRAZZA et al., 1992).

The aerometric phase Doppler particle analyzer is a point–sampling, flux–sensitive instrument (Fig. 2.10) that uses a low–power laser. The laser is split into two beams by a beam splitter and a frequency module. The two beams intersect again into a single beam at the sample volume location. When a drop passes through the intersection region of the two beams, an interference fringe pattern is formed by the scattered light. It is important to test several points within the spray in order to obtain a combined result that is representative.

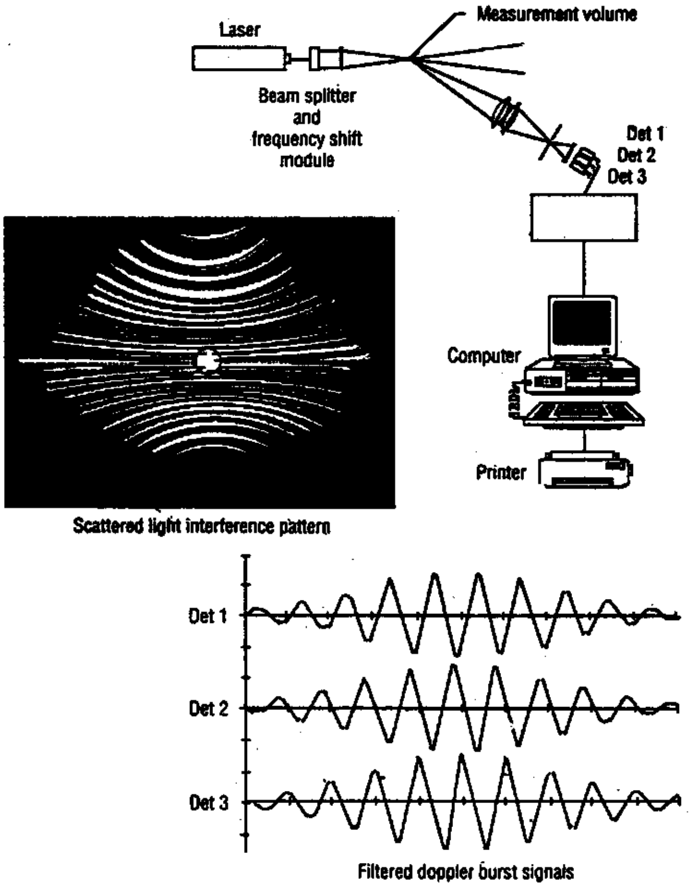

Measurement volume
Laser
Beam splitter and frequency shift module
Det 1
Det 2
Det 3
Computer
Printer
Scattered light interference pattern
Det 1
Det 2
Det 3
Filtered doppler burst signals

Fig. 2.10 – Aerometric phase Doppler particle analyzer (FERRAZZA et al., 1992).

3. MODELLING OF DROP MOVEMENT IN WIND

3.1. INTRODUCTION

Once they are formed on a drop–former or catapulted from a nozzle, drops begin to move under the action of gravity and frictional forces. The latter arises from their motion relative to the air, which can also be moving under wind. They will undergo complex hydrodynamic interactions, causing phenomena such as drop deformation, internal circulation, evaporation, vibration, and collision. All these phenomena make a quantitative description of their hydrodynamic behaviour extremely difficult. Their can be understood reasonably well, however through the use of several simplifications.

OKAMURA (1968) found, from a photographic study, that the deformation of drops in flight is random and concluded that the assumption of an average spherical shape is reasonable. The phenomena of internal circulation, vibration, evaporation, breakup, and collision will not be included in the present study.

Flow past an object may be regarded as incompressible under certain conditions (PRUPPACHER and KLETT, 1978). As these conditions hold for all cloud particle motions, we shall henceforth assume that the flow under consideration is incompressible (PRUPPACHER and KLETT, 1978). For Newtonian fluids, of which air and water are examples, the continuity and the momentum equation (Navier–Stokes equation) for incompressible flow in the presence of gravity acquires the form:

$$\nabla \cdot \vec{u} = 0 \qquad (1)$$

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\frac{\nabla p}{\rho_a} + \nu_a \nabla^2 \vec{u} + \vec{g} \qquad (2)$$

where $\vec{u}$ is the velocity, t is time, ρ_a is the fluid density, g is the gravitational acceleration, p is the fluid pressure, ν_a is the fluid kinematic viscosity, ∇ is a gradient operator, and $\cdot$ is the divergence operator.

This system of equations must be supplemented with suitable boundary conditions.

A model was developed based on the movement and physical conditions of individual drops, in order to estimate necessary distances and times for water drops of various sizes to reach the ground from the nozzle. The original momentum of the drop is acted upon by drag forces, wind, and gravity.

The ballistics of a water drop depend on drop size, so a thorough study must consider a range of drop sizes.

3.2. THREE-DIMENSIONAL MODEL DESCRIPTION

3.2.1. Theoretical development

For a nozzle rainfall simulator, zero wind-speed is the only situation for which two-dimensional analysis is sufficient. For non-zero wind-speed, a three-dimensional model (which is symmetrical to the wind direction) must be constructed.

The movement of a drop is an unsteady-flow problem because the drop accelerates under the influence of gravity, a possible catapulting action (in the case of a nozzle rainfall simulator), buoyancy, and aerodynamic force. A completely rigorous treatment of this problem of accelerating motion is complicated (see 3.1). Arguments presented elsewhere (PRUPPACHER and KLETT, 1978) demonstrate that the effect of local fluid acceleration is negligible because $\rho_a/\rho_w << 1$. Hence it is sufficient to use the steady-state drag formulas to describe the hydrodynamic resistance to the drops. In view of the above, Newton's second law of motion will be applied to the description of the velocity history of the drop involving the drag, buoyancy, and gravity forces only. These forces must sum vectorially to equal the acceleration of the drop:

$$m\frac{du_x}{dt} = -\frac{1}{2}\rho_a V_R^2 A\, C_D\, e_x \tag{3}$$

$$m\frac{du_y}{dt} = -\frac{1}{2}\rho_a V_R^2 A\, C_D\, e_y \tag{4}$$

$$\underset{(I)}{m\frac{du_z}{dt}} = \underset{(II)}{-\frac{1}{2}\rho_a V_R^2 A\, C_D\, e_z} + \underset{(III)}{mg} - \underset{(IV)}{\frac{\rho_a}{\rho_w}mg} \tag{5}$$

where

$$e_x = \frac{u_x - w}{V_R}\ ;\quad e_y = \frac{u_y}{V_R}\ ;\quad e_z = \frac{u_z}{V_R} \tag{6}$$

where u_x, u_y and u_z are the velocity components of the drop (m/s), e_x, e_y and e_z are the components of the unit vector giving the direction of the relative speed of the drop with respect to the wind (m/s), w is the wind speed which is assumed to blow in the X-direction (m/s), m is the mass of the drop (kg), ρ_a and ρ_w are the densities of the air and of the water (kg/m^3), A is the characteristic cross section of the drop perpendicular to the relative flow (m^2), C_D is the drag coefficient of the spherical drop, and V_R is the absolute value of the relative velocity of the drop with respect to the wind (m/s).

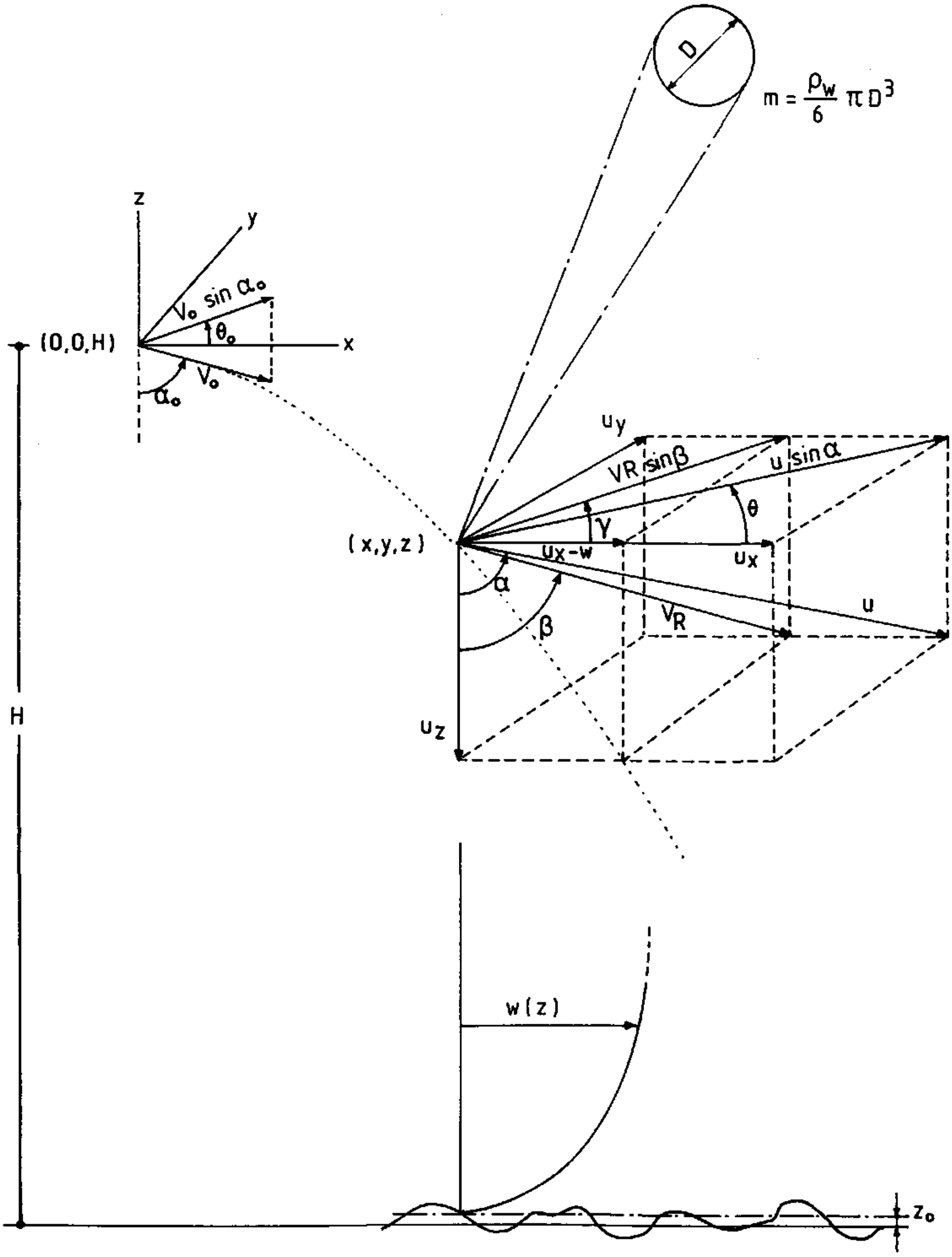


Fig. 3.1 – Parameters involved in the study of the movement of a drop in wind, from the nozzle to the ground surface.

In equations (3) to (5), term (I) represents the acceleration of the drop, term (II) is the drag force, term (III) is the gravity, and term (IV) is the buoyancy force. Equations (1) to (3) can be rewritten in the following form (see Fig. 3.1):

$$\frac{du_x}{dt} = -\frac{3}{4}\frac{\rho_a}{\rho_w}\frac{C_D}{D}V_R^2\,e_x \tag{7}$$

$$\frac{du_y}{dt} = -\frac{3}{4}\frac{\rho_a}{\rho_w}\frac{C_D}{D}V_R^2\,e_y \tag{8}$$

$$\frac{du_z}{dt} = g(1-\frac{\rho_a}{\rho_w}) - \frac{3}{4}\frac{\rho_a}{\rho_w}\frac{C_D}{D}V_R^2\,e_z \tag{9}$$

with

$$V_R = \sqrt{(u_x-w)^2 + u_y^2 + u_z^2} \tag{10}$$

$$\beta = \arctan\,(\sqrt{(u_x-w)^2 + u_y^2} \,/\, u_z) \tag{11}$$

$$\gamma = \arctan\,(u_y \,/\, (u_x - w)) \tag{12}$$

$$m = \frac{\rho_w}{6}\,\pi\,D^3 \tag{13}$$

where D is the drop diameter (assumed spherical) (m), V_R is the velocity with respect to the air (m/s), w is the wind speed (m/s), β is the angle of the relative drop velocity with respect to the wind with the vertical (rad), and γ is the azimuth of the relative drop velocity with respect to the wind (rad).

The drag coefficient expresses the total force from friction and form components. If C_D were constant, these equations could be integrated analytically. The drag coefficient, however, depends upon the velocity (Reynolds number) and shape of the drops. A complete analytical solution is not possible and the problem was solved numerically using Euler's method. Factors acting on the drops are considered at short–time increments and interaction continues until the ground surface is reached. The drop Reynolds number is defined by:

$$Re_d = \frac{V_R\ D}{\nu_a} \tag{14}$$

where Re$_d$ is drop Reynolds number (–), V$_R$ is the velocity with respect to the air (m/s), D is the drop diameter, and ν_a is the kinematic viscosity of the air (m^2/s), defined by (EDLING, 1985):

$$\nu_a = 1.3045\ 10^{-5}+1.222\ 10^{-7}\ T_a\ -\ 9.6471\ 10^{-10}\ T_a^2\ + \\ +\ 7.2873\ 10^{-12}\ T_a^3 \tag{15}$$

where T$_a$ is the air temperature (°C), assumed constant with height. The density of the air can also be expressed as a function of temperature by the equation (EDLING, 1985):

$$\rho_a = 1.29331\ -\ 0.00496\ T_a\ +\ 2.807\ 10^{-5}\ T_a^2\ - \\ -\ 1.88\ 10^{-7}\ T_a^3 \tag{16}$$

Following HOERNER (1958) and WILLIAMSON and THREADGILL (1974) the drag coefficient of a sphere as a function of the drop Reynolds number is calculated from the following equations:

$$C_D = \frac{24}{Re_d} \qquad if\ \ Re_d < 0.5 \tag{17}$$

$$C_D = \frac{26.38}{Re_d^{0.845}} + 0.49 \qquad if\ \ 0.5 \le Re_d < 200 \tag{18}$$

$$C_D = \sqrt{0.525\ +\ (\frac{24}{Re_d})^2} \qquad if\ \ 200 \le Re_d \le 10000 \tag{19}$$

$$C_D = 0.44 \qquad\qquad if\ \ Re_d > 10000 \qquad\qquad (20)$$

In this study, a logarithmic wind profile above the soil surface was assumed in all cases. Wind blows in the direction of the positive X–axis.

A constant wind speed is assumed, which may be unrealistic in a convective storm when drop size and wind velocities vary all the time. No allowance is made for the local turbulence created by the simulator. Under neutral conditions, when temperature is constant with height, the mean wind speed increases linearly with the logarithm of the height (Rosenberg et al., 1983) (see Fig. 3.2):

$$w = \left(\frac{u*}{K}\right) \ln\frac{z_1}{z_o} \qquad for\ \ z_1 > z_o \qquad (21)$$

with

$$u* = \frac{w_{10}K}{\ln\dfrac{10}{z_o}} = \sqrt{\frac{\tau}{\rho_a}} \qquad (22)$$

where z_0 is the roughness length (m), z_l is the height above the ground along the Z_l–axis (if an overland flow sheet of thickness h is present then $z_l = z_l - h$, and z_0 refers to the roughness of the water surface; see Equation 41 for the calculation of z_l) (m), w_{10} is the wind speed at 10 m (standard measuring height at meteorological stations), K is the Von Karman constant (approximately 0.4), τ is the frictional shear stress (Pa), and u* is the friction velocity (m/s).

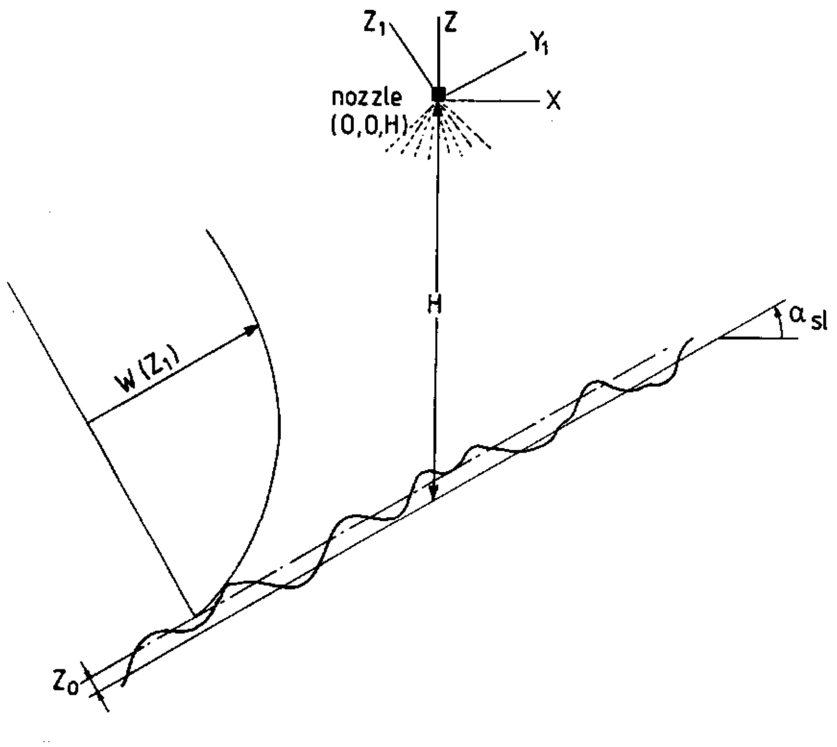


Fig. 3.2 – The nozzle spraying a sloping surface,with a logarithmic wind profile, blowing upslope.

4. MODELLING OF WATER APPLICATION AND KINETIC ENERGY DISTRIBUTIONS

4.1 WATER APPLICATION DISTRIBUTION

Loss of water between the nozzle and the ground can be divided into two components: (1) drift loss (out of the area under study); (2) evaporation loss. The drift loss, which almost by definition is a result of the wind alone, need not be well–correlated with the evaporation loss, which is only partially affected by the wind. When measuring with rain gauges, one should also expect wind–induced losses. Fig 4.1 shows a fulljet making a solid circular pattern of drops under windless conditions.

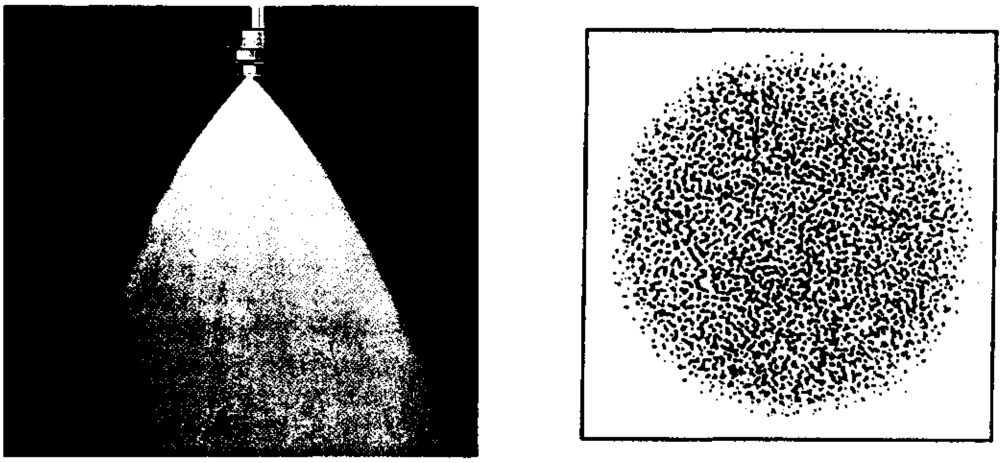


Fig. 4.1 – Fulljet nozzle making a solid circular pattern of drops under windless conditions.

The pattern of water application is distorted by wind. If we assume that the windless application is uniform, any distortion of application may lead to lower uniformity. If application is deficient, lower results can be expected from the rainfall simulation.

When we compute the rate of soil erosion by rainfall, it is important to assess how much rain fell on sloping ground.

The rainfall flux intercepted on sloping ground depends on the angle of incidence of the rain, the inclination of the surface, and the relative orientation of the sloping surface to the rain vector (STRUZER, 1972; SHARON, 1980). Thus, for a given inclination of rainfall, the proportion of rain actually intercepted by a sloping surface will differ from the rainfall collected on a horizontal surface.

