SANDIA REPORT

SAND87—0951 • UC—70

Unlimited Release

Printed January 1988

FREEFALL: A Seabed Penetrator Flight Code

James Hickerson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789

***When printing a copy of any digitized SAND
Report, you are required to update the
markings to current standards.***

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

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

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

NTIS price codes
Printed copy: A05
Microfiche copy: A01

SAND87-0951
Unlimited Release
Printed January 1988

Distribution
Category UC-70

FREEFALL: A SEABED PENETRATOR FLIGHT CODE

James Hickerson
Seabed Programs Division
Sandia National Laboratories
Albuquerque, New Mexico 87185

ABSTRACT

This report presents a one-dimensional model and computer program for predicting the motion of seabed penetrators. The program calculates the acceleration, velocity, and depth of a penetrator as a function of time from the moment of launch until the vehicle comes to rest in the sediment. The code is written in Pascal language for use on a small personal computer. Results are presented as printed tables and graphs. A comparison with experimental data is given which indicates that the accuracy of the code is perhaps as good as current techniques for measuring vehicle performance.

CONTENTS

1. INTRODUCTION	1
1.1 Background	1
1.2 General Description of the Code	4
2. COMPARISON OF FREEFALL PREDICTIONS WITH EXPERIMENT	6
3. THE WATER FLIGHT MODEL	15
3.1 Mathematical Basis	15
3.2 Drag Calculations	17
4. THE BURIAL MODEL	23
5. A USER'S GUIDE TO FREEFALL.PAS	28
5.1 General Guidance	28
5.2 Operational Hints	30
5.3 Running the Code	32
6. REFERENCES	37
APPENDIX A: Principal FREEFALL VIDEO SCREENS	A-1
APPENDIX B: A FREEFALL.PAS PROGRAM LISTING	B-1

FIGURES

1. The Marine Sediment Penetrator, MSP-1, 3-in. Diameter	2
2. The MK-7A Reprocessed High-Level-Waste Penetrator	3
3. Representative Penetrators Tested by the Seabed Working Group	10
4. A Comparison of Predicted vs. Measured Impact Velocities for Penetrators in Tables 1 and 2	11
5. A Comparison of Predicted vs. Measured Burial Depths for Penetrators in Tables 1 and 2	11
6. A Comparison of Predicted vs. Measured Velocities after Launch for a MK-1A Penetrator	13
7. A Comparison of Predicted vs. Measured Velocities During Soil Penetration for Sandia MK-1A and BRE Type HM Penetrators	13

FIGURES (Continued)

8.	Decelerations During Penetration for an ISP Penetrator Encountering Sand Layers Compared with FREEFALL Predictions for a Uniform Sediment of Similar Average Strength	14
9.	Decelerations During Penetration for an ISP Penetrator in the Relatively Uniform Sediments of the Nares Abyssal Plain Compared with FREEFALL Predictions	14
10.	The Reference Temperature Profile for the Water Column Over the Nares Abyssal Plain Compared with the Temperature Algorithm in FREEFALL	16
11.	Relationship between the Friction Drag Coefficient and the Reynolds Number Calculated by FREEFALL	20
12.	The Delft Piezocone	24

TABLES

1.	Penetrators That Have Been Used to Verify FREEFALL	8
2.	Measured and Predicted Velocities and References for the Penetrator Types Listed in Table 1	9
3.	Summary of Drag Contributions	18
4.	Relationship Between f and q_c As Determined by Particle Size and Soil Type	29
5.	Guidance for Selecting Roughness Factors	30

FREEFALL: A SEABED PENETRATOR FLIGHT CODE

1. INTRODUCTION

1.1 Background

Seabed penetrators are a class of vehicles developed for the purpose of dynamically emplacing a payload beneath the seafloor. That payload is typically an instrumentation package and a telemetry system (Colp et al., 1975; Calloway, 1983; James, 1983; Hauser and Hickerson, 1987). Figure 1 illustrates the Sandia Marine Sediment Penetrator that was developed to measure sediment strengths. The seabed penetrator has also been considered as a vehicle for emplacing high-level radioactive waste in deep sea sediments (Hinga et al., 1982; Seabed Programs Division, 1983; Hickerson et al., 1987). The Sandia MK-7A (Figure 2) typifies these penetrators.