4.2 KINETIC ENERGY DISTRIBUTION

The kinetic energy output of rainfall simulators. Erosivity indices based on the kinetic energy of the rainfall have been proposed to characterise erosion by overland flow and rills. Several researchers (MOLDENHAUER and KEMPER, 1969; STILLMUNKES and JAMES, 1982; and KOHL et al, 1985) have indicated that soil surface sealing under sprinkler irrigation is related to the kinetic energy of the sprinkler discharge per unit area at the soil surface and to its accumulation in time. Therefore, the distribution of kinetic energy over the wetted area is of interest. KOHL et al. (1985) showed that large nozzle sizes produced greater kinetic energy than smaller sizes. A reducing of the water pressure results in an increase in kinetic energy per unit of discharged water. Figure 4.2 shows the kinetic energy distribution for a sprinkler (KOHL et al., 1985).

When the size distribution of raindrops is known, the total energy is a function of the wind velocity and, generally, of the shape of the wind profile.

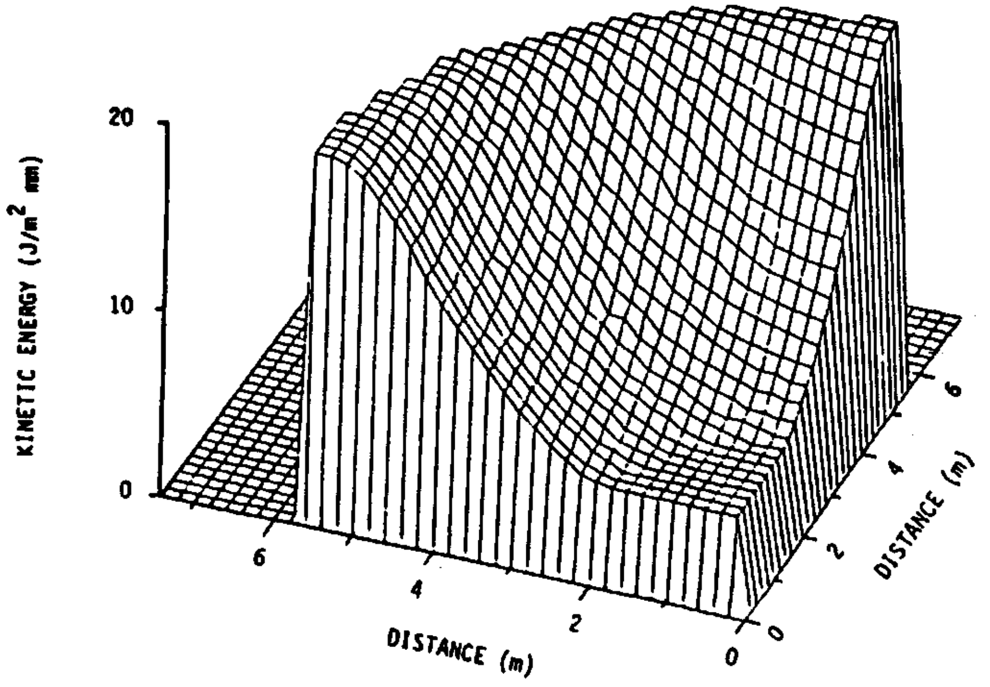


Fig. 4.2 – The distribution of kinetic energy over one–fourth of the pattern of a smooth–plate sprinkler with a 6.4–mm nozzle operated at 100 kPa pressure (KOHL et al., 1985).

In the model, the velocity of each drop was calculated in accordance with the nozzle elevation and the analysis of the velocity history of falling drops (assumed spherical) under the influence of gravity, wind and frictional forces. The kinetic energy was summed for all drops falling on every square element of the ground surface:

$$KE = \frac{1}{2} \sum_{i=1}^{n} m_i \, u_i^2 \tag{23}$$

where n is the number of raindrops impinging at the time interval dt on a section dxdt of the horizontal plane surface, and KE is the kinetic energy on that section (W/m^2).

The intensity of the impinging raindrops is defined by:

$$P_{ef} = \frac{\sum_{i=1}^{n} m_i}{\rho_w \, \Delta t \, \Delta x \, \Delta y} \tag{24}$$

where P_{ef} is the effective precipitation (m/s).

5. INFLUENCE OF SPATIAL VARIABILITY OF SIMULATED RAINFALL ON OVERLAND FLOW

5.1. OVERLAND FLOW

Overland flow is a key process within the hydrologic cycle, and it is very important to people and their environment. Practical applications are found in the design of hydraulic structures, in irrigation, drainage, flood control, water erosion and sediment control, waste water treatment, and environment and wildlife protection.

Not all the rain that reaches the surface is removed by overland flow. Part of it is involved in other processes such as infiltration, interception by vegetation, accumulation in depressions, evaporation, and transpiration. The interactions between these processes are not yet fully understood.

Infiltration is usually the most important conditioning factor of rainfall loss, determining the balance between the gain of soil water and overland flow. A general approach to infiltration problems requires simultaneous solution of the equations describing the process of energy and mass transfer in a complex system embracing all the zones of water movement in the liquid and vapour phases.

In overland flow two, major mechanisms can be discerned. The first, often called Hortonian overland flow, occurs whenever the rainfall intensity exceeds the infiltration rate (CHORLEY, 1978). The second, called saturation overland flow, is produced when the storage capacity of the soil is completely filled, so that all subsequent additions of water at the surface, irrespective of their rate of application, are forced to flow over the surface (KIRKBY and CHORLEY, 1967). The incorporation of these concepts into overland flow modelling is still in progress (KIRKBY, 1988). For example, to describe the unsaturated and saturated zones and the rising water table requires complex models and detailed soil data.

The morphological factors that affect overland flow on slopes are the gradient, length, and shape of the slope, and its exposure to prevailing rain–bringing winds (HOLY, 1980; LIMA, 1988). The degree to which they influence overland flow is closely related to other factors such as soil characteristics, climate, and vegetation (DUNNE and LEOPOLD, 1978).

For the topography, the description of depression storage is important for assessing infiltration and overland flow retardation. The complexity of the surface depression storage and the boundary conditions makes it difficult to find exact solutions to overland flow from hydrodynamics. The infiltration of soils with disturbed surfaces (e.g. tilled soils) is generally poorly understood (MOORE et al., 1980). Interception losses to vegetation can also be significant.

Surface seal development strongly affects infiltration processes (RÖMKENS et al., 1990). The dynamic interaction between solute transport in overland flow and physico–chemical processes can cause a reduction of the infiltration rate on crusted saline and sodic soils and, consequently, increase flow during rainfall (GERITS and LIMA, 1991). The application of overland flow models to these conditions may fail if that dynamic interaction is not considered. Seal development can also be caused by the dispersive and compactive action of raindrop impact and deposition of suspended sediment particles in pores and by filtration (RÖMKENS et al., 1990).

5.2 THEORETICAL DEVELOPMENT

The hydraulic characteristics of overland flow are strongly related to the properties and spatial variability of simulated rainfall. The spatial variability of rainfall intensity, mean drop size, and drop–incidence angles have been combined with an analysis of wind–shear stress at the water–air boundary in order to study the influence of wind on the mechanics of overland flow under spraying systems.

The objective of this chapter is to determine, through a force–balance analysis, the two–dimensional overland flow paths on a plane surface under a full–jet nozzle, with and without wind effects. In this analysis, the total kinetic energy of the falling raindrops is assumed to be dissipated on the overland flow water sheet.

Three forces are involved in every elementary horizontal section ΔxΔy of which the middle point P(x,y,z) lies on the inclined receiving ground surface (Fig. 3.2). These forces are:

(1) Force due to impinging raindrops.

This force can be estimated through the impulse–momentum principle:

$$\vec{F}_d = \frac{\sum_{i=1}^{n} m_i \, \vec{u}_i}{\Delta t} \qquad (25)$$

where n is the number of raindrops impinging at the time interval Δt on an elementary section ΔxΔy, and $\mathbf{F}_d$ is the force due to raindrops (N).

Force $\mathbf{F}_d$ makes an angle α with the vertical and has an azimuth θ (Fig. 4.2), and is assumed to be dissipated on the overland flow water layer.

(2) The tangential wind–shear force.

The length of this force can be expressed in terms of τ by:

$$F_t = \Delta x \; \Delta y \; \tau \qquad (26)$$

where F_t is the tangential wind–shear force (N), and τ is the frictional shear stress (Pa).

If the mean horizontal wind speed increases linearly with the logarithm of the height, then τ is:

$$\tau = \rho_a \left(\frac{w\ K}{\ln(z_1/z_o)}\right)^2 \tag{27}$$

Force $\mathbf{F}_t$ is horizontal and points to the positive X–axis (Fig. 4.2).

(3) The force expressing gravity in an overland flow elementary section.

This force can be expressed by:

$$\vec{F}_1 = \vec{g}\ \rho_w\ \Delta x\ \Delta y\ h \tag{28}$$

where $\mathbf{F}_1$ is the force expressing gravity in an overland flow elementary section (N), and h is the average depth of flow in the elementary section (m).

Force $\mathbf{F}_1$ is vertical. It assumes a uniform overland flow sheet over the entire surface. For simplification, the increase of depth and velocity and related forces due to momentum, pressure, and resistance on the flow, are not considered.

The force obtained from the sum of forces $\mathbf{F}_1$, $\mathbf{F}_t$, and $\mathbf{F}_d$ in every elementary square (ΔxΔy) is vector **F**:

$$\begin{aligned}\vec{F} = F_x\ \hat{i} + F_y\ \hat{j} + F_z\ \hat{k} = (F_t{+}F_d\ \sin\alpha\ \cos\theta)\ \hat{i}\ + \\ +\ (F_d\ \sin\alpha\ \sin\theta)\ \hat{j} + (-F_1{-}F_d\ \cos\alpha)\ \hat{k}\end{aligned} \tag{29}$$

where **i**, **j**, and **k** are the unit vectors along the horizontal axis (X–axis and Y–axis) and the vertical axis (positive upwards).

The length of force **F** is:

$$F = \sqrt{(F_t{+}F_d\ \sin\alpha\ \cos\theta)^2 + (F_d\ \sin\alpha\ \sin\theta)^2 + (-F_1{-}F_d\ \cos\alpha)^2} \tag{30}$$

For the analysis of overland flow, we need to determine the projection of force **F** on the receiving ground surface.

The surface, assumed to be a plane, can be represented by two unit vectors ($\mathbf{i}_1$ along the X1–axis, the interception of the surface with the horizontal plane XY, and $\mathbf{j}_1$ along the steepest line of the surface in the uphill direction) (Fig 5.2 and 5.3):

$$\begin{aligned}\vec{i}_1 &= a_1\,\hat{i} + a_2\,\hat{j} \\ &= \sin(\omega-\Omega)\,\hat{i} - \cos(\omega-\Omega)\,\hat{j}\end{aligned} \tag{31}$$

$$\begin{aligned}\vec{j}_1 &= b_1\,\hat{i} + b_2\,\hat{j} + b_3\,\hat{k} \\ &= \cos\alpha_{sl}\cos(\omega-\Omega)\,\hat{i} + \cos\alpha_{sl}\sin(\omega-\Omega)\,\hat{j} + \sin\alpha_{sl}\,\hat{k}\end{aligned} \tag{32}$$

The azimuth of the steepest line of the slope (uphill direction) is Ω and the azimuth of the wind direction (X–axis) is ω. The angle of the steepest line of the receiving surface with the horizontal is $\alpha_{sl.}$

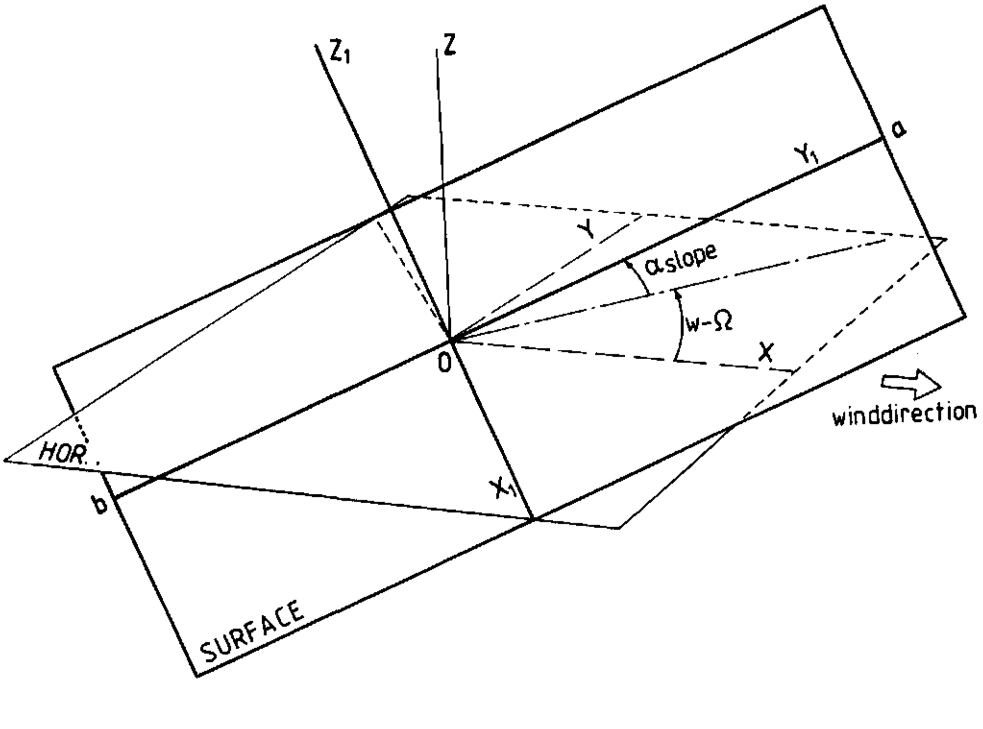


Fig. 5.2 – Planes and axis.

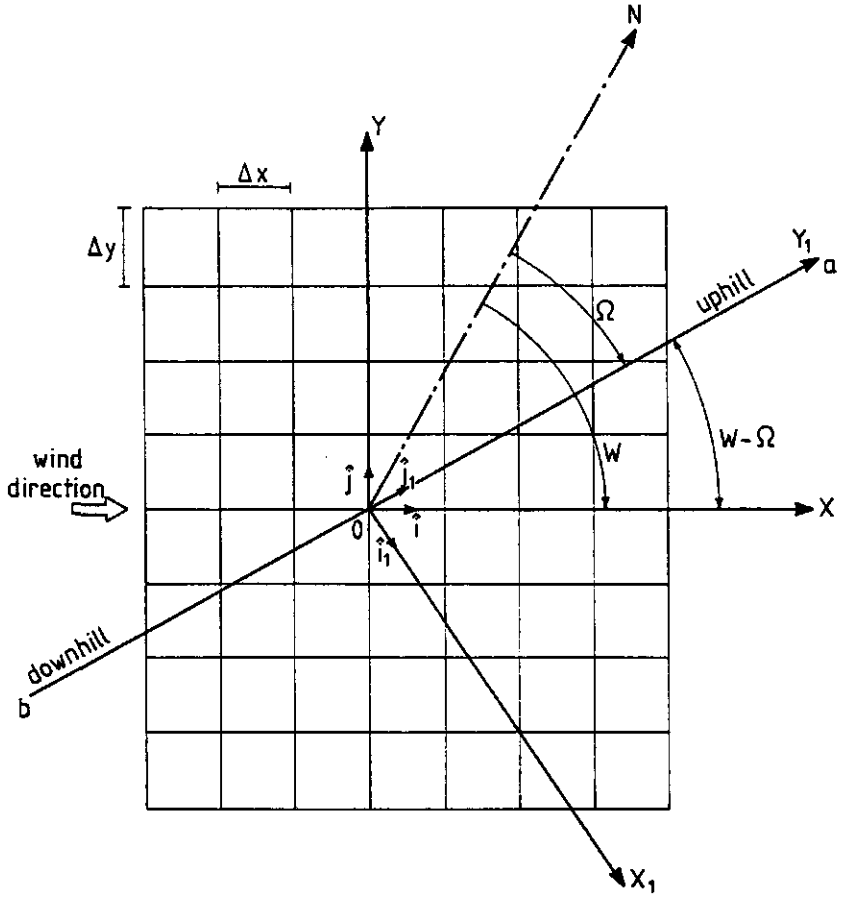


Fig. 5.3 – Top view of axis and grid with elementary sections.

Let $\mathbf{k_1}$ be a unit vector perpendicular to both the X_1–axis and the Y_1–axis of the surface, pointing upwards:

$$\begin{aligned} \hat{k}_1 &= \hat{i}_1 \times \hat{j}_1 = [-\cos(\omega - \Omega)\ \sin\alpha_{sl}]\ \hat{i}\ + \\ &+ [-\sin(\omega - \Omega)\ \sin\alpha_{sl}]\ \hat{j} + \cos\alpha_{sl}\ \hat{k} = \\ &= A\ \hat{i} + B\ \hat{j} + C\ \hat{k} \end{aligned} \tag{33}$$

where × is the cross product of two vectors.

Because the origin (0,0,0) lies on the receiving ground surface, the equation is:

$$Ax + By + Cz = 0 \tag{34}$$

To find the component $\mathbf{F}_{x1}$ of the projection of **F** on the surface, vector **F** will be separated into two components, $\mathbf{F}_{x1}$ and $\mathbf{F}_{aux}$, where the first is to be parallel to the unit vector $\mathbf{i}_1$ and the second is to be perpendicular to $\mathbf{i}_1$:

$$\begin{aligned} &\vec{F} = \vec{F}_{xl} + \vec{F}_{aux} \qquad \textit{with} \\ &\vec{F}_{xl} = c\ \hat{i}_1 \\ &\vec{F}_{aux} \cdot \hat{i}_1 = 0 \end{aligned} \tag{35}$$

where . is the dot product of two vectors. Then, the scalar c is determined by the equation:

$$0 = \vec{F}_{aux} \cdot \hat{i}_1 = (\vec{F} - c\ \hat{i}_1).\hat{i}_1 = \vec{F}.\hat{i}_1 - c \tag{36}$$

or

$$c = \vec{F} \cdot \hat{i}_1 = F_x a_1 + F_y a_2 \tag{37}$$

Then:

$$\vec{F}_{xl} = (F_x a_1^2 + F_y a_1 a_2)\ \hat{i}_1 + (F_x a_1 a_2 + F_y a_2^2)\ \hat{j}_1 \tag{38}$$

Similarly, the component $\mathbf{F}_{y1}$ of the projection of **F** on the surface will be:

$$\begin{aligned} \vec{F}_{yl} &= (F_x b_1^2 + F_y b_1 b_2 + F_z b_1 b_3)\ \hat{i}_1 + \\ &+ (F_x b_1 b_2 + F_y b_2^2 + F_z b_2 b_3)\ \hat{j}_1 + (F_x b_1 b_3 + F_y b_2 b_3 + F_z b_3^2)\ \hat{k}_1 \end{aligned} \tag{39}$$

The coordinate transformation between the two coordinate systems (see Fig. 5.2) will be:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sin(\omega-\Omega) & \cos\alpha_{sl}\cos(\omega-\Omega) & -\sin\alpha_{sl}\cos(\omega-\Omega) \\ -\cos(\omega-\Omega) & \cos\alpha_{sl}\sin(\omega-\Omega) & -\sin\alpha_{sl}\sin(\omega-\Omega) \\ 0 & \sin\alpha_{sl} & \cos\alpha_{sl} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{40}$$

and consequently:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = [A]^{-1} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{41}$$

Matrix **[A]** can be used to calculate the components of vector **F** on the $X_1Y_1Z_1$ coordinate system instead of to derive the projection components of **F** on the surface as described in equations 35 to 39:

$$\begin{bmatrix} F_{x_1} \\ F_{y_1} \\ F_{z_1} \end{bmatrix} = [A]^{-1} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{42}$$

6. COMPUTER PROGRAMS

6.1. INTRODUCTION

In this Chapter, two computer programs are briefly described: DROP and NOZZLE. They were developed in ANSI C and are presented in Appendix I. The programs listed in this report are provided "as is", without warranty of any kind.

6.2. DROP

The purpose of DROP (Appendix I) is to calculate the trajectory of a drop once catapulted from a nozzle. The drop is acted upon by drag forces, by wind, and by gravity.

A. Data structure

The data are subdivided into classes. First we have the global input variables. Then there is the structure defining the state of a drop. Finally there are global data that are calculated from the input variables. These data can be found in files "*drop.h*" and "*drop.c*" (Appendix I).

B. Function structure

In file "*drop.c*" there are three basic functions, which are prototyped in "*drop.h*":

(1) *initiate_systemconstants*

This function calculates the variables that are constant for each trajectory. Hence, this function has to be called once for every trajectory.