Most applications for seabed penetrators are satisfied by a vehicle that is accelerated by gravity and achieves a desired burial depth by virtue of its shape and the kinetic energy generated during free fall. James (1983) discusses an alternate vehicle that is propelled from a gun into the sediment. This report concerns itself with gravity-accelerated penetrators, although the burial portions of the code have been used to predict the penetration of gun-launched vehicles.

A simple design code, named FREEFALL, has been written and used to predict the axial motion of seabed penetrators. It allows a designer to vary the dimensions and weight of simple bodies of revolution and to study the effect of those changes on the resultant drag of components, accelerations, velocities, and burial depths. Similarly, the burial behavior of a specific vehicle can be predicted at different sites by entering the soil parameters appropriate to each.

Figure 1. The Marine Sediment Penetrator, MSP-1, 3-in. Diameter (from Colp et al., 1975).

Dimensions: mm (in.)

NOTES:

1. Waste Form: 0.256 m³ of Reprocessed, Glassy, High-level Waste, 2.9 g/cc
2. Heat Load: 1.91 kW/Canister, 10 yr year out of Reactor
3. Heavy Metal Content: 1.97 MTU
4. Max. Emplacement Temperature in Canister Wall: 200°C
5. Weight: 8200 kg dry, 6850 kg wet
6. Static Margin: 8%
7. Terminal Velocity: 64 m/s
8. Burial (SNAP Site): 44 m to Tail
9. Design Concept: Double Containment

ITEM	DESCRIPTION
1	Nose, ASTM A181 Class 60
2	REF. HLW Canister, Ti-6Al-4V
3	Body, HY-80, 550 MPa Yield
4	Assy Joint, ELECTROSLAG Weld
5	Tail, ASTM A181 Class 60
6	Acoustic Transmitter
7	Fin, 12 mm ASTM A516 Gr.B

Figure 2. The MK-7A Reprocessed High-Level-Waste Penetrator (from Hickerson et al., 1987).

1.2 General Description of the Code

FREEFALL combines two codes written by the author, FLIGHT and BURIAL, into a single design-oriented program. FLIGHT calculates the velocity, depth, and drag characteristics of the penetrator in water at 96 time steps after release from the surface. BURIAL takes the results of FLIGHT and user-specified soil constants to calculate analogous data for the sediment penetration phase. FREEFALL is written in Pascal language for compilation by Turbo Pascal Version 3.0 (Borland, 1985a) on an IBM PC computer. Borland's Turbo Graphix Toolbox (Borland, 1985b) supports the graphic input and plotting routines. Hardcopy output is obtained by activating the computer's screen printing software. GRAFPLUS (Jewell, 1984) has been used by the author as the software to produce graphics printouts.

The model is derived from the assumption of a simple point mass system that falls under the influence of gravity and is retarded by drag effects. The model only allows for axial motions, and as such gives no predictions of vehicle or path stability. It is assumed that the body is released vertically from a submerged position in the water and falls downward with no angle of attack. This allows a straightforward calculation of the time-related values of acceleration, velocity, and distance traveled.

FREEFALL assumes that the geometry of the penetrator will be described by a cylindrical body, a tangent-ogive nose, a tapered tail, and any number of fins. The body may also truncate with no tail, or the tail may possess an extension in the form of a stinger. Although the program assumes that the nose is a tangent-ogive, conical noses and other rounded shapes have been represented by a tangent-ogive with the same ratio of length to body diameter and resulted in predictions that were within about 10 percent of measurements. A similar procedure for approximating a nonconical tail section with the frustum of a cone works equally well. Allowance is made for the quality of the surface finish and fairing of jointed surfaces through a user-defined roughness factor.

Once the program is initiated, the user is prompted to enter the dimensions of the penetrator and its weight in both air and water. These data

and the default values of other input data may then be corrected. Afterwards the code calculates the axial acceleration, velocity, and distance traveled in the water column, assuming that the submerged body is released from a nose-down position. Drag is adjusted as a function of Reynolds number and the kinematic viscosity of seawater. The latter is expressed as a function of temperature that is, in turn, related to depth. Tabular results of time, acceleration, velocity, depth, and drag of vehicle components are provided. The user may, if desired, adjust any of the input parameters and repeat the calculations. Once these iterations are completed, tabular results of critical values and a plot of velocity vs. depth are presented for the final configuration.