(2) *make_first_step*

This function calculates the data describing the initial state of the drop from the global variables given. Hence, this function also has to be called once for every trajectory.

(3) *do_step*

This function calculates the state of the drop one time–step later.

In "*drop.c*" there is one extra function: *complete_state*, which completes the calculation of a new state, once x, y, z, V_x, V_y and V_z have the right value. It is called by *make_first_state* and *do_step*.

C. The driver program

The driver program is listed in "*do_traj.c*". It reads in the input, does the calculations of the trajectories, and writes the results to a file. It is called by:
do_traj name_input_file name_output_file

C.1. Structure of the input file

The listing "*indrop*" contains a typical example of an input file for the calculation of four trajectories. The driver program *do_traj* scans the input file for an equal sign and reads a number. So, all the text before an equal sign is irrelevant. First the geometrical information is given, then the physical information, the time steps and finally, for each trajectory to be calculated, the necessary values for the calculation of the initial state.

C.2. Structure of the output file

The structure of the output file is very sober: for each trajectory and for each time step the position of the drop is given through the coordinates t, x, y, z, and V_{drop}. Two trajectories are separated by a blank line. The listing "*outdrop*" contains a typical example of an output file.

6.3. NOZZLE

The purpose of NOZZLE (Appendix I) is to calculate the effect of the drop hits from a single full–cone spray nozzle on water application, on kinetic energy, and on overland flow. For this, the plane is subdivided into rectangles, and for each rectangle a number of statistics are calculated.

A. Data structure

The data are subdivided into classes. These data can be found in files "*nozzle.h*" and "*nozzle.c*" (Appendix I):

(i) "*nozzle.h*":

For the description of the nozzle, a number of variables are needed: the discharge of the nozzle, the speed of drop release by the nozzle (which is assumed to be constant), the spray angle (which is the maximum angle with the vertical of a drop released by the nozzle, assumed to be the same in all directions), and the statistical description of drop diameter released by the nozzle.

With respect to the distribution of drops produced by the nozzle, drop diameters (the quantiles) are required to describe the statistics (see also Fig. 6.1). If, for example, 4 diameters are specified (q_1,...,q_4), ⅓ of the drops will fall between q_1 and q_2, ⅓ will fall between q_2 and q_3, and ⅓ will fall between q_3 and q_4 (see also Fig. 6.1). In this way, available data as in Table 2.1 (page 10) can be incorporated into the program.

To gather statistical information in the plane, one needs the minimum and maximum values of x and y (e.g. the field plot size) and the number of subdivisions in the X and Y directions. With this information, the selected area is regularly subdivided into rectangles.

Because rainfall simulations are limited to short time periods, the number of drops to be generated from the nozzle is also necessary.

(ii) "*nozzle.c*":

A few variables are declared in "*nozzle.c*". First, the total volume of water sprayed during the simulation. Then, the typical size of a rectangle in the plane in the X and in the Y directions (*step_x* and *step_y*). Finally, a number of two–dimensional arrays to hold the statistical information gathered (e.g., *stat_diameter* is used to calculate the average diameter in each rectangle in the selected area).

B. Function structure

In file "*nozzle.c*" there are several basic functions implemented. These are:

(1) *random_initials*

This function generates a "random" drop release from the nozzle. For this, the function assumes a random number generator with a uniform deviate, generating random numbers between 0 and 1 (file "*uniform.c*"). The initial conditions for each drop are determined as follows:

(i) θ_0 is chosen, spaced uniformly between 0 and 2π;

(ii) α_0 is chosen, spaced uniformly in 0 and α_{max};

(iii) the diameter is chosen with an inverse cumulative distribution function approach. Figure 6.1 illustrates this technique for four quantiles.

The function *random_initials* should be adapted to the type and caracteristics of the nozzle spray (see Chapter 2).

(2) *initiate_statistics*

This function dynamically allocates space for the arrays with the statistical information.

(3) *calc_planehit*

This function calculates the exact position, in plane coordinates, of each drop hit on the plane surface.

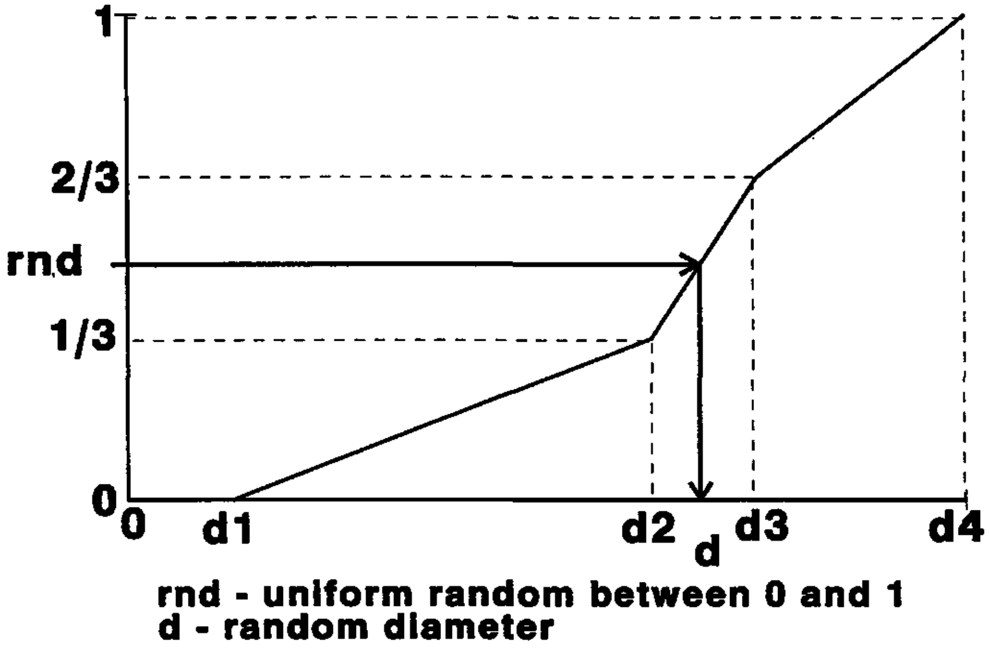


Fig. 6.1 - The inverse cumulative distribution function approach.

(4) *to_tally*

Function *to_tally* updates the statistics whenever a new hit has been calculated.

(5) *report_tally*

Function *report_tally* writes the results of the statistical information to a file (see also output description, further on).

C. The driver program

The driver program, listed in "*do_spray.c*" combines all functions to simulate the spray from the nozzle. It generates initial conditions for a drop (through *random_initials*) and then calculates the drop trajectory (using the functions in *drop.c*) until the drop hits the plane. When enough drops have been simulated, and the statistical information has been processed, the results are written to output files.

The program can be executed with a number of options. To implement this, the files "*option.h*" and "*option.c*" are used.

The command line for *do_spray* is:

do_spray name_input_file [-s name_scalar_file] [-v name_vector_file] [-h name_hit_file]

(i) an input file has to be given;

(ii) if the user wants a file that describes all scaler information, he has to add the option "*-s*" and the name of the file to which this information should be written;

(iii) if the user wants a file that describes all vector information, he has to add the option "*-v*" and the name of the file to which this information should be written;

(iv) if the user wants a file that describes all the positions where the drops hit the plane, he has to add the option "*-h*" and the name of the file to which this information should be written.

C.1. Structure of the input file

The listing "*innozzle*" contains a typical example of an input file. The program scans the input file for an equal sign and then reads a number, in the same way as for *do_traj*. So, all the text before an equal sign is irrelevant. Data of the input file, having always the same order as presented in file "*innozzle*", consist of: geometrical information, physical information, description of nozzle and spray pattern (e.g., accurate drop–size distribution), information for computations, and information for the output.

C.2. Structure of the scalar output file

The scalar output file contains all scalar statistical information. Every line contains information for one rectangle of the plane. First the x and y coordinates of the middle point of the rectangle, then the mean drop diameter, the drop intensity, the mean volume intensity, and, finally, the mean kinetic energy. The listing "*scalar*" contains a typical example of an output file.

C.3. Structure of the vector output file

The vector output file contains all vector information. The information on the constant vector fields (gravity force field and wind–shear stress force) is given first.

Every line contains information for one rectangle on the plane. First the x and y coordinates of the middle point of the rectangle, then the number of drops, then the x and y components of the force induced by the rain, and finally the components of the total force. The listing "*vector*" contains a typical example of an output file.

C.4. Structure of the hit output file

The file "*hit*" contains all x and y coordinates on the plane corresponding to the points where drops hit the surface.

7. RESULTS

7.1. DROP MOVEMENT

The effect of wind on drop trajectories catapulted from a nozzle is shown in Fig. 7.1 (two–dimensional trajectories, side view), and in Figs. 7.2 and 7.3 (three–dimensional trajectories, top view and perspective view). See Chapters 3 and 6 for a description of DROP.

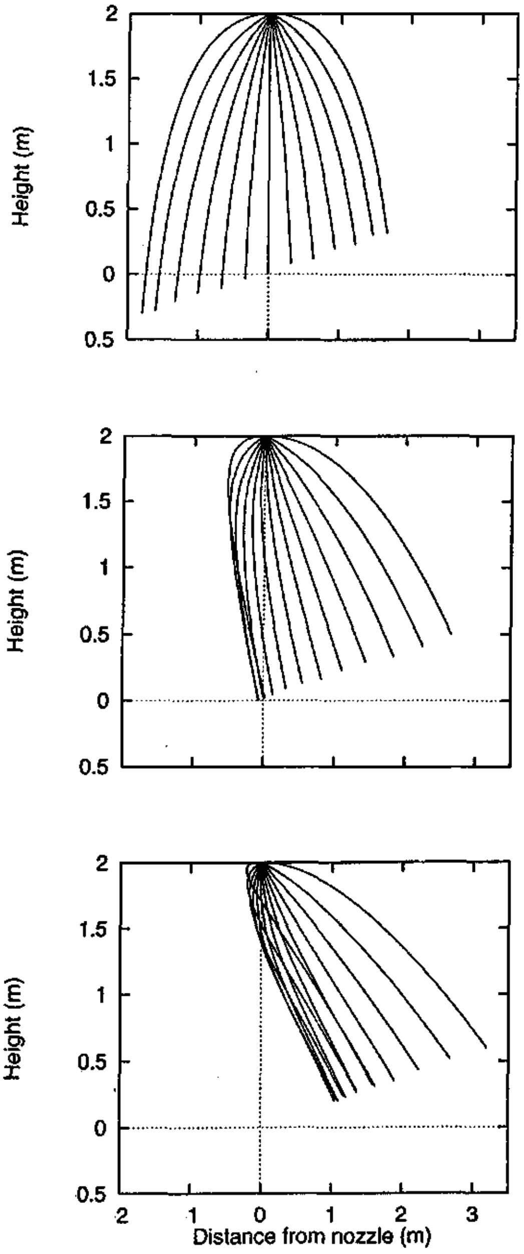


Fig. 7.1 – Two–dimensional drop paths of a 1–mm drop ejected in different directions with a velocity of 5 m/s. Top: without wind; Centre: with V_{wind} = 3.5 m/s blowing upslope; Bottom: with V_{wind} = 7 m/s, also blowing upslope. The slope of the receiving plane is 10%.

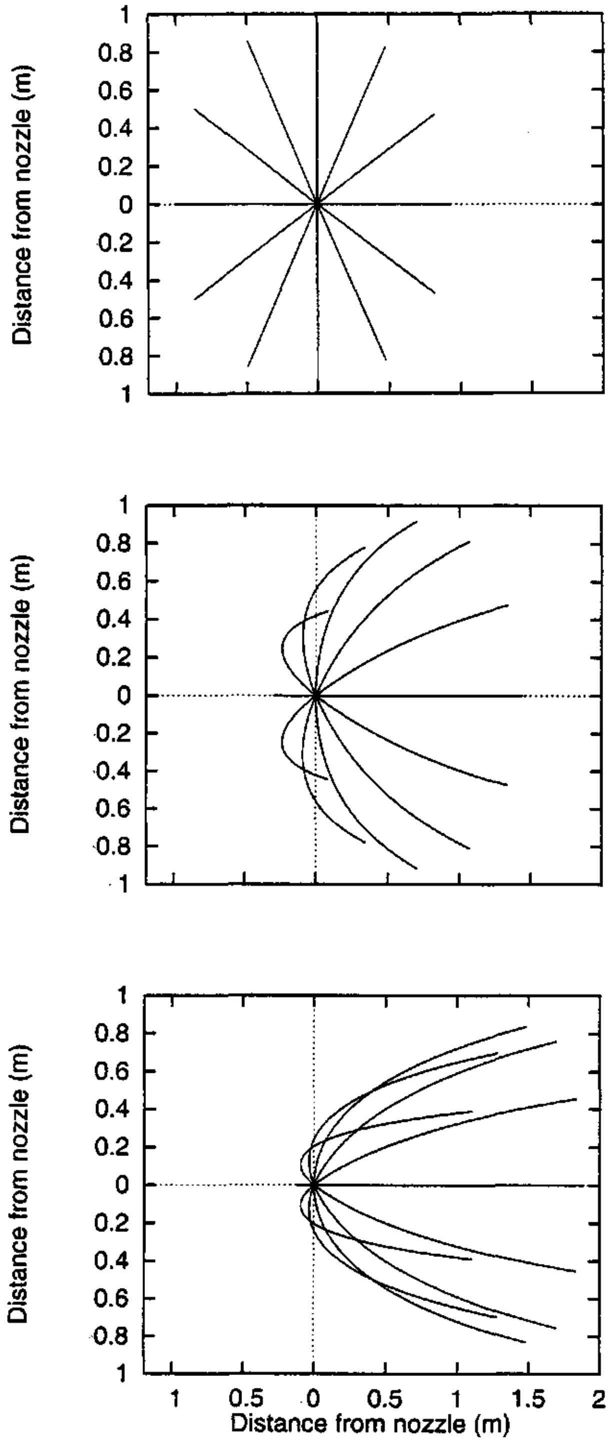


Fig. 7.2 – Top view of three-dimensional drop paths of a 1-mm drop ejected in different directions at a velocity of 5 m/s. Top: still air; Centre: with V_{wind} = 3.5 m/s blowing upslope; Bottom: with V_{wind} = 7 m/s, also blowing upslope. The slope of the receiving plane is 10%.

Drops that are sprayed into the wind do not go as far as drops sprayed downwind. Those that are sprayed crosswind have a small reduction of distance of throw crosswind, but this is translated downwind.

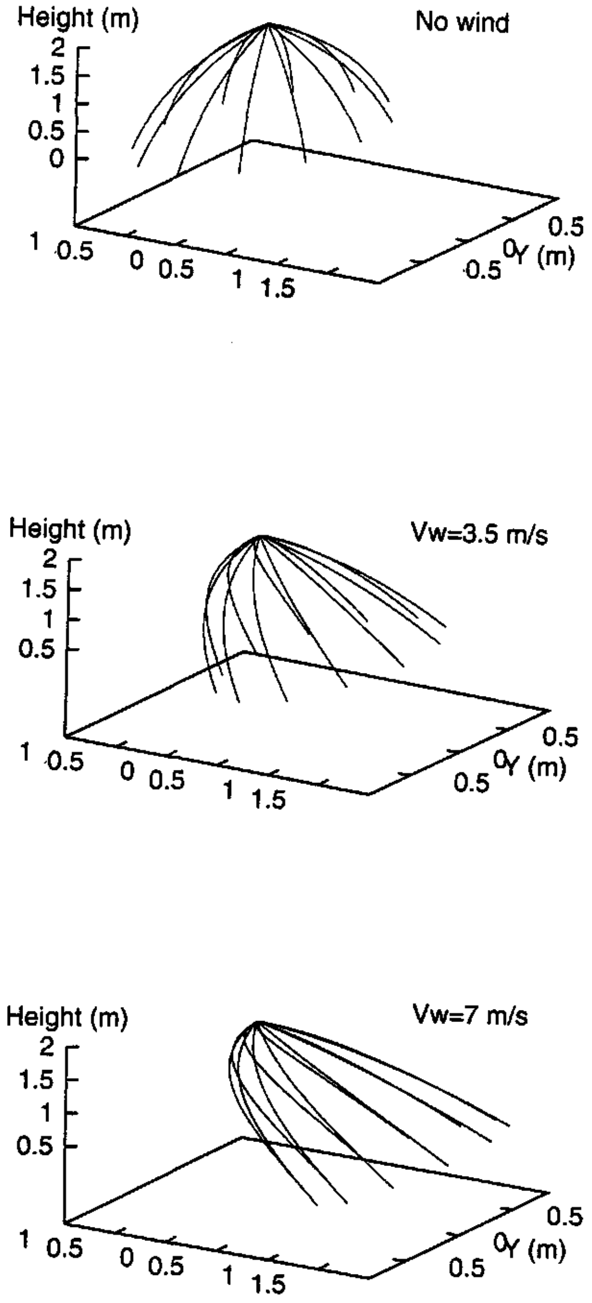


Fig. 7.3 – Three-dimensional drop paths of a 1-mm drop ejected with a velocity of 5 m/s in different directions. Top: still air; Centre: with V_{wind} = 3.5 m/s blowing upslope; Bottom: with V_{wind} = 7 m/s, also blowing upslope. The slope of the receiving plane is 10%.

The greater effect of the wind on smaller drops is due primarily to greater drag (Fig. 7.4). Because smaller drops have a lower fall velocity, they are more time–subjected to the wind action (Figs. 7.4 and 7.5).

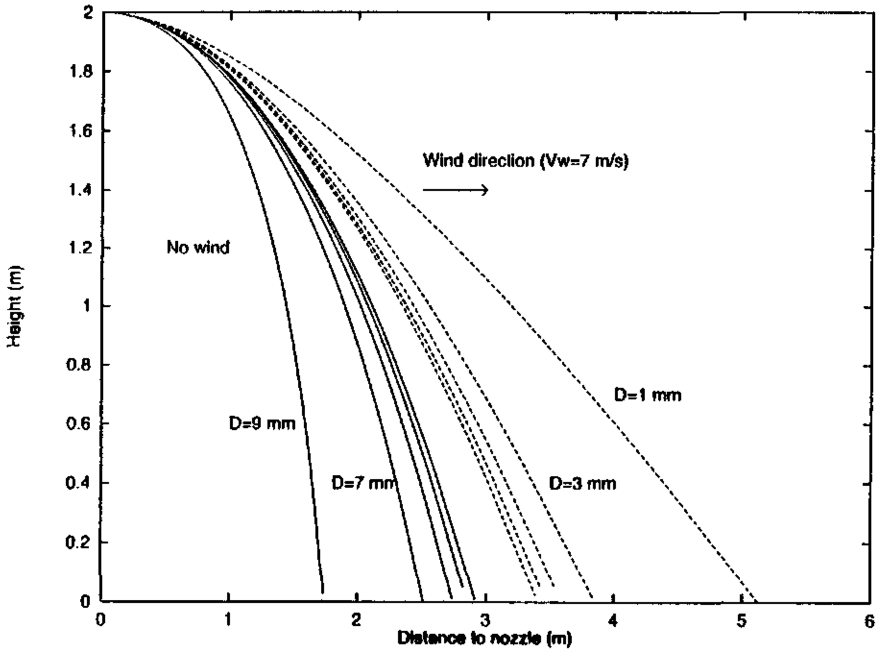


Fig. 7.4 – Trajectories of different drops (diameters: 1, 3, 5, 7 and 9 mm) in still air (solid lines) and in wind (V_{wind}=7 m/s: dashed lines).

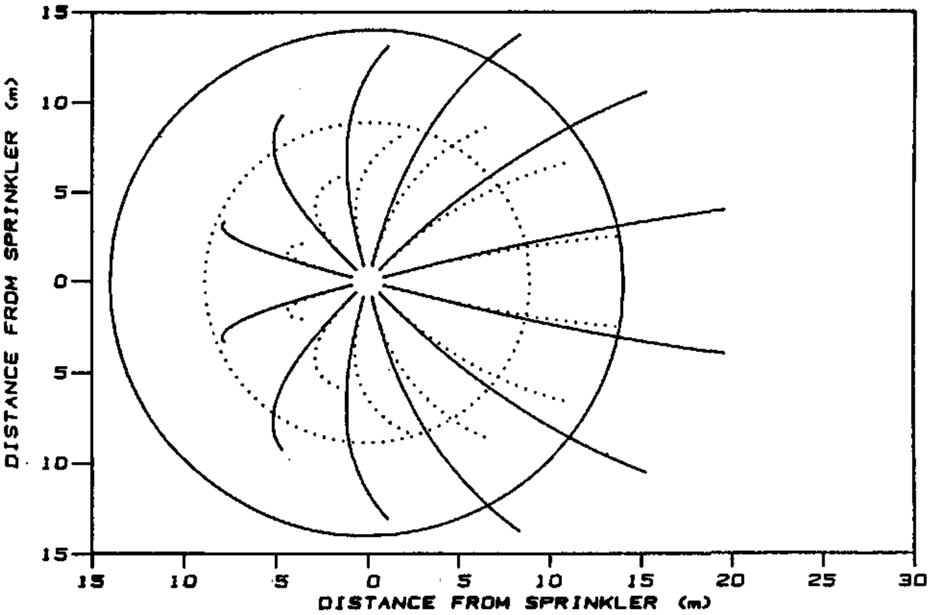


Fig. 7.5 – 1.7–mm (dotted lines) and 3.4–mm (solid lines) drop paths in a constant 4 m/s wind from a 25° trajectory sprinkler. Circles are the still–air radius (BERNUTH, 1988).

Drops that are catapulted at high speeds are subject to deceleration due to drag forces. Those released from the nozzle at low speeds accelerate with gravity (Figs. 7.6 and 7.7).

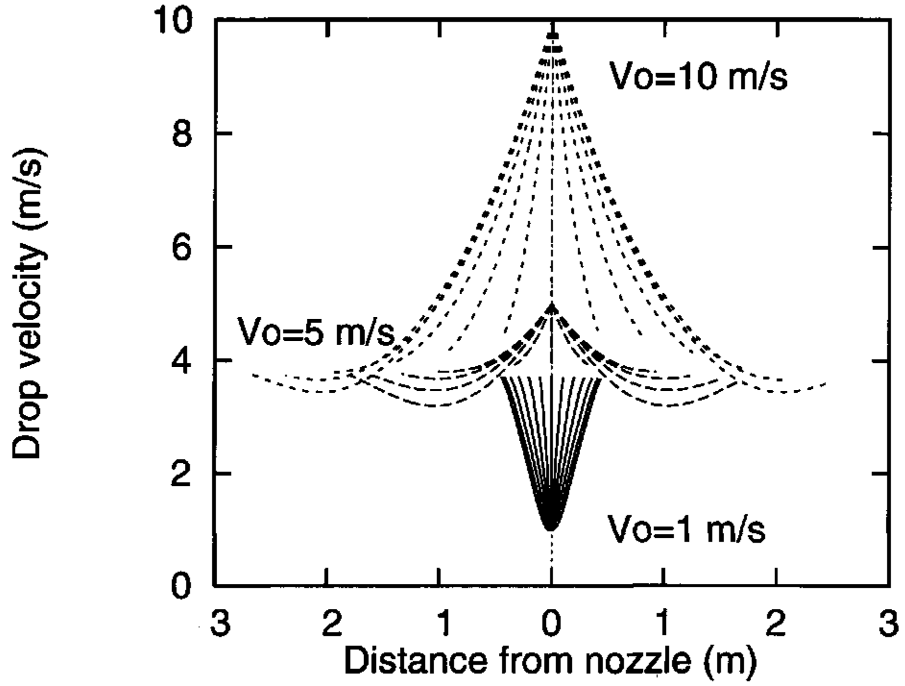


Fig. 7.6 – 1–mm drop velocity at distance from nozzle, for three different nozzle released velocities: Vo=1, 5, and 10 m/s. The slope of the receiving plane is 10%.

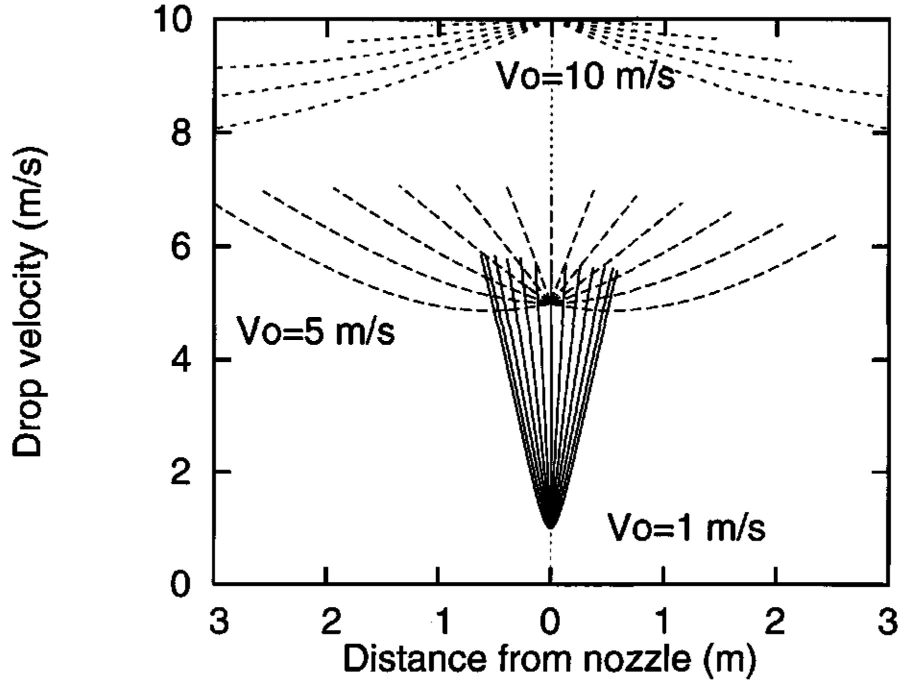


Fig. 7.7 – 7–mm drop velocity at distance from nozzle, for three different nozzle released velocities: Vo=1, 5, and 10 m/s. The slope of the receiving plane is 10%.

Drop velocities over time are strongly affected by wind (Fig. 7.8). The greatest variation in velocity occurs when the drop is ejected into the wind and when the drop is airborne the longest time (about 0.75 s for a headwind of 7 m/s).

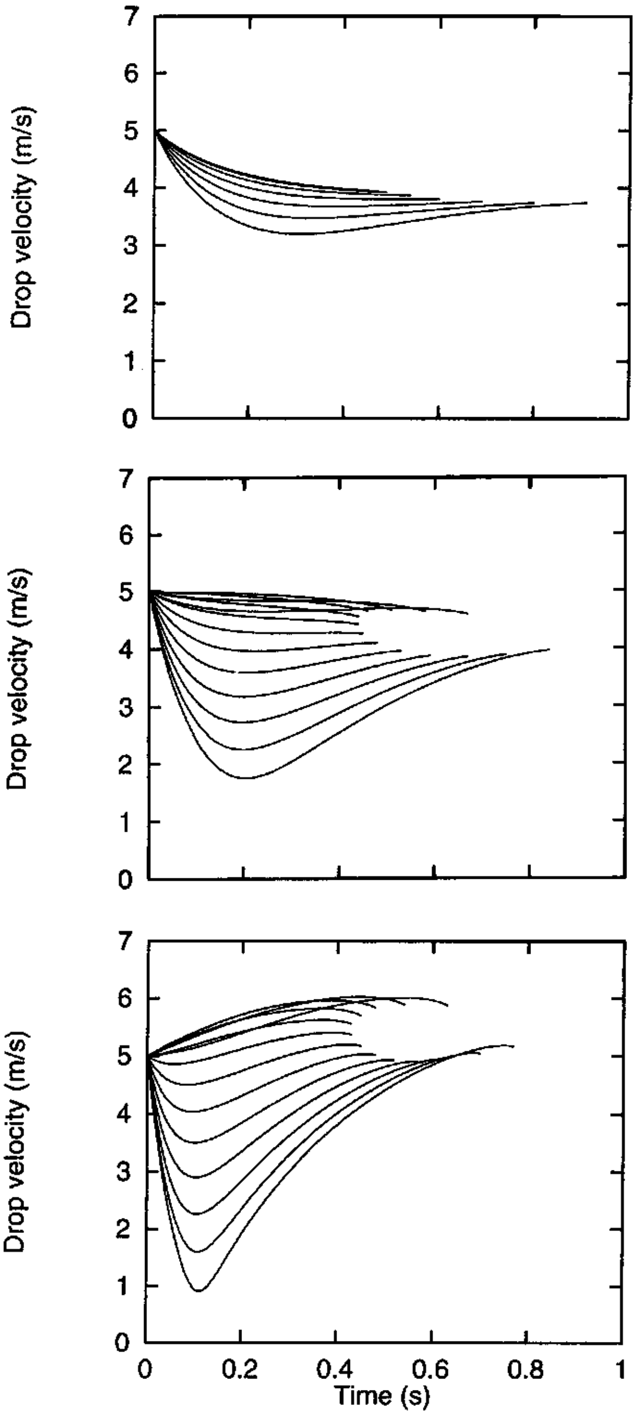


Fig. 7.8 – A 1-mm drop ejected with a velocity of 5 m/s in different directions (Fig. 7.1 shows the corresponding two-dimensional drop paths). Top: without wind; Centre: with V_{wind} = 3.5 m/s blowing upslope; Bottom: with V_{wind} = 7 m/s, also blowing upslope. The slope of the receiving plane is 10%.

The effect of different wind speeds is shown in Fig. 7.9. Wind speed was increased from still–air to 21 m/s headwind (at 10 m) for a drop released horizontally against the wind direction (V_0=5 m/s). At high wind speeds, the drop path actually curls back over itself. This occurs when the horizontal velocity is reversed by wind drag before the vertical velocity is overcome by gravity. As the speed (headwind) increases, the travelling time also increases.

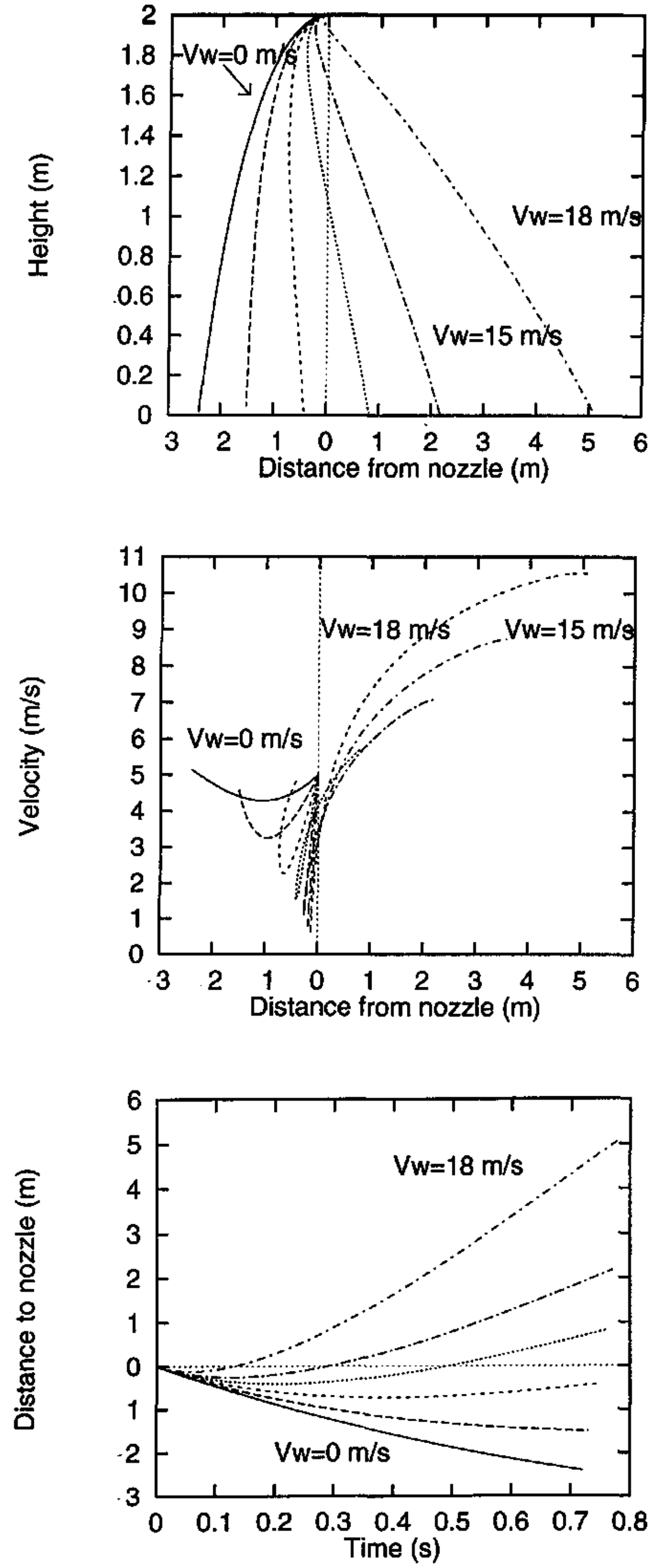


Fig. 7.9 – Effect of different wind speeds on a 2.5–mm drop. Wind speed was increased from still–air to 18 m/s headwind (at 10 m). Top: two-dimensional drop paths; Centre: drop velocity against travel distance; Bottom: travel distance over time.

7.2. WATER APPLICATION AND KINETIC ENERGY

Kinetic energy and water application distributions under a nozzle, both in still-air and in wind, are normally highly concentrated (see Appendix 2). Fig 7.10 shows the spray patterns of 20 000 drops that hit a 30% sloping plane under different wind conditions. An example of kinetic energy distributions for an increasing downslope wind is shown in Fig. 7.11. Figure 7.12 illustrates the effect of wind direction on kinetic energy distributions. In all cases, there is a strong concentration of water application and kinetic energy below, or close below, the nozzle.

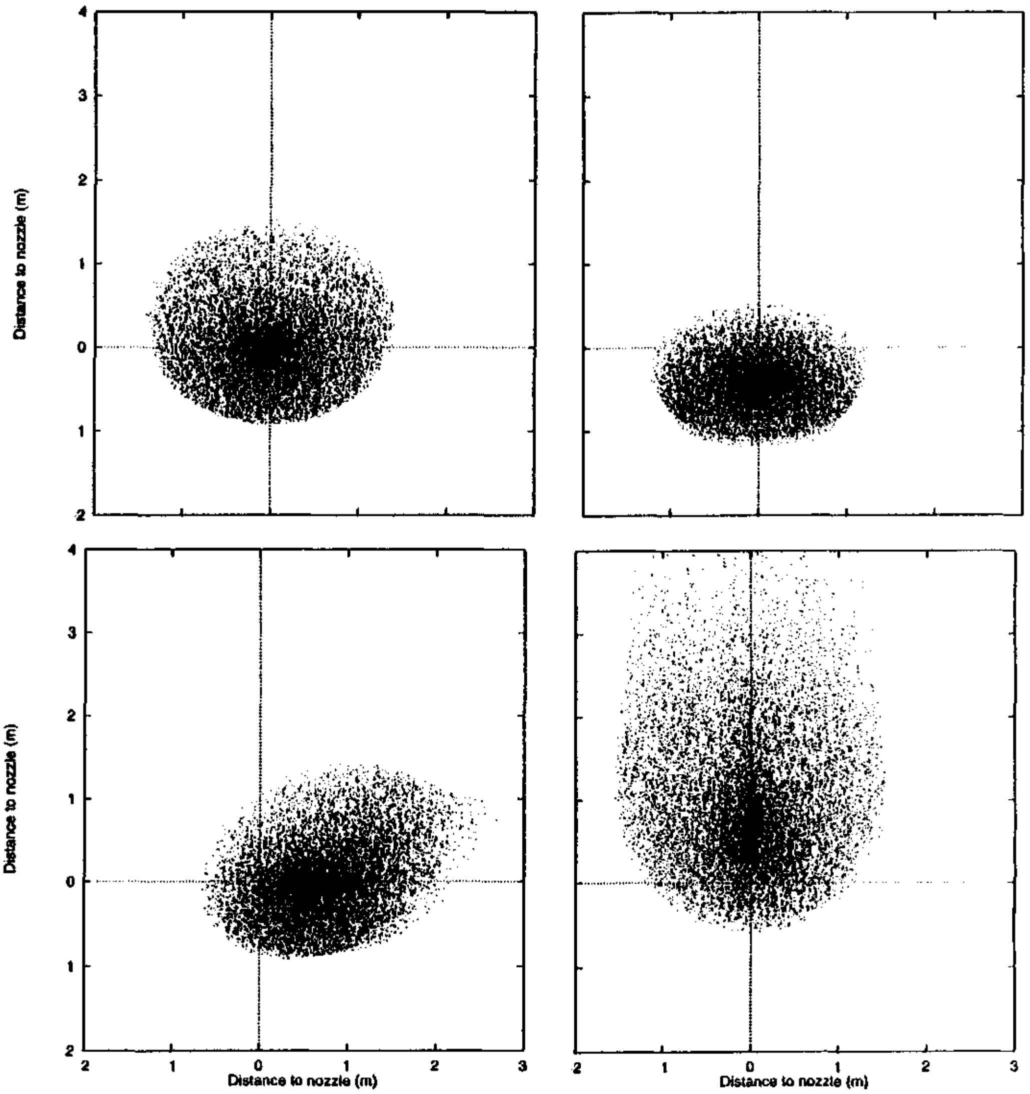


Fig. 7.10 – Spray patterns of 20 000 drops on a 30% slope: Top left: still air; Bottom left: wind blowing from the left; Bottom right: wind blowing downslope; Top right, wind blowing upslope. The maximum spray angle is 100 deg (see Fig. 2.1).

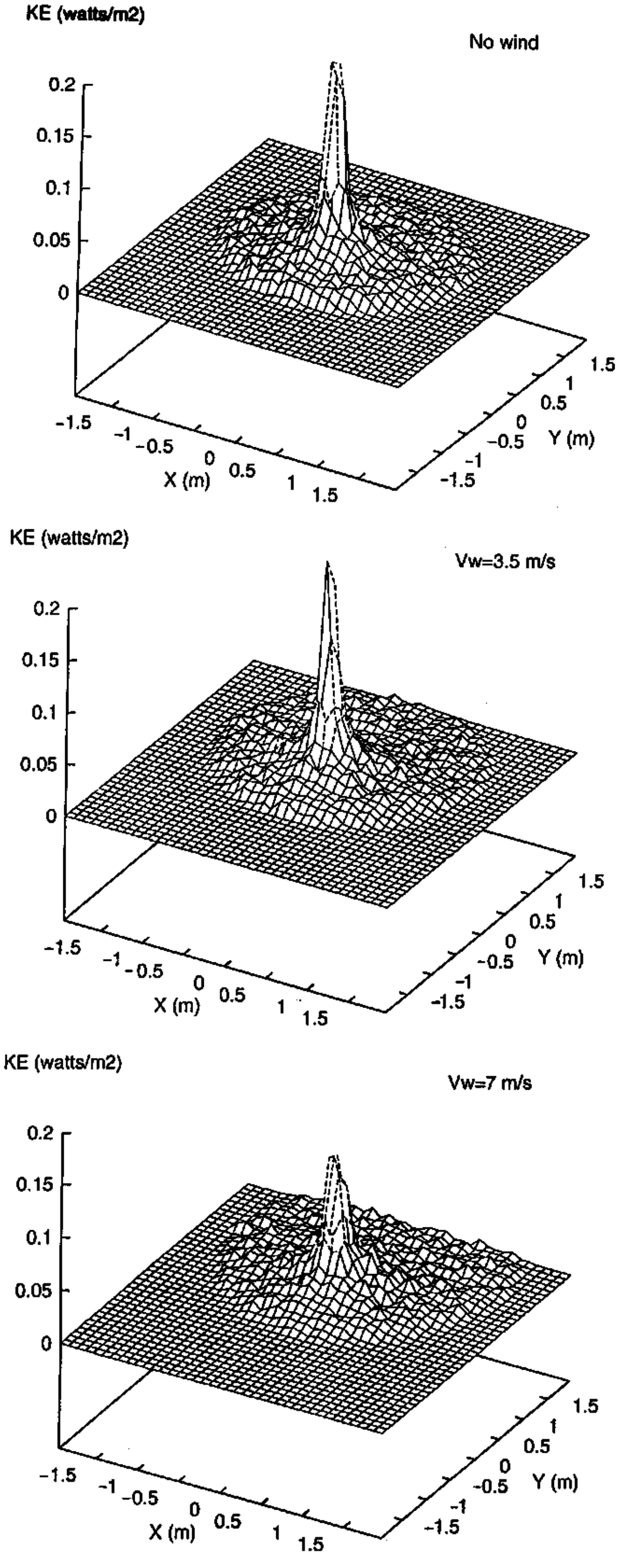


Fig. 7.11 – Kinetic energy under a single full–cone nozzle in an increasing downslope wind. Top: still air; Centre: with V_{wind} = 3.5 m/s blowing upslope; Bottom: with V_{wind} = 7 m/s, also blowing upslope. The slope of the receiving plane is 10% and the maximum spray angle is 100 deg (see Fig. 2.1).