On completing the flight in the water column, the code proceeds to the burial portion of its calculations after prompting the user to check and adjust the default values of the soil constants. FREEFALL takes the results of the flight calculations and the soil constants as input to a calculation of axial soil-penetration dynamics. The soil constants allow soil strength and sidewall friction to vary linearly with depth and are adjusted for strain rate. The code calculates the accelerations, velocity, and depth of soil penetration as a function of time at 96 intervals after impact and presents the results in tabular form. At this point the user may choose to adjust the impact velocity or the soil constants and to repeat the penetration calculation. Once these iterations are finished, plots of the results against time and depth are presented.

After the burial iterations have been completed, the user may either terminate the session or return to the flight portion of FREEFALL and begin the cycle again by adjusting the input data. In this way some measure of optimization can be obtained for hypothetical configurations.

2. COMPARISON OF FREEFALL PREDICTIONS WITH EXPERIMENT

Data now exist from many seabed penetrator tests covering over 20 years of development. Some of these were performed by Sandia during the Marine Sediment Penetrator Program (Colp et al., 1975) and later during the Subseabed Disposal Program (Seabed Programs Division, 1983). A large number of experiments were also performed by members (including Sandia) of the international Seabed Working Group of the Nuclear Energy Agency of the Organization for Economic Cooperation and Development (Hickerson et al., 1987). Data from these ocean tests have been collected and compared with the predictions of FREEFALL to determine its accuracy.

Before presenting the comparison, a discussion of the data and its limitations is in order. Penetrator dynamics are measured by two methods:

- Onboard instrumentation senses and records accelerations. These data are then integrated to produce velocity and the distance traveled.
- Fixed-frequency acoustic transmitters are placed in the tail and monitored by hydrophones located near the launch point. The change in the frequency of the received signal, the Doppler shifted frequency, is a measure of the velocity of the penetrator. This information can be differentiated with time to obtain average acceleration or can be integrated to produce distance traveled.

Direct measurements of acceleration have never been made during the early stages of flight; therefore, these data are missing. Accurate acoustic measurements of initial velocities have not been made either, because of the inconvenience of placing the monitoring hydrophone at the launch point. Mathematical correction of initial velocity data has been attempted (Hickerson et al., 1987; Hickerson in ENEA, 1987), but is of questionable accuracy for the first few meters of flight.

Similar problems exist in defining the moment of impact between the penetrator and the bottom, resulting in some uncertainty as to the actual depth of penetration. Onboard accelerometers are usually sensitive enough to detect decelerations accumulating during penetration of the nose; however, it is usually not known how much nose penetration has occurred before impact is sensed. It is often assumed that the burial depth obtained from accelerometer measurements is shorter than the actual depth attained by about one-half the length of the nose. This might result in an over-correction for impact into a very stiff soil, or an under-correction for a very soft soil.

Acoustic Doppler measurements can be more ambiguous. Velocity changes during the first body length of penetration are often imperceptible, and the point of impact must then be deduced by observing when the Doppler signal suddenly changes strength, apparently due to the submergence of the tail transmitter beneath the sediment. This might occur as much as half a body length below the actual seafloor, since the hole left by the penetrator probably takes 25 to 50 ms to close behind the tail (Dzwileski and Karnes, 1982; Karnes in Burdett and Karnes, 1986). Consequently, acoustic measurements of penetration have an inherent inaccuracy of some fraction of one body length that probably tends towards underestimation of the burial depth.

Data from over 40 penetrator experiments have been collected and compared with FREEFALL predictions for the flight and burial. Table 1 lists the tests from which data have been taken. Table 2 lists the measured and predicted ranges of values for the penetrators in Table 1 and identifies the references pertaining to each. Figure 3 illustrates some of the shapes identified in the tables. In some cases, more than one unit of a given type was launched, often over different water depths or sediments. Not all tests collected data during the water flight, and not all collected burial data.

Figures 4 and 5 illustrate the comparison between predicted and measured impact velocity and burial depth for the penetrators just described. Measured impact velocities tend to lie within about 10 percent of predictions. Burial depths are more difficult to predict because of the limited amount of available soil data and the simplicity of the penetration