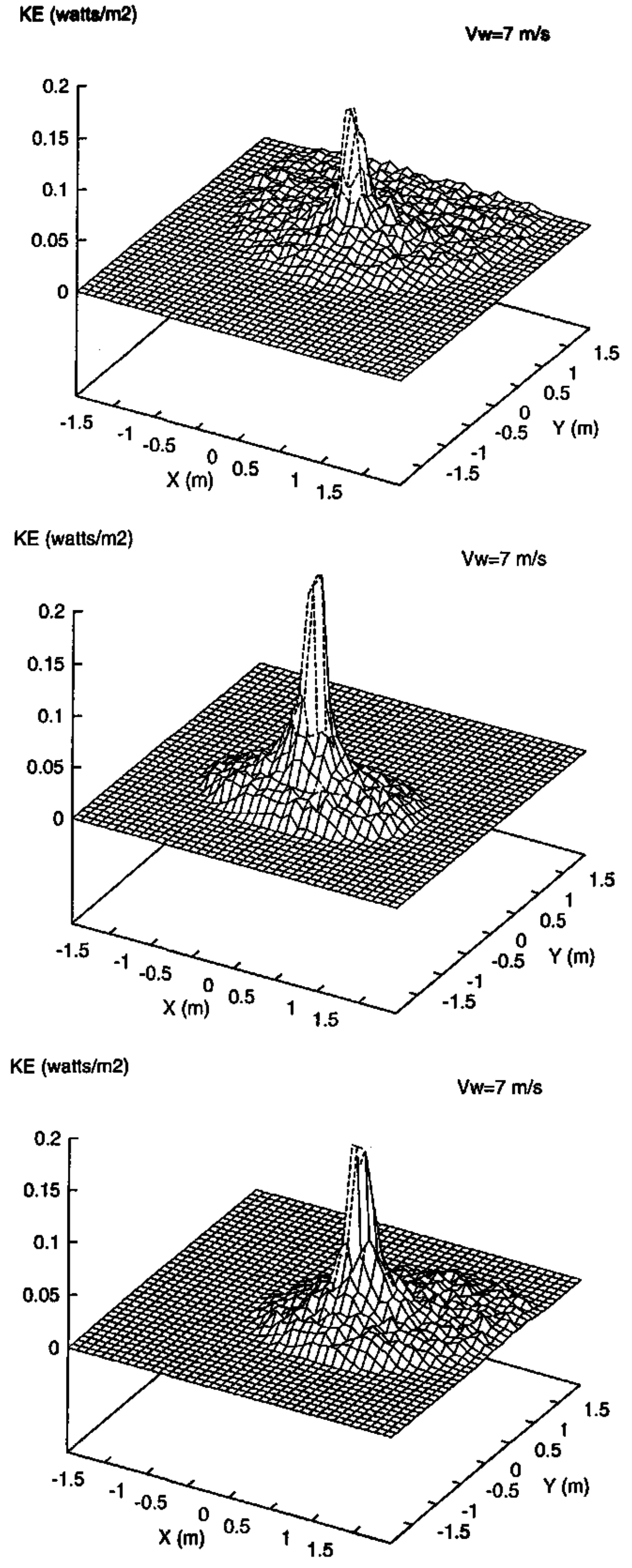
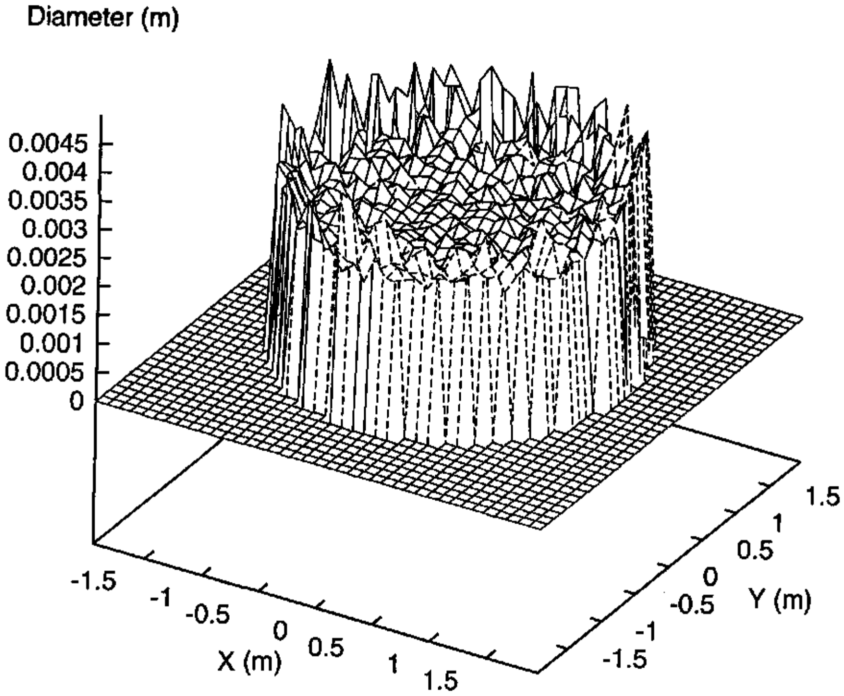


Fig. 7.12 – Kinetic energy under a single full-cone nozzle in different wind directions. Top: with wind blowing downslope; Centre: with wind blowing upslope; Bottom: with wind blowing from the left. The wind speed is $V_{wind} = 7$ m/s and the slope of the receiving plane is 10% and the maximum spray angle is 100 deg (see Fig. 2.1).

Because the effects of wind are greater on smaller drops, the distribution of drop diameters is quite different for still–air and windy conditions (Fig. 7.13).



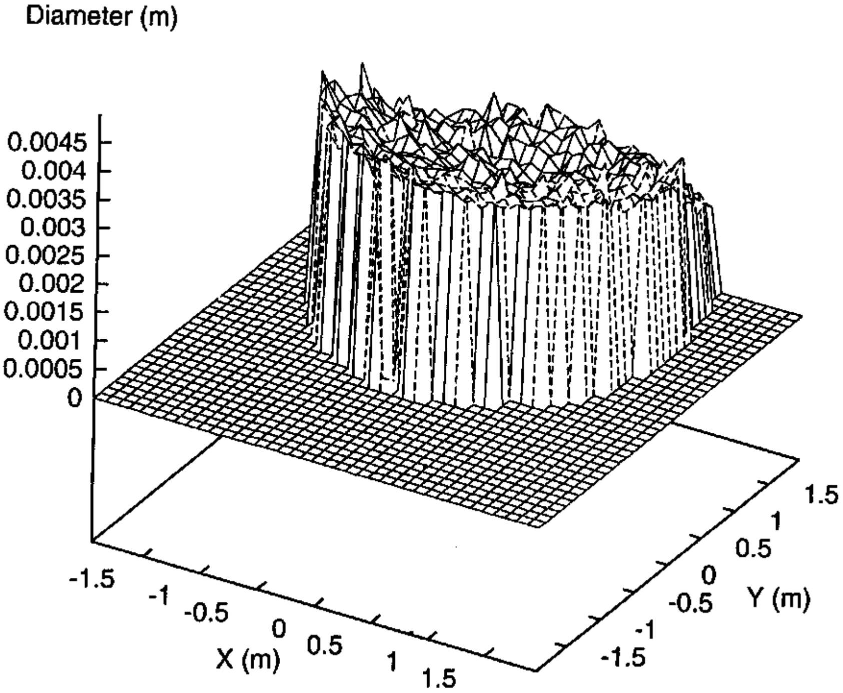


Fig. 7.13 – Diameter distributions under a single full–cone nozzle for an increasing downslope wind. Top: still air; Bottom: with V_{wind} = 7 m/s blowing downslope. The slope of the receiving plane is 10% and the maximum spray angle is 100 deg (see Fig. 2.1).

7.3 OVERLAND FLOW

Figures 7.15 and 7.17 give the spatial variation of the field for the overland flow driving force, which allows comparisons between different wind speeds and directions. Figure 7.14 shows the two simulation situations (right and left of Figs. 7.15 to 7.17). The length and directionality of the vectors were based on vector and trigonometrical analysis of the force balance for every elementary area of the surface plane.

In windless conditions (Fig. 7.15), overland flow can not be considered straight because of the angles of incidence on the receiving surface.

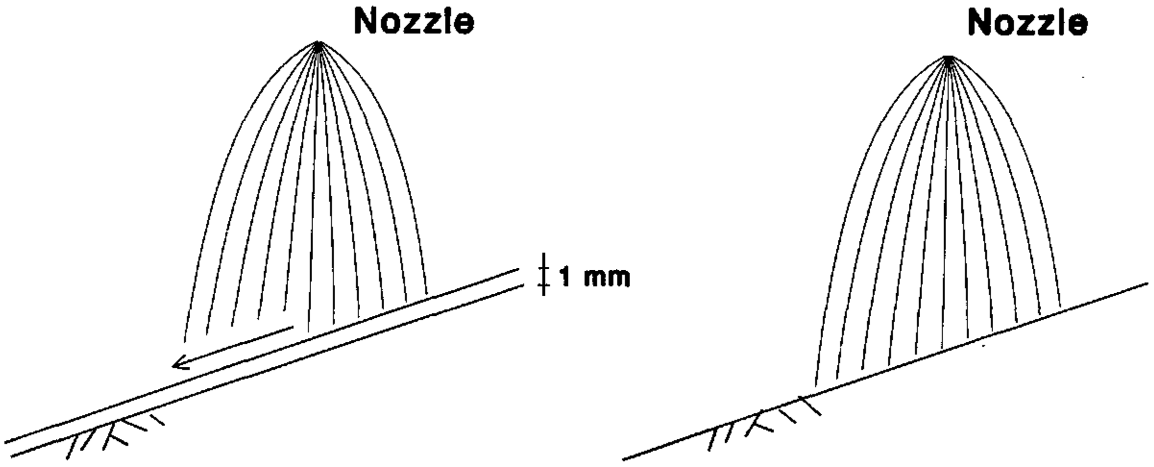


Fig. 7.14 – Simulation situations. Right: with only forces due to drop impact; Left: considering wind shear stress and an overland flow water depth of 1 mm.

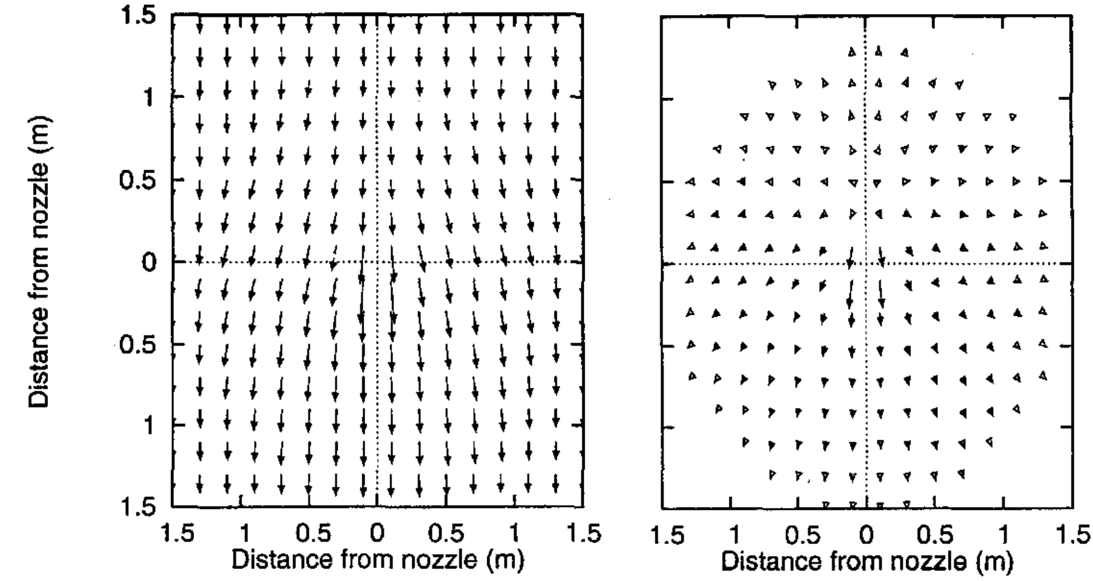


Fig. 7.15 – **Overland flow driving force field in windless conditions. The nozzle is located at 2 m height at (0,0).** Right: with only forces due to drop impact; Left: with wind-shear stress and an overland flow water layer of 1 mm. The slope of the receiving plane is 10% and the maximum spray angle is 100 deg (see Fig. 2.1).

The non–uniformity of rainfall intensity and drop incidence angles, combined with the tangential wind–shear stress in the water–air boundary, affected the driving–force balance and changed the downslope and upslope hydraulic characteristics of overland flow. Consequently, flow velocities and depths both will be affected.

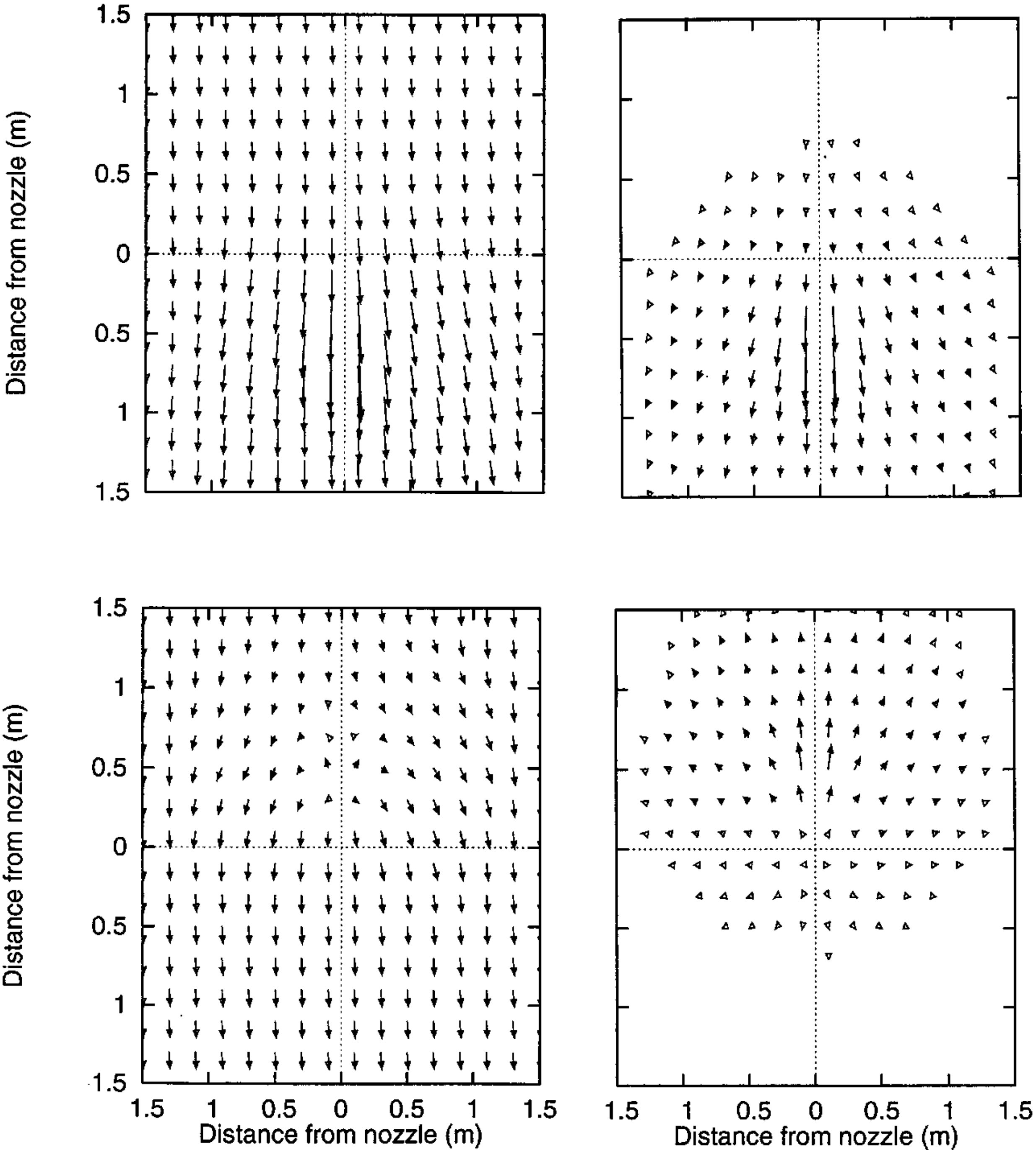


Fig. 7.16 – Overland flow driving–force field in wind. The nozzle is at 2 m height at (0,0). Top: with wind blowing downslope; Bottom: with wind blowing upslope; Right: with only forces due to drop impact; Left: with wind–shear stress and an overland flow water layer of 1 mm. The slope of the receiving plane is 10% and the maximum spray angle is 100 deg (see Fig. 2.1).

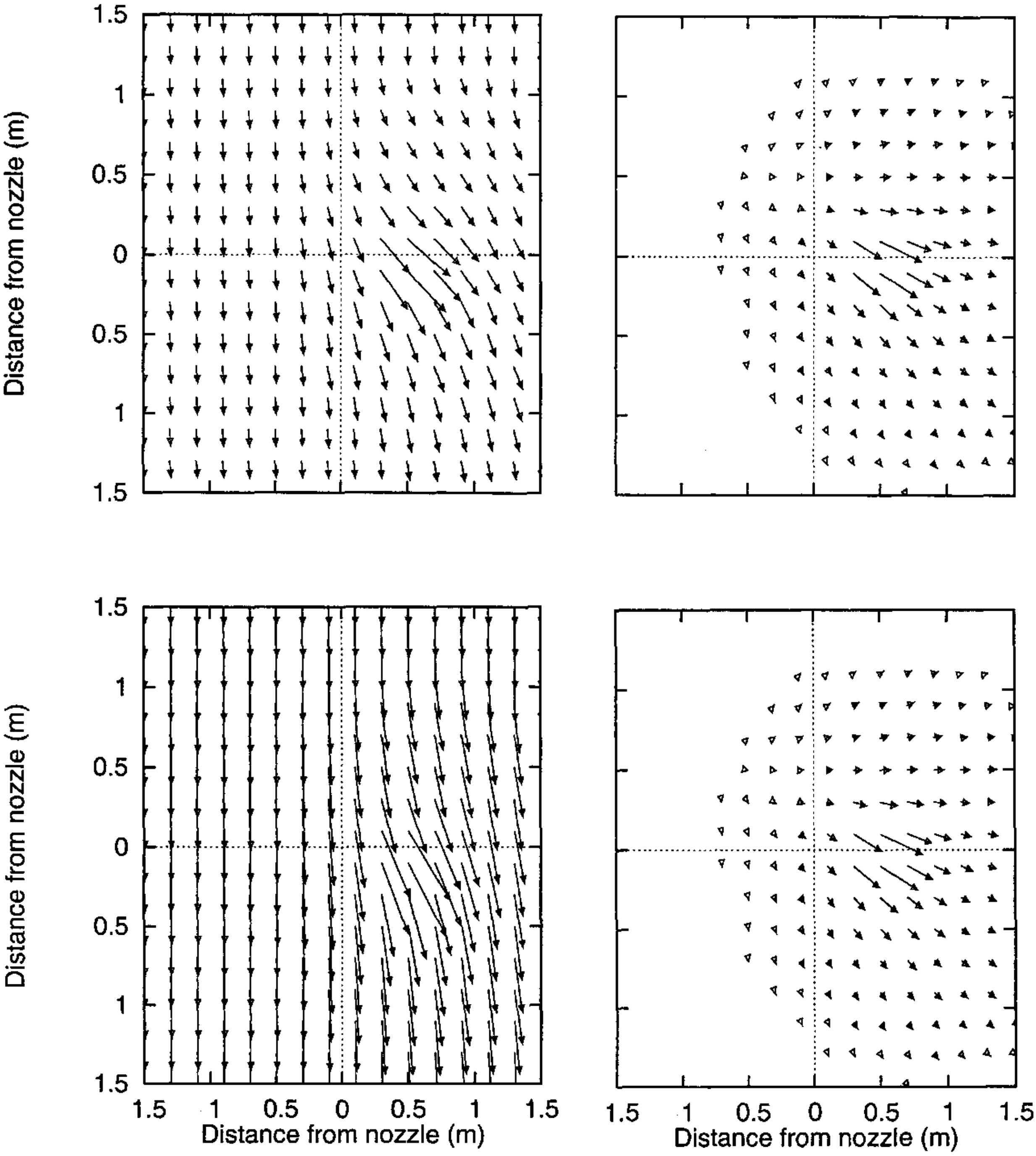


Fig. 7.17 – Overland flow driving-force field in lateral wind. The nozzle is at 2 m height at (0,0). Top left: with a water layer of 1 mm; Bottom left: with a water layer of 3 mm; Right: with only forces due to drop impact. The slope of the receiving plane is 10% and the maximum spray angle is 100 deg (see Fig. 2.1).

8. CONCLUSIONS

The effect of wind on nozzle sprays and underlying overland flow was investigated. Two computer programs were developed: DROP, which calculates the trajectory of drops once catapulted from a nozzle, and NOZZLE, which calculates the effect of the drop hits released from a single full–cone spray nozzle on water application, on kinetic energy, and on overland flow. Results showed that wind has a significant effect on rainfall simulations.

Physically based mathematical models for nozzle sprays, that account for all the processes involved and their interactions, can never be made. There are limitations owing to the assumptions underlying the theoretical developments. Simplifying of assumptions, however, should assure that the model still retains the basic characteristics of the physical system being modelled. The numerical computer models proposed in this report require specification of the nozzle characteristics, wind conditions, and surface properties.

The following conclusions can be drawn:

(i) The distribution of water application and kinetic energy over the wetted area is very important in many studies. Water application and kinetic energy distributions can be accurately studied by coupling a hydrodynamic model for drop movement and an appropriate stochastic drop generator.

(ii) The pattern of water application under a nozzle is distorted by wind. Because windless applications are already non–uniform, any further distortion can lead to still lower uniformity coefficients. If application is deficient, lower results can be expected from the rainfall simulation. Also, drop size distributions with and without wind are considerably different, because the trajectories of smaller drops are more affected by wind than those of bigger drops.

(iii) The non–uniformity of simulated rainfall (e.g. rainfall intensity pattern and drop incidence angles) induced by wind can significantly affect the hydraulics of underlying overland flow. Wind causes a change in the driving force of overland flow by affecting drop trajectories and impact velocities and through tangential shear stress in the overland flow water–air boundary, by affecting water depths and overland flow velocities. Wind also affects the shape and size of drops and their splash shapes, which appears to have an effect on the flow resistance of overland flow (LIMA, 1989b and d). These cases were not investigated here.

(iv) Rainfall simulation experiments should be supplemented with information about the wind characteristics. This is particularly important in field experiments in regions where rainfall is accompanied by strong winds.

(v) Because the models presented in this report have no fitted parameters, they are especially helpful in the selecting of nozzle types (spray angles, ejection velocities, and distribution of drop sizes), to determine the size and configuration of the spray area for the expected field situation (rainfall characteristics, wind conditions, slope of terrain, and plot size). It provides a simple way of visualizing spray pattern and overland flow on different sloping surfaces and for different wind conditions.

(vi) While manufacturers offer a wide variety of nozzles with a variety of spray angles, one should consider drop–size distributions and expected wind speeds before choosing which spray angle to use.

(vii) Many rainfall simulators with multiple nozzles are purported to generate a uniformly distributed rainfall pattern; it is normal to find coefficients of uniformity (CHRISTIANSEN, 1942) of around 85% for the rainfall intensity within an area bounded by the corner nozzles. For rainfall simulators with only one nozzle, a lower uniformity coefficients should be expected. Multiple nozzle overlap should be considered in order to increase water application and kinetic–energy uniformity in the studied plot areas.

Further research should focus on the coupling of NOZZLE to two–dimensional overland and sediment transport models. Field experiments are also required to fully evaluate the suitability and effectiveness of the formulation presented in this report.

ACKNOWLEDGEMENT

The authors wish to thank Dr. A.F.J. Jacobs (Department of Meteorology of the WAU) for his helpful suggestions on the drop movement model and Ir. R. Uijlenhoet (Department of Water Resources of the WAU) for his comments during preparation of the manuscript. The first author is indebted to the *International Agricultural Centre* (Wageningen, The Netherlands) and the *Gulbenkian Foundation* (Lisbon, Portugal) for financial support.

REFERENCES

AINA, P.O., LAL, R. and TAYLOR, G.S., 1977. Soil and crop management in relation to soil erosion in the rainforest of western Nigeria. In: Soil erosion: prediction and control (G.R. FOSTER, ed.), SCSA, Ankeny, Iowa, 75–82.

ASTM, 1981. Standard practice for determining data criteria and processing for liquid drop size analysis. American Society for Testing and Materials, Designation E799–81, USA.

AZIZOV, M.T., 1977. Influence of soil moisture on the resistance of soil to wind erosion. Pochvovedeniye, 1, 102–105.

BERNUTH, R.D. von, 1988. Effect of trajectory angle on performance of sprinklers in wind. J. of Irrigation and Drainage Engineering, ASCE, 114(4), 579–587.

CALDWELL, D.R. and ELLIOTT, W.P., 1972. The effect of rainfall on the wind in the surface layer. Boundary–Layer Meteorology, 3, 146–151.

CHORLEY, R. J. 1978. The hillslope hydrologic cycle. In Hillslope hydrology, M. J. Kirkby (ed.), 1–42. Chichester: John Wiley.

CHRISTIANSEN, J.E., 1942. Irrigation by sprinkling. Research Bulletin 670, California Agricultural Experiment Station, California.

DISRUD, L.A. and KRAUSS, R.K., 1971. Examining the process of soil detachment from clods exposed to wind–driven simulated rainfall. Trans. ASAE, 14, 90–92.

DISRUD, L.A., LYLES, L. and SKIDMORE, E.L., 1969. How wind affects the size and shape of raindrops. Agric. Eng. 50, 617–626.

DUNNE, T. and L. B. LEOPOLD, 1978. Water in environment planning. San Francisco: Freeman.

EDLING, R.J., 1985. Kinetic energy, evaporation and wind drift of droplets from low pressure irrigation nozzles. Transactions of the American Society of Agricultural Engineers, 28(5), 1543–1550.

EIGEL, J.D. and I.D. MOORE, 1983. A simplified technique for measuring raindrop size and distribution. Transactions of the ASAE, 26(4), 1079–1084.

FERRAZZA, J., W. BARTELL and R. SCHICK, 1992. Spray nozzle drop size: How to evaluate measurement techniques and interpret data and reporting procedures. Spraying Systems Co, Bulletin No. 336, Wheaton, USA.

GERITS, J. J. P. and J. L. M. P. de LIMA, 1991. Solute transport and wind action in relation to overland flow and water erosion. Cremlingen, Catena Verlag.

HOERNER, S.F., 1958. Dynamics. University Tutorial Press, London, England.

HOLY, M., 1980. Erosion and environment. Environment Sciences and Applications, Vol. 9, Pergamon Press, Oxford, 225 p.

JUNGERIUS, P.D. and DEKKER, L.W., 1990. Water erosion in the dunes. Catena Supplement 18, 185–193.

JUNGERIUS, P.D., VERHEGGEN, A.J.T. and WIGGERS, A.J., 1981. The development of blowouts in 'De Blink', a coastal dune area near Noordwijkerhout, The Netherlands. Earth Surf. Proc. and Landforms 6, 375–396.

KIRKBY, M. J. and R. J. CHORLEY, 1967. Throughflow, overland flow and erosion. Bull. Intern. Assoc. Sci. Hydrology 12, 5–21.

KIRKBY, M. J. 1988. Hillslope runoff processes and models. Journal of Hydrology 100: 315—39.

KNOTTNERUS, D.J.C., 1980a. Maatregelen aan het stuifgevoelig grondoppervlak ter bestrijding van erosie door wind. Inst. Voor Bodemvruchtbaarheid, Nota 92, 15 pp.

KNOTTNERUS, D.J.C., 1980b. Relative humidity of the air and critical wind velocity in relation to erosion. In: Assessment of erosion (M. de BOODT and D. GABRIELS, eds.), Wiley, Chichester, 531–539.

KOHL, R.A., D.W. DEBOER and P.D. EVENSON, 1985. Kinetic energy of low pressure spray sprinklers. Transactions of the American Society of Agricultural Engineers, 28(5), 1526–1529.

LAL, R., LAWSON, T.L., and ANASTASE, A.H., 1980. Erosivity of tropical rains. In: Assessment of erosion (M. de BOODT and D. GABRIELS, eds.), Wiley, Chichester, 143–151.

LIMA, J.L.M.P. de, 1988. Morphological factors affecting overland flow on slopes. In: Land qualities in space and time. J. Bouma and A.K. Bregt (Eds.), PUDOC, Wageningen, 321–324.

LIMA, J.L.M.P. de, 1989a. The influence of the angle of incidence of the rainfall on the overland flow process. In: New Directions for Water Modeling (Proc. IAHS Third Scientific Assembly, Baltimore, 10–19 May 1989), IAHS Publ. no. 181, 73–82.

LIMA, J.L.M.P. de, 1989b. Overland flow under rainfall: some aspects related to conditioning factors and modelling. Ph.D. Thesis, Wageningen Agricultural University, 160 pp.

LIMA, J. L. M. P. de, 1989c. Overland flow under simulated wind-driven rain. In Land and water use (Proc. of the Eleventh International Congress on Agricultural Engineering), 493–500, Rotterdam: Balkema.

LIMA, J. L. M. P. de, 1989d. Raindrop splash anisotropy: slope, wind and overland flow velocity effects. Soil Technology 2, 71—8.

LIMA, J.L.M.P. de, 1990. The effect of oblique rain on inclined surfaces: a nomograph for the rain-gauge correction factor. J. of Hydrology, 115: 407–412.

LIMA, J.L.M.P. de, 1992. Splash-saltation transport under wind-driven rain. Soil Technology, 5, 151–166.

LYLES, L., 1977. Wind erosion: processes and effects on soil productivity. Trans. ASAE, 20, 880–884.

LYLES, L., DICKERSON J.D. and SCHMEIDLER, N.F., 1974. Soil detachment from clods by rainfall: effects of wind, mulch cover, and initial soil moisture. Transactions of the ASAE, 17, 697–700.

LYLES, L., DISRUD, L.A. and WOODRUFF, N.P., 1969. Effect of soil physical properties, rainfall characteristics, and wind velocity on clod disintegration by simulated rainfall. Soil Sci. Soc. Am. Proc., 33(2), 302–306.

MOLDENHAUER, W.C. and W.D. KEMPER, 1969. Interdependence of water drop energy and clod size on infiltration and clod stability. Soil Science Society of America Proc. 33(2), 297–301.

MOORE, I. D., C. L. LARSON and D. C. SLACK, 1980. Predicting infiltration and micro-relief surface storage for cultivated soils. Water Resources Research Center, Univ. of Minnesota, Minneapolis, Minnesota, Bulletin 102.

MOSS, A.J. and GREEN, P., 1983. Movement of solids in air and water by raindrop impact: effects of drop-size and water-depth variations. Aust. J. Soil Res., 21(3), 373–382.

OKAMURA, S. and K. NAKANISHI, 1969. Theoretical study on sprinkler sprays (Part four) Geometric pattern form of single sprayer under wind condition. Transactions of the Japanese Society of Irrigation, Drainage and Reclamation Engineering, 29, 35–43.

PLOEY, J. de, 1980. Some field measurements and experimental data on wind blown sands. In: Assessment of erosion (M. de BOODT and D. GABRIELS, eds.), Wiley, Chichester, 143–151.

PRUPPACHER, H.R. and J.D. KLETT, 1978. Microphysics of clouds and precipitation. D. Reidel Publishing Company, Dordrecht, Holland, 714 p.

RÖMKENS, M. J. M., S. N. PRASAD, F. D. and WHISLER, 1990. Surface sealing and infiltration. In Process studies in hillslope hydrology, M. G. Anderson & T. P. Burt (eds.), 127—72. Chichester, England: John Wiley.

ROSENBERG, N.J., BLAD, B.L. and VERMA, S.B., 1983. Microclimate: the biological environment. John Wiley & Sons, New York.

SHARON, D., 1980. The distribution of hydrologically effective rainfall incident on sloping ground. J. of Hydrology, 46, 165–188.

SPRAYING SYSTEMS Co, 1991. An engineer's guide to spray nozzles for pollution control equipment. Bulletin No. 312, Wheaton, USA.

SPRAYING SYSTEMS Co, 1992a. Performance engineered spray nozzles for efficient air pollution control. Bulletin No. 333, Wheaton, USA.

SPRAYING SYSTEMS Co, 1992b. Gas conditioning with flowback nozzle lances. Bulletin No. 294B, Wheaton, USA.

STILLMUNKES, R.T. and L.G. JAMES, 1982. Impact energy of water droplets from irrigation sprinklers. Transactions of the American Society of Agricultural Engineers, 25(1), 130–133.

STRUZER, L.R., 1972. Problem of determining precipitation falling on mountain slopes. Sov. Hydrol. Selected Pap., 2, 129–142.

UMBACK, C.R. & LEMBKE, W.D., 1966. Effect of wind on falling water drops. Transactions of the ASAE, 9, 805–808.

VORIES, E.D. and R.D. von BERNUTH, 1986. Single nozzle sprinkler performance in wind. Transactions of the American Society of Agricultural Engineers, 29(5), 1325–1330.

WILLIAMSON, R.E. and E.D. THREADGILL, 1974. A simulation for the dynamics of evaporating spray droplets in agricultural spraying. Transactions of the American Society of Agricultural Engineers, 17(2), 254–261.

YOUNG, R.A. and WIERSMA, J.L., 1973. The role of rainfall impact in soil detachment and transport. Water Resources Res., 9(6), 1629–1636.

ZASLAVSKY, D. and SINAI, G., 1981. Surface hydrology: II – distribution of raindrops. J. Hydraulic. Div., 107, 17–36.

NOTATION

Dimensionless parameters:

c	scalar
C_D	drag coefficient of the spherical shaped drop
$\mathbf{i}$	unit vector along the horizontal axis (X–axis)
$\mathbf{i_1}$	unit vector along X_1–axis (interception of the surface with the horizontal plane XY)
$\mathbf{j}$	unit vector along the horizontal axis (Y–axis)
$\mathbf{j_1}$	unit vector along Y_1–axis (steepest line of the surface in the uphill direction)
$\mathbf{k}$	unit vector along the vertical axis (positive upwards)
$\mathbf{k_1}$	unit vector perpendicular to both X_1–axis and Y_1–axis of the surface, pointing upwards
K	Von Kármán constant
n	number of impinging raindrops
Re_d	drop Reynolds number
π	3.14159265...

Dimensioned parameters:

A	characteristic cross section of the drop perpendicular to the relative flow (m^2)
D	drop diameter (assumed spherical shaped) (m)
e_x	unit vector giving the direction, along the X–axis, of the relative speed of the drop with respect to the wind (m/s)
e_y	unit vector giving the direction, along the Y–axis, of the relative speed of the drop with respect to the wind (m/s)
e_z	unit vector giving the direction, along the Z–axis, of the relative speed of the drop with respect to the wind (m/s)
$\mathbf{F}$	sum the forces $\mathbf{F_1}$, $\mathbf{F_t}$ and $\mathbf{F_d}$ (N)
$\mathbf{F_1}$	force expressing gravity in a overland flow elementary section (N)
$\mathbf{F_{aux}}$	component of vector $\mathbf{F}$ perpendicular to the unit vector $\mathbf{i_1}$
$\mathbf{F_d}$	force due to the impinging raindrops (N)
$\mathbf{F_t}$	tangential wind shear force (N)
$\mathbf{F_{x1}}$	component of vector $\mathbf{F}$ parallel to the unit vector $\mathbf{i_1}$
$\mathbf{F_{y1}}$	component of vector $\mathbf{F}$ parallel to the unit vector $\mathbf{j_1}$
g	gravitational acceleration (m/s^2)
h	average depth of flow in the elementary section (m)
H	height of the nozzle (m)
KE	kinetic energy ($watts/m^2$)
m	mass of the drop (kg)

P	rainfall intensity (mm/min)
p	fluid pressure (Pa)
P_{ef}	effective precipitation (m/s)
T_a	air temperature (°C)
u	velocity (m/s)
$u*$	friction velocity (m/s)
u_x	velocity components of the drop along the X–axis (m/s)
u_y	velocity components of the drop along the Y–axis (m/s)
u_z	velocity components of the drop along the Z–axis (m/s)
V_R	absolute value of the relative velocity of the drop with respect to the wind (m/s)
x	distance along the X–axis (m)
y	distance along the Y–axis (m)
z	distance along the Z–axis (m)
z_0	roughness length (m)
z_1	height above the ground surface along Z_1–axis (m)
w	wind speed (m/s)
w_{10}	wind speed at 10 m height (m/s)

α	angle of force $\mathbf{F}_d$ with the vertical (deg)
α_0	angle with the vertical of a drop released by the nozzle (deg)
α_{sl}	angle of the steepest line of the receiving surface with the horizontal (deg)
β	angle of the relative drop velocity with respect to the wind with the vertical (deg)
τ	frictional shear stress (Pa)
Ω	azimuth of the steepest line of the slope (uphill direction) (deg)
ω	azimuth of the wind direction (X–axis) (deg)
γ	azimuth of the relative drop velocity with respect to the wind (deg)
ν_a	kinematic viscosity of the air (m^2/s)
ρ_a	densities of the air (kg/m^3)
ρ_w	densities of the water (kg/m^3)
θ	azimuth of force $\mathbf{F}_d$ (rad)

Other symbols:

$\rightarrow$	vector
∇	gradient operator
$\cdot$	divergence operator; dot product of two vectors
$\times$	cross product of two vectors.
[A]	matrix
☹	sad face
☺	happy face

APPENDICES

APPENDIX 1 - COMPUTER PROGRAMS

A.1.1. DROP

File: "drop.h"

```
/*
 *  header file for the file drop.c
 *
 *  -defines the global variables
 *
 *  -defines the basic structure describing a state
 *
 *  -prototypes the basic functions of drop.c
 *
 */

#ifdef DROP_MAIN
#define GLOBAL
#else
#define GLOBAL extern
#endif

/*
 *   definition of fundamental constants
 */

#define Karman 0.4
                /*  Karman constant         [-] */
#define rho_w   1000.0
        /*  rho_ of the water        [kg\m3]    */
#define g       9.81
                /*  Acceleration due to gravity [m\s2]    */

/*
 *    the fundamental structure
 */

typedef struct {
   float t;            /* Time                              [s]   */
   float x, y, z;       /* coordinates of drop                  [m]   */
   float z_plane;        /* z in plane-relative coordinates         [m]   */
   float Vx, Vy, Vz;  /* components of speed              [m/s]  */
   float Vwind;       /* speed  of wind directionccc      [m/s]  */
   float drag_factor; /* used in differential equation    [1/s]  */
}
state;

/*
 * the global variables
 */

GLOBAL float plane_slope;  /* slope of plain with horizontal  [rad] */
GLOBAL float angle_wind;   /* azimuth of wind direction       [rad] */
GLOBAL float H;            /* release height of drop          [m] */

GLOBAL float Vwind_10;     /* wind speed at 10 m              [m/s] */
GLOBAL float z0;           /* surface roughness               [m] */
GLOBAL float temp;         /* air temperature                 [C] */

GLOBAL float del_t;        /* timestep                        [s] */

GLOBAL float V0;           /* initial speed                   [m/s] */
GLOBAL float diameter;     /* diameter of drop                [m] */
GLOBAL float alpha0;       /* angle with vertical             [rad] */
GLOBAL float theta0;       /* azimuth                         [rad] */

GLOBAL float abs_to_plane[3][3];

/*
 * the functions
 */

void initiate_systemconstants (void);
void make_first_state (state *to_make);
void do_step (state *from, state *to);
```

File: "drop.c"

```
/*
 *    main file of drop
 *
 *    -implements the basic functions
 *
 */
```

```
#define DROP_MAIN

#include <math.h>
#include <stdio.h>
#include "drop.h"


 float inv_kin_visc_air;          /* inverse of kin.visc. of air */
 float w_coeff, w_const;
                                /* wind speed = w_coeff * [log(z)-w_const] */
 float diffeq_coeff, grav_term;  /* used in diff equations      */

/*
 *    local procedures
 */
 void initiate_systemconstants();
 float drag_coeff(float Reynolds);
 void complete_state(state *s);

/*
 *    implementation of procedures
 */

void initiate_systemconstants()
{
   float rho_air;
   float t = temp;

   abs_to_plane[0][0] = sin(angle_wind);
   abs_to_plane[0][1] = -cos(angle_wind);
   abs_to_plane[0][2] = 0.0;
   abs_to_plane[1][0] = cos(plane_slope) * cos(angle_wind);
   abs_to_plane[1][1] = cos(plane_slope) * sin(angle_wind);
   abs_to_plane[1][2] = sin(plane_slope);
   abs_to_plane[2][0] = -sin(plane_slope) * cos(angle_wind);
   abs_to_plane[2][1] = -sin(plane_slope) * sin(angle_wind);
   abs_to_plane[2][2] = cos(plane_slope);

   rho_air = (1293.31 - 4.96*t+0.02807*t*t-0.000188*t*t*t)/1000.0;

   inv_kin_visc_air =
        1.0/(1.3045e-5 + 1.222e-7*t - 9.6471e-10*t*t+7.2873e-12*t*t*t);

   w_coeff    = Vwind_10/log(10.0/(z0));
   w_const    = log(z0);
```

```
    diffeq_coeff = del_t * (-0.75) * rho_air / (rho_w * diameter);
    grav_term = del_t * g * (1.0 - rho_air/rho_w);

}

float drag_coeff(float Reynolds)
{
    float drag_coeff = 24.0/Reynolds;

    if (Reynolds >0.5)
    {
        if (Reynolds<200)
        {
            drag_coeff = 0.49 + 26.38 / pow(Reynolds,0.845);
        }
        else if (Reynolds<10000)
        {
            drag_coeff = sqrt(0.525+drag_coeff*drag_coeff);
        }
        else
        {
            drag_coeff = 0.44;
        }
    }

    return drag_coeff;
}

void complete_state(state  *s)
{
    float Vx_rel, V_rel;
    float Reynolds;

    s->z_plane = abs_to_plane[2][0]*s->x
              + abs_to_plane[2][1]*s->y
              + abs_to_plane[2][2]*s->z;

    if (s->z_plane > 0)
    {
        s->Vwind = w_coeff * (log(s->z_plane) - w_const);
    }
    else
    {
        s->Vwind = 0;
    }
    if (s->Vwind<0) s->Vwind = 0.0;
    Vx_rel = s->Vx - s->Vwind;

    V_rel = sqrt(Vx_rel*Vx_rel+s->Vy*s->Vy+s->Vz*s->Vz);

    Reynolds  = inv_kin_visc_air * diameter  * V_rel;
    s->drag_factor = diffeq_coeff * drag_coeff(Reynolds) * V_rel;
}

void make_first_state(state *fs)
{
    fs->t  = 0.0;
    fs->x  = 0.0;
    fs->y  = 0.0;
    fs->z  = H;
    fs->Vx = V0*sin(alpha0)*cos(theta0);
    fs->Vy = V0*sin(alpha0)*sin(theta0);
    fs->Vz = -V0*cos(alpha0);

    complete_state(fs);
}


void do_step(state *from,state *to)
{

    to->Vx = from->Vx + from->drag_factor * (from->Vx - from->Vwind);
    to->Vy = from->Vy + from->drag_factor * from->Vy;
    to->Vz = from->Vz + from->drag_factor * from->Vz
                  - grav_term;


    to->x = from->x + del_t * to->Vx;
    to->y = from->y + del_t * to->Vy;
    to->z = from->z + del_t * to->Vz;

    to->t = from->t + del_t;

    complete_state(to);
}

#undef DROP_MAIN
```

File: "do_traj.c"

```
/*
 *  driver program to calculate trajectories
 *
 *  -reads input
 *
 *  -by using the functions implemented in drop.c,
 *   calculates a number of trajectories
 *
 *  -writes the results to a file
 *
 */

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include "drop.h"


#define TO_ES(f)  while((!feof(f))&&(fgetc(f)!=’=’));
#define deg_to_rad 0.0174553293

/*
 *  local procedures
 */
void get_input(FILE *from);
int get_initials(FILE *from);

/*
 *  implementation of procedures
 */

void get_input(FILE *from)
{

        TO_ES(from); fscanf(from,"%g",&plane_slope);
        plane_slope *= deg_to_rad;
        TO_ES(from); fscanf(from,"%g",&angle_wind);
        angle_wind *= deg_to_rad;
        TO_ES(from); fscanf(from,"%g",&H);

        TO_ES(from); fscanf(from,"%g",&Vwind_10);
        TO_ES(from); fscanf(from,"%g",&z0);
        TO_ES(from); fscanf(from,"%g",&temp);
        TO_ES(from); fscanf(from,"%g",&del_t);

}

int get_initials(FILE *from)
{
        TO_ES(from);
        if (!feof(from))
        {
                if (4==fscanf(from,"%f %f %f %f",&V0,&diameter,&alpha0,&theta0))
                {
                        alpha0 *= deg_to_rad;
                        theta0 *= deg_to_rad;
                        return 1;
                }
        }
        return 0;
}


int main(int argc, char* argv[])
{
        FILE *input, *output;
        state prevs, s;
        float V_tot;

        if (argc<3)
        {
                fprintf(stderr,"usage : do_traj infile statfile\n");
                exit(-1);
        }
        input = fopen(argv[1],"r");
        if (!input)
        {
                fprintf(stderr," file %s not found !!\n",argv[1]);
                exit(-1);
        }

        output = fopen(argv[2],"w");

        get_input(input);

        fprintf(output,"\#[time      x      y      z      V_total]\n");

        while (get_initials(input))
        {
```

```
        initiate_systemconstants();
        make_first_state(&s);
        fprintf(output,"\n");

        while (s.z_plane>0)
        {
            V_tot = sqrt(s.Vx*s.Vx+s.Vy*s.Vy+s.Vz*s.Vz);
            fprintf(output,"%f %f %f %f %f\n",s.t,s.x,s.y,s.z,V_tot);
            prevs = s;
            do_step(&prevs,&s);
        }

    }

    fclose(output);
    return 1;
}
```

A.1.2. NOZZLE

File: "nozzle.h"

```
/*
 *     header file for the file nozzle.c
 *
 *     -defines the global variables
 *
 *     -prototypes teh basic functions of nozzle.c
 *
 */


#ifdef NOZZLE_MAIN
#define GLOBAL
#else
#define GLOBAL extern
#endif
/*
 *   input global variables
 */

        GLOBAL float Qnozzle;      /* discharge of nozzle            [m3/s] */
        GLOBAL float max_alpha; /* maximum angle with vertical      [rad] */
        GLOBAL float layer_thickness; /* thickness of water layer      [m] */
        GLOBAL int   num_diameters;/* number of quantiles of drop diameters */
        GLOBAL float *dia_quant;   /* list of drop diameter-quantiles  [m]  */

        GLOBAL int   num_drops;    /* number of drops to be calulated       */

        GLOBAL float min_x, max_x; /* for the statistics */
        GLOBAL int   num_x_divisions;
        GLOBAL float min_y, max_y; /* for the statistics */
        GLOBAL int   num_y_divisions;

/*
 *   procedures
 */

void random_initials (void);
void initiate_statistics (void);
void calc_planehit(state *s_beforehit, state *s_afterhit,
                   float *x_plane, float *y_plane,
                  float *Vx_plane, float *Vy_plane,
                  float *V_total);
void to_tally(float x_plane, float y_plane,
             float Vx_plane, float Vy_plane,
             float V_total);
void report_tally (FILE *scalar, FILE *vector);
```

File: "nozzle.c"

```
/*
 * main file of nozzle
 *
 *     -defines the arrays for the statistics
 *
 *     -implements the basic function to similate the spray
 *
 *     -implements the basic functions for the statistics
 *
 */

#define NOZZLE_MAIN

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include "drop.h"
#include "nozzle.h"
#include "uniform.h"

#define PI 3.14159265359
#define volume_factor 1.0471976


/*
 *   local global variables
 */


 static float total_volume;
 float step_x, step_y;
 int   **stat_num_drops;
 float **stat_diameter;
 float **stat_volume;
 float **stat_kin_energy;
 float **stat_impuls_x;
 float **stat_impuls_y;

/*
 *   implementation of procedures
 */

void random_initials()
{
   int quant_num;
   float rnd,x;

   theta0 = 2*PI*uniform();
   alpha0 = max_alpha * uniform();
   rnd = uniform()*(num_diameters-1);
   quant_num = floor(rnd);
   x = rnd - quant_num;
   diameter = x * dia_quant[quant_num] +
       (1-x) * dia_quant[quant_num+1];
}



void initiate_statistics()
{
   int i,j;

   total_volume = 0.0;

   step_x = (max_x - min_x)/num_x_divisions;
   step_y = (max_y - min_y)/num_y_divisions;

   stat_num_drops =
              (int**) malloc(sizeof(int*)*(unsigned) num_x_divisions);
   stat_diameter =
              (float**) malloc(sizeof(float*)*(unsigned) num_x_divisions);
   stat_volume    =
              (float**) malloc(sizeof(float*)*(unsigned) num_x_divisions);
   stat_kin_energy =
              (float**) malloc(sizeof(float*)*(unsigned) num_x_divisions);
   stat_impuls_x        =
              (float**) malloc(sizeof(float*)*(unsigned) num_x_divisions);
   stat_impuls_y        =
              (float**) malloc(sizeof(float*)*(unsigned) num_x_divisions);
```

```
    for (i=0; i<num_x_divisions; i++)
    {
        stat_num_drops[i] =
                        (int*) malloc(sizeof(int) * (unsigned) num_y_divisions);
        stat_diameter[i] =
                        (float*) malloc(sizeof(float) * (unsigned) num_y_divisions);
        stat_volume[i] =
                        (float*) malloc(sizeof(float) * (unsigned) num_y_divisions);
        stat_kin_energy[i] =
                        (float*) malloc(sizeof(float) * (unsigned) num_y_divisions);
        stat_impuls_x[i] =
                        (float*) malloc(sizeof(float) * (unsigned) num_y_divisions);
        stat_impuls_y[i] =
                        (float*) malloc(sizeof(float) * (unsigned) num_y_divisions);

        for (j=0; j<num_y_divisions; j++)
        {
            stat_num_drops[i][j]  = 0;
            stat_diameter[i][j]   = 0.0;
            stat_volume[i][j]     = 0.0;
            stat_kin_energy[i][j] = 0.0;
             stat_impuls_x[i][j]  = 0.0;
             stat_impuls_y[i][j]  = 0.0;
        }
    }
}

void calc_planehit(state *s_beforehit, state *s_afterhit,
                   float *x_plane, float *y_plane,
                   float *Vx_plane, float *Vy_plane,
                   float *V_total)
{
    float x_hit, y_hit, Vx_hit, Vy_hit, Vz_hit;
    float lambda;

    lambda = - s_afterhit->z_plane
        /(s_beforehit->z_plane-s_afterhit->z_plane);

    x_hit    = lambda *s_beforehit->x+ (1-lambda)*s_afterhit->x;
    y_hit    = lambda *s_beforehit->y+ (1-lambda)*s_afterhit->y;

    Vx_hit   = lambda * s_beforehit->Vx + (1-lambda) * s_afterhit->Vx;
    Vy_hit   = lambda * s_beforehit->Vy + (1-lambda) * s_afterhit->Vy;
    Vz_hit   = lambda * s_beforehit->Vz + (1-lambda) * s_afterhit->Vz;

    *V_total = sqrt(Vx_hit * Vx_hit + Vy_hit * Vy_hit + Vz_hit * Vz_hit);
```

```
   *x_plane = abs_to_plane[0][0] * x_hit
           + abs_to_plane[0][1] * y_hit;
   *y_plane = abs_to_plane[1][0] * x_hit
           + abs_to_plane[1][1] * y_hit;

   *Vx_plane = abs_to_plane[0][0] * Vx_hit
            + abs_to_plane[0][1] * Vy_hit
           + abs_to_plane[0][2] * Vz_hit;
   *Vy_plane = abs_to_plane[1][0] * Vx_hit
           + abs_to_plane[1][1] * Vy_hit
            + abs_to_plane[1][2] * Vz_hit;

}

void to_tally(float x_plane, float y_plane,
            float Vx_plane, float Vy_plane,
            float V_total)
{
   int i,j;
   float volume, mass;

   volume = volume_factor * diameter * diameter * diameter;
     mass   = rho_w * volume;

   total_volume += volume;

   if ((x_plane>min_x)&&(x_plane<max_x)
     &&(y_plane>min_y)&&(y_plane<max_y))
   {
     i = floor((x_plane-min_x)/step_x);
    j = floor((y_plane-min_y)/step_y);


    stat_num_drops[i][j]++;
     stat_diameter[i][j] +=diameter;
    stat_volume[i][j]   += volume;
         stat_kin_energy[i][j] += 0.5 * mass * V_total * V_total;
    stat_impuls_x[i][j]   += mass * Vx_plane;
    stat_impuls_y[i][j]   += mass * Vy_plane;
   }
}

void report_tally(scalarfile,vectorfile)
FILE *scalarfile, *vectorfile;
{
     int i,j;
    float surface;
    float gx, gy, Fx_rain, Fy_rain;
    float root_tau_over_rho, wind_shear_force;
    float mid_x, mid_y;
    float mean_diam;
    float tot_time;

    tot_time = total_volume / Qnozzle;

    printf("total time of simulation [s] = %g\n",tot_time);

    printf("number of surfaces = %d\n",
    num_x_divisions * num_y_divisions);

    surface = step_x * step_y;

    printf("surface of each unit [m^2] = %f\n",surface);

    if (scalarfile)
    {
        fprintf(scalarfile,"#[%10s %11s %11s %11s %11s %11s]=\n",

        "midx","midy","diam","drop intens","vol intens","kin energ");

        mid_x = min_y + step_x/2.0;
        for (i=0; i<num_x_divisions; i++)
        {
            mid_y = min_y + step_y/2.0;
            for (j=0; j<num_y_divisions; j++)
            {
                if (stat_num_drops[i][j]>0)
                {
                    mean_diam = stat_diameter[i][j]/stat_num_drops[i][j];
                }
                else
                {
                    mean_diam = 0;
                }
                fprintf(scalarfile,"% 10.4e % 10.4e % 10.4e % 10.4e % 10.4e
% 10.4e\n",
                mid_x,mid_y,
                mean_diam,
                stat_num_drops[i][j]/tot_time,
                stat_volume[i][j]/tot_time,
                stat_kin_energy[i][j]);
```

```
                        mid_y += step_y;
                    }
                    fprintf(scalarfile,"\n");
                    mid_x += step_x;
                }
            }

        if (vectorfile)
        {
                gx = surface * layer_thickness * g * rho_w * abs_to_plane[0][2];
                gy = surface * layer_thickness * g * rho_w * abs_to_plane[1][2];
                fprintf(vectorfile,"  gravity force in x-direction = %f\n",gx);
                fprintf(vectorfile,"  gravity force in y-direction = %f\n",gy);

                root_tau_over_rho  = Vwind_10 * Karman / log(10/z0);
                wind_shear_force   = surface * root_tau_over_rho * root_tau_over_rho;
                fprintf(vectorfile,"  wind stress force in x-direction = %f\n",
                wind_shear_force);

                fprintf(vectorfile,"\n[%10s %11s %10s %11s %11s %11s %11s]=\n",
                "midx", "midy","num drops", "Fx_rain", "Fy_rain", "Fx_total", "Fy_total");
                mid_x = min_y + step_x/2.0;
                for (i=0; i<num_x_divisions; i++)
                {
                    mid_y = min_y + step_y/2.0;
                    for (j=0; j<num_y_divisions; j++)
                    {
                        Fx_rain = stat_impuls_x[i][j]/tot_time;
                        Fy_rain = stat_impuls_y[i][j]/tot_time;

                        fprintf(vectorfile,"% 10.4e % 10.4e %10d % 10.4e % 10.4e %
10.4e % 10.4e\n",
                        mid_x,mid_y,    stat_num_drops[i][j],    Fx_rain,    Fy_rain,
Fx_rain+gx+wind_shear_force, Fy_rain+gy);

                        mid_y += step_y;
                    }
                    mid_x += step_x;
                }
        }
}

#undef NOZZLE_MAIN
```

File: "uniform.h"

```
/*
 * prototypes the function uniform
 */

float uniform(void);
```

File: "uniform.c"

```
/*
 * implements a uniform random number generator by
 * using the random number generator ran3 from the book
 * Numerical Recipes
 * by Press e.a.
 */


#define MBIG 1000000000
#define MSEED 161803398
#define MZ 0
#define FAC (1.0/MBIG)

float ran3(idum)
int *idum;
{
        static int inext,inextp;
        static long ma[56];
        static int iff=0;
        long mj,mk;
        int i,ii,k;

        if (*idum < 0 || iff == 0) {
                iff=1;
                mj=MSEED-(*idum < 0 ? -*idum : *idum);
                mj %= MBIG;
                ma[55]=mj;
                mk=1;
                for (i=1;i<=54;i++) {
                        ii=(21*i) % 55;
                        ma[ii]=mk;
                        mk=mj-mk;
                        if (mk < MZ) mk += MBIG;
                        mj=ma[ii];
                }
                for (k=1;k<=4;k++)
                        for (i=1;i<=55;i++) {
                                ma[i] -= ma[1+(i+30) % 55];
                                if (ma[i] < MZ) ma[i] += MBIG;
                        }
                inext=0;
                inextp=31;
                *idum=1;
        }
        if (++inext == 56) inext=1;
        if (++inextp == 56) inextp=1;
        mj=ma[inext]-ma[inextp];
        if (mj < MZ) mj += MBIG;
        ma[inext]=mj;
        return mj*FAC;
}

#undef MBIG
#undef MSEED
#undef MZ
#undef FAC

/*
 * change the following value to alter the seed
 */

static int newseed = -3;

float uniform(void)
{
        return ran3(&newseed);
}
```

File: "options.h"

```
/*
 * header file for the use of options.c
 *
 *      -defines the global file names
 *
 *      -prototypes the basic function
 *
 */

#ifndef OPTIONS_MAIN
#define GLOBAL
#else
#define GLOBAL extern
#endif

GLOBAL char* progname ;
GLOBAL char* inputfile_name;
GLOBAL char* scalar_stat_file_name  ;
GLOBAL char* vector_stat_file_name  ;
GLOBAL char* hit_file_name  ;

void get_options(int argc, char* argv[]);
```

File: "options.c"

```
/*
 * main file of options
 *
 *      -implements the function get_options
 *
 */

#define OPTIONS_MAIN

#include <stdio.h>
#include <stdlib.h>
#include "options.h"

static char* correct_use = "%s inputfile [-s scalar_stat_file] [-v vector_stat_file] [-h
hit_file]";
static char* default_inputfile_name = "";
static char* default_scalar_stat_file_name  = "";
static char* default_vector_stat_file_name  = "";
static char* default_hit_file_name  = "";


void error(char *message)
{
        fprintf(stderr,"%s : %s\n",progname, message);
        fprintf(stderr,"correct usage : ");
        fprintf(stderr,correct_use,progname);
        fprintf(stderr,"\n");
        exit(-1);
}


void set_defaults(void)
{
        inputfile_name = default_inputfile_name;
        scalar_stat_file_name = default_scalar_stat_file_name;
        vector_stat_file_name = default_vector_stat_file_name;
        hit_file_name = default_hit_file_name;
}

void get_options(int argc, char* argv[])
{
        set_defaults();

        progname = argv[0];
        argc--; argv++;

        if (argc==0)
        {
                error("input file name expected");
        }
        inputfile_name = argv[0];
        argc--; argv++;

        while (argc>0 && argv[0][0]=='-')
        {
                switch(argv[0][1])
                {
                        case 's' :
                                argc--; argv++;
                                if (argc == 0)
                                {
                                        error("after -s file name expected");
                                }
                                else
                                {
                                        scalar_stat_file_name = argv[0];
                                        argc--; argv++;
                                }
                                break;
```

```
            case 'v' :
                argc--; argv++;
                if (argc == 0)
                {
                    error("after -v file name expected");
                }
                else
                {
                    vector_stat_file_name = argv[0];
                    argc--; argv++;
                }
                break;
            case 'h' :
                argc--; argv++;
                if (argc == 0)
                {
                    error("after -h file name expected");
                }
                else
                {
                    hit_file_name = argv[0];
                    argc--; argv++;
                }
                break;
            default :
                error("unknown option");
        }
    }
}

#undef OPTIONS_MAIN
```

File: "do_spray.c"

```
/*
 * driver program to simulate the nozzle
 *
 * -reads input
 *
 * -by using the function random_initial generates a drop
 *
 * -by using the functions in drop.c calculates the trajectory of the drop
 *
 * -calculates the hit on the plane
 *
 * -reports the hits to the statistical procedures in nozzle.c
 *
 * -writes the desired results to files
 *
 */

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include "drop.h"
#include "nozzle.h"
#include "options.h"

#define TO_ES(f) while((!feof(f))&&(fgetc(f)!=’=’));
#define deg_to_rad 0.0174553293
/*
 *  local procedures
 */
void get_input(FILE *from);

/*
 *  implementation
 */

void get_input(FILE *from)
{
        int i;

        TO_ES(from); fscanf(from,"%g",&plane_slope);
        plane_slope *= deg_to_rad;
        TO_ES(from); fscanf(from,"%g",&angle_wind);
        angle_wind *= deg_to_rad;
        TO_ES(from); fscanf(from,"%g",&H);
        TO_ES(from); fscanf(from,"%g",&layer_thickness);
        TO_ES(from); fscanf(from,"%g",&Vwind_10);
        TO_ES(from); fscanf(from,"%g",&z0);
        TO_ES(from); fscanf(from,"%g",&temp);

        TO_ES(from); fscanf(from,"%g",&Qnozzle);
        TO_ES(from); fscanf(from,"%g",&V0);
        TO_ES(from); fscanf(from,"%g",&max_alpha);
        max_alpha *= deg_to_rad;
        TO_ES(from); fscanf(from,"%d",&num_diameters);
        dia_quant =  (float*) malloc(sizeof(float)*(unsigned)num_diameters);
        for (i=0;i<num_diameters;i++)
        {
                TO_ES(from);fscanf(from,"%g",&dia_quant[i]);
        }
        TO_ES(from);fscanf(from,"%d",&num_drops);
        TO_ES(from);fscanf(from,"%g",&del_t);

        TO_ES(from);fscanf(from,"%g",&min_x);
        TO_ES(from);fscanf(from,"%g",&max_x);
        TO_ES(from);fscanf(from,"%d",&num_x_divisions);
        TO_ES(from);fscanf(from,"%g",&min_y);
        TO_ES(from);fscanf(from,"%g",&max_y);
        TO_ES(from);fscanf(from,"%d",&num_y_divisions);


}



int main(int argc, char* argv[])
{
        FILE *input,*scalar,*vector,*hit;
        state prevs, s;
        float x_plane, y_plane, Vx_plane, Vy_plane, V_total;

        get_options(argc,argv);

        input = fopen(inputfile_name,"r");
        if (!input)
        {
                fprintf(stderr,"%s : file %s not found!\n",progname,inputfile_name);
                exit(1);
        }

        if (hit_file_name)
        {
                hit= fopen(hit_file_name,"w");
        }
        else
        {
                hit = NULL;
        }

        get_input(input);

        initiate_statistics();
```

```
while (num_drops-->0)
{
        if (num_drops%100==0)
        {
                fprintf(stderr," still to do : %d drops\n",num_drops);
        }
        random_initials();
        initiate_systemconstants();
        make_first_state(&s);
        while (s.z_plane>0)
        {
                prevs = s;
                do_step(&prevs,&s);
        }
        calc_planehit(&prevs,&s,&x_plane,
                        &y_plane,&Vx_plane,&Vy_plane,&V_total);
        if (hit)
        {
                fprintf(hit,"%f %f\n",x_plane, y_plane);
        }
        to_tally(x_plane, y_plane, Vx_plane, Vy_plane, V_total);
}

if (hit) fclose(hit);


if (scalar_stat_file_name)
{
        scalar = fopen(scalar_stat_file_name,"w");
}
else
{
        scalar = NULL;
}

if (vector_stat_file_name)
{
        vector = fopen(vector_stat_file_name,"w");
}
else
{
        vector = NULL;
}

report_tally(scalar,vector);
```

```
        if (scalar) fclose(scalar);
        if (vector) fclose(vector);

        return 1;

}
```

A.1.3. EXAMPLE OF INPUT FILES

File: "indrop"

```
geometrical information :

    slope of plane [deg] = 10.0
    azimuth of wind direction [deg] = -10.0
    height of nozzle [m]  = 2

physical information :

    Vwind at 10 m [m/s] = 7
    z0  [m] = 0.003
    temperature [C] = 21

for the calculations :

    time step = 0.01

initial values :
          V0 [m/s]  diameter [m]  alpha0 [deg]  theta0 [deg]
1st =     5           1.0e-3        0             0
2nd =      5          1.0e-3        30            0
3rd =     5           2.0e-3        60            90
4th =     5           4.0e-3        90            180
```

File: "innozzle"

```
geometrical information :

    slope of plane [deg] = -10.0
    azimuth of wind direction [deg] = 0.0
    height of nozzle [m]  = 2
    thickness of water layer [m] = 0.005

physical information :

    Vwind at 10 m [m/s] = 3.5
    z0  [m] = 0.003
    temperature [C] = 21

description of nozzle :

    discharge    [m3/s] = 0.002
    initial speed [m/s] = 5
    maximum alpha [deg] = 50
    number of diameters = 9
        diameter 1 [m] = 1.165e-3
        diameter 2 [m] = 1.575e-3
        diameter 3 [m] = 1.900e-3
        diameter 4 [m] = 2.195e-3
        diameter 5 [m] = 2.485e-3
        diameter 6 [m] = 2.780e-3
        diameter 7 [m] = 3.100e-3
        diameter 8 [m] = 3.490e-3
        diameter 9 [m] = 4.050e-3

information for computations :

    number of drops = 50000
    time step = 0.01

information for output :
    min_x = -1.0
    max_x = 1.0
    num_x_divisions = 15
    min_y = -1.0
    max_y = 1.0
    num_y_divisions = 15
```

A.1.4. EXAMPLE OF OUTPUT FILES

File: "outdrop"

```
#[time      x        y        z      V_total]

0.000000 0.000000 0.000000 2.000000 5.000000
0.010000 0.048362 0.000000 1.999029 4.837145
0.020000 0.095190 0.000000 1.997109 4.686752
0.030000 0.140579 0.000000 1.994268 4.547775
0.040000 0.184614 0.000000 1.990532 4.419300
0.050000 0.227371 0.000000 1.985925 4.300526
0.060000 0.268922 0.000000 1.980468 4.190742
0.070000 0.309329 0.000000 1.974181 4.089321
0.080000 0.348651 0.000000 1.967084 3.995698
0.090000 0.386940 0.000000 1.959193 3.909368
0.100000 0.424244 0.000000 1.950524 3.829874
0.110000 0.460610 0.000000 1.941095 3.756800
0.120000 0.496076 0.000000 1.930918 3.689764
0.130000 0.530682 0.000000 1.920009 3.628417
0.140000 0.564461 0.000000 1.908381 3.572433
0.150000 0.597445 0.000000 1.896046 3.521507
0.160000 0.629664 0.000000 1.883018 3.475357
0.170000 0.661145 0.000000 1.869308 3.433714
0.180000 0.691914 0.000000 1.854928 3.396323
0.190000 0.721994 0.000000 1.839891 3.362943
0.200000 0.751408 0.000000 1.824207 3.333344
etc...
```

File: "scalar"

```
#[  midx        midy         diam      drop intens   vol intens    kin energ]=

-9.3333e-01 -9.3333e-01  0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00
-9.3333e-01 -8.0000e-01  0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00
-9.3333e-01 -6.6667e-01  2.3970e-03  1.8411e+02  3.3800e-06  5.9876e-04
-9.3333e-01 -5.3333e-01  3.7475e-03  9.2053e+01  5.0731e-06  1.0521e-03
-9.3333e-01 -4.0000e-01  2.8514e-03  1.8411e+02  4.5866e-06  8.4286e-04
-9.3333e-01 -2.6667e-01  1.7880e-03  3.6821e+02  2.4226e-06  3.4131e-04
-9.3333e-01 -1.3333e-01  3.0186e-03  1.8411e+02  5.5564e-06  1.0763e-03
```

```
-9.3333e-01 -2.9802e-08  2.6368e-03  1.8411e+02  4.1883e-06  7.9582e-04
-9.3333e-01  1.3333e-01  2.7443e-03  9.2053e+01  1.9923e-06  3.6287e-04
-9.3333e-01  2.6667e-01  2.4056e-03  4.6026e+02  7.1115e-06  1.2186e-03
-9.3333e-01  4.0000e-01  3.7515e-03  9.2053e+01  5.0896e-06  1.0904e-03
-9.3333e-01  5.3333e-01  1.7342e-03  1.8411e+02  1.0067e-06  1.3107e-04
-9.3333e-01  6.6667e-01  2.5197e-03  1.8411e+02  3.3810e-06  6.0134e-04
-9.3333e-01  8.0000e-01  0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00
-9.3333e-01  9.3333e-01  0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00

-8.0000e-01 -9.3333e-01  3.1139e-03  9.2053e+01  2.9107e-06  5.3720e-04
-8.0000e-01 -8.0000e-01  3.0811e-03  9.2053e+01  2.8195e-06  5.1447e-04
-8.0000e-01 -6.6667e-01  2.3011e-03  9.2053e+01  1.1746e-06  1.8485e-04
-8.0000e-01 -5.3333e-01  2.4692e-03  9.2053e+01  1.4513e-06  2.4254e-04
-8.0000e-01 -4.0000e-01  1.6393e-03  9.2053e+01  4.2466e-07  5.3071e-05
-8.0000e-01 -2.6667e-01  2.3644e-03  1.8411e+02  2.7649e-06  4.7737e-04
-8.0000e-01 -1.3333e-01  0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00
-8.0000e-01 -2.9802e-08  2.8097e-03  5.5232e+02  1.4021e-05  2.7004e-03
etc...
```

File: "vector"

```
  gravity force in x-direction = 0.000000
  gravity force in y-direction = -1.514387
  wind stress force in x-direction = 0.000000


[     midx        midy   num drops    Fx_rain     Fy_rain      Fx_total     Fy_total]=

-9.3333e-01 -9.3333e-01          0  0.0000e+00  0.0000e+00  0.0000e+00 -1.5144e+00
-9.3333e-01 -8.0000e-01          0  0.0000e+00  0.0000e+00  0.0000e+00 -1.5144e+00
-9.3333e-01 -6.6667e-01          2 -5.9465e-03 -1.0855e-03 -5.9465e-03 -1.5155e+00
-9.3333e-01 -5.3333e-01          1 -9.8013e-03 -2.4392e-04 -9.8013e-03 -1.5146e+00
-9.3333e-01 -4.0000e-01          2 -8.0753e-03  7.7868e-04 -8.0753e-03 -1.5136e+00
-9.3333e-01 -2.6667e-01          4 -3.4439e-03  9.9004e-04 -3.4439e-03 -1.5134e+00
-9.3333e-01 -1.3333e-01          2 -1.0356e-02  4.4270e-03 -1.0356e-02 -1.5100e+00
-9.3333e-01 -2.9802e-08          2 -7.4522e-03  3.9222e-03 -7.4522e-03 -1.5105e+00
-9.3333e-01  1.3333e-01          1 -3.2097e-03  2.5992e-03 -3.2097e-03 -1.5118e+00
-9.3333e-01  2.6667e-01          5 -1.1625e-02  9.6291e-03 -1.1625e-02 -1.5048e+00
-9.3333e-01  4.0000e-01          1 -9.1159e-03  9.5857e-03 -9.1159e-03 -1.5048e+00
-9.3333e-01  5.3333e-01          2 -1.1527e-03  1.5202e-03 -1.1527e-03 -1.5129e+00
-9.3333e-01  6.6667e-01          2 -5.2132e-03  6.6865e-03 -5.2132e-03 -1.5077e+00
-9.3333e-01  8.0000e-01          0  0.0000e+00  0.0000e+00  0.0000e+00 -1.5144e+00
-9.3333e-01  9.3333e-01          0  0.0000e+00  0.0000e+00  0.0000e+00 -1.5144e+00
```

```
-8.0000e-01 -9.3333e-01      1 -4.2464e-03 -2.2843e-03 -4.2464e-03 -1.5167e+00
-8.0000e-01 -8.0000e-01      1 -4.5496e-03 -2.0045e-03 -4.5496e-03 -1.5164e+00
-8.0000e-01 -6.6667e-01      1 -1.6551e-03 -2.8263e-04 -1.6551e-03 -1.5147e+00
-8.0000e-01 -5.3333e-01      1 -2.1291e-03 -3.8172e-05 -2.1291e-03 -1.5144e+00
-8.0000e-01 -4.0000e-01      1 -4.9049e-04  8.6237e-05 -4.9049e-04 -1.5143e+00
-8.0000e-01 -2.6667e-01      2 -4.0701e-03  1.2873e-03 -4.0701e-03 -1.5131e+00
-8.0000e-01 -1.3333e-01      0  0.0000e+00  0.0000e+00  0.0000e+00 -1.5144e+00
etc...
```

File: "hit"

```
0.558939 0.288504
0.922508 -0.266035
-0.066993 -0.224129
0.362926 -0.281723
0.016137 0.003500
0.997323 -0.194311
-0.200751 0.522201
-0.120800 1.280214
-1.205076 0.165641
0.943974 -0.266750
0.122412 0.277993
0.005106 -0.865314
-0.128340 -0.385061
0.537177 0.977040
-0.003817 -0.634556
-0.270335 -0.202089
0.450188 0.906229
-0.286875 0.138478
0.553735 -0.406856
1.105880 0.323322
0.082548 -0.101891
-0.201789 1.135718
0.167809 0.284890
-0.529383 0.023909
-0.295914 0.084560
0.948872 -0.049986
0.764479 -0.357817
0.098759 0.967760
-1.057035 0.114657
-0.512973 0.753780
0.206265 0.059189
etc...
```

APPENDIX 2 - LABORATORY EXPERIMENTS WITH FULL-CONE NOZZLE UNDER WINDLESS CONDITIONS

Laboratory experiments were executed with a single full-cone nozzle under windless conditions to determine the water application distribution. In Table A.2.1, x and y are distances (cm) and P is the rainfall intensity (mm/min). Nozzle is at x = 60 cm and y = 60 cm.

Table A.2.1 - Rainfall intensity (mm/min).

X	Y	P
0	132	4.5
22	132	9.9
44	132	10.0
66	132	9.7
88	132	9.4
110	132	5.7
132	132	3.8
0	110	8.0
22	110	8.5
44	110	8.5
66	110	8.2
88	110	8.2
110	110	9.3
132	110	9.4
0	88	7.1
22	88	8.8
44	88	11.0
66	88	10.9
88	88	8.2
110	88	8.0
132	88	10.5
0	66	7.3
22	66	10.4
44	66	15.9
66	66	13.9
88	66	9.7
110	66	7.1
132	66	9.3
0	44	7.5
22	44	10.6
44	44	15.4
66	44	14.4

Table A.2.1 - Rainfall intensity (mm/min) (continuation).

X	Y	P
88	44	8.8
110	44	6.5
132	44	8.4
0	22	7.5
22	22	8.1
44	22	11.2
66	22	10.4
88	22	7.3
110	22	6.9
132	22	8.8
0	0	8.1
22	0	8.3
44	0	9.2
66	0	9.4
88	0	9.0
110	0	9.3
132	0	6.9

Calculated parameters:

Average rainfall intensity (Simpson's Rule) = 9.42 mm/min
Area of positive volume = 36 %
Area of negative volume = 64 %

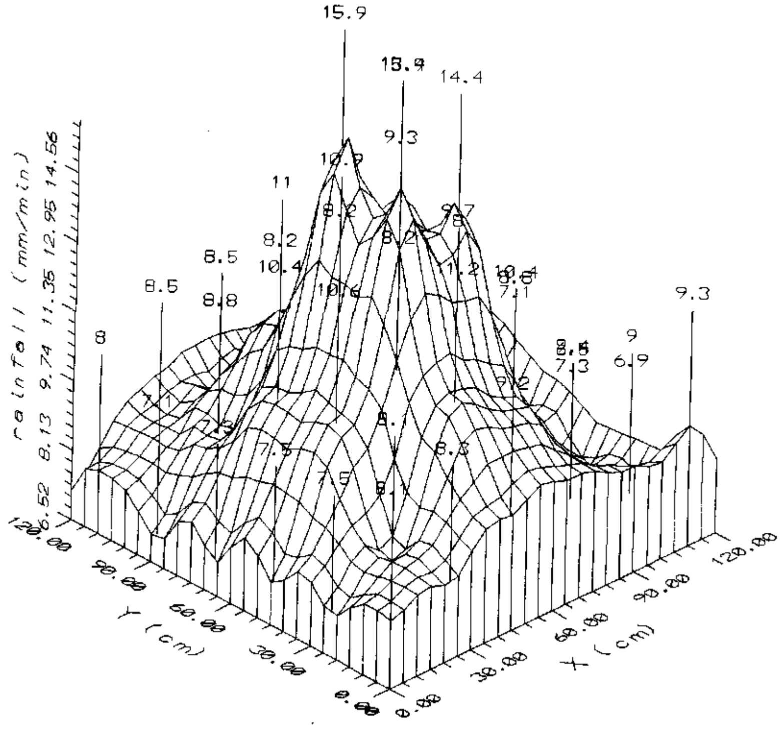


Fig. A.2.1 - Rainfall distribution on a 1.20 x 1.20 m^2 plot showing points of measured rainfall intensities (see Table A.2.1).

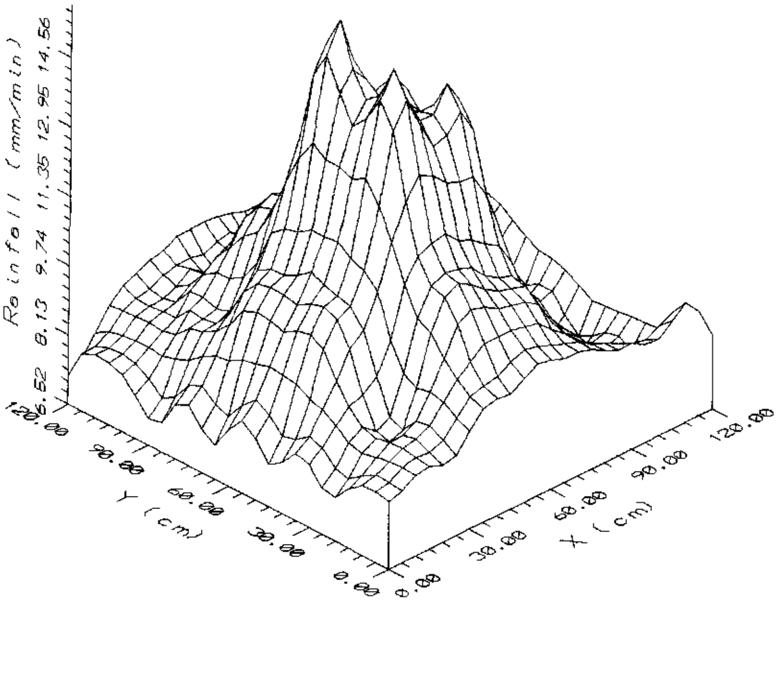


Fig. A.2.2 - Rainfall distribution on a 1.20 x 1.20 m^2 plot.

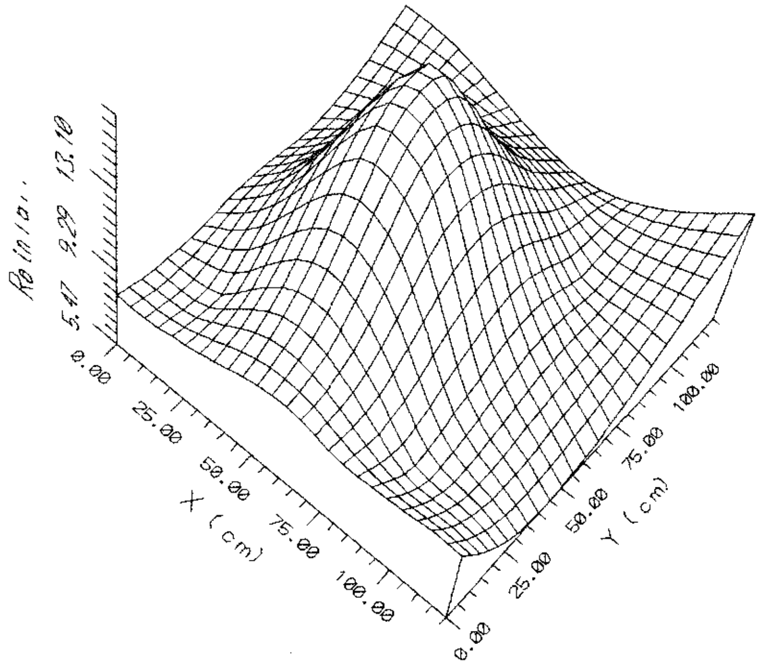


Fig. A.2.3 - Rainfall distribution on a 1.20 x 1.20 m^2 plot after smoothing with the minimum curvature method.