TABLE 1. PENETRATORS THAT HAVE BEEN USED TO VERIFY FREEFALL

<u>Penetrator</u>	<u>Institution</u>	<u>Date Tested</u>	<u>Water Depth (m)</u>	<u>Diameter (m)</u>	<u>Length (m)</u>	<u>Weight (kg)</u>	<u>Reference Given in Table 2</u>
BESA	Sandia	8/65	>200	0.229	4.47	632	1
MSP-1	Sandia	5/70	30-105	0.076	1.27	47	1,2
MSP-2	Sandia	11/70	60	0.076	1.52	48	1,2
ISP-1	Sandia	5/82	gun test	0.203	2.36	320	3
Type I	BRE/UK	3/83	5840	0.325	3.25	1780	4
ISP-3	Sandia	8/83	1700	0.203	3.40	468	5
ISP-Type 7	Sandia	3/84	5840	0.324	4.67	2360	6
ISP-Type 8	Sandia	3/84	5840	0.324	4.67	2665	6
Type I	BRE/UK	3/84	5840	0.325	3.25	1800	6
Type II	BRE/UK	3/84	5840	0.230	5.75	1800	6
Type III	BRE/UK	3/84	5840	0.325	3.60	1800	6
Type IV	BRE/UK	3/84	5840	0.325	3.25	2360	6
Type V	BRE/UK	3/84	5840	0.400	4.10	1800	6
Type VI	BRE/UK	3/84	5840	0.500	2.00	1800	6
2/3 Scale	CEA/Fr.	3/85	850	0.240	2.40	500	7
Full Scale	CEA/Fr.	7/85	5400	0.360	3.60	1900	8
Type I	BRE/UK	7/85	5400	0.325	3.25	1800	9
Type II	BRE/UK	7/85	5400	0.230	5.75	1800	9
Unit A	BRE/UK	8/85	250	0.415	2.71	1600	10
Unit B	BRE/UK	8/85	230	0.400	3.21	1900	10
ISP w/float	Sandia	4/86	390	0.324	4.67	2665	11
2/3 Scale	CEA/Fr.	4/86	390	0.240	2.40	500	7,8
Type I MOD	BRE/UK	10/86	5400	0.356	3.56	3200	12
Type I	BRE/UK	11/86	260	0.325	3.25	1800	13
Type IB	BRE/UK	11/86	260	0.382	2.85	1800	13
Type HM	BRE/UK	11/86	260	0.320	3.00	1550	13
MK-1A	Sandia	11/86	260	0.273	3.79	1475	14

TABLE 2. MEASURED AND PREDICTED VELOCITIES AND REFERENCES FOR THE PENETRATOR TYPES LISTED IN TABLE 1

Penetrator	Terminal or Impact Velocities:			Reference
	Measured (m/s)	Predicted (m/s)	Reference Number	
BESA	23-24	22-25	1	Thigpen (1971)
MSP-1	13-17	14-17	2	Colp, Caudle, and Schuster (1975)
MSP-2	24	24	2	Ibid
ISP-1	87	(gun- launched)	3	James (1983)
Type I (3/83)	49-53	50	4	Murray and Visintini (1985)
ISP-3	35-37	35	5	Talbert (1984)
ISP Type 7	50-52	51	6	Freeman and Burdett (1986)
ISP Type 8	53	54	6	Ibid
Type I	49-54	52	6	Ibid
Type II	50-54	54	6	Ibid
Type III	55	54	6	Ibid
Type IV	61-64	62	6	Ibid
Type V	45	39	6	Ibid
Type VI	48	45	6	Ibid
2/3 Scale	45	42	7	Hembise et al. (1987)
Full Scale	53-56	58	8	Boisson (1986)
Type I	53-54	52	9	Freeman et al. in ESOPE (1987)
Type II	54	54	9	Ibid
Unit A	25	26	10	Freeman in Istit.di Biologia (1985)
Unit B	24	28	10	Ibid
ISP w/float	47-48	46	11	Hickerson in Boisson (1986)
Type I mod	68	67	12	Freeman et al. (1987)
Type IB	37	42	13	Freeman et al. in ENEA (1987)
Type HM	29	32	13	Ibid
MK-1A	44-46	46	14	Hickerson in ENEA (1987)

TYPE	I,IA	II	III	IV	V	VI	VII	VIII
NAME	ESP	TALL BOY	CYRANO	HEAVY	JUMBO	FATMAN	SANDMAN I	SANDMAN II
L/D	10	25	11	10	10	4	14.8	14.8
KG	1800	1800	1800	2360	1800	1800	2230	2720
MAT.	ST.	ST.	ST.	ST.+Pb	ST.+CONCR	ST.	ST.	ST.+Pb
N ₀	4	2	2	2	2	2	2	2
INSTR.	I-ADSS	ADSS	ADSS	ADSS	ADSS	ADSS	EATS	EATS
	IA-LFTS						ADSS	ADSS

NOTE: ADSS - Acoustic Doppler
EATS - Explosive Acoustic Telemetry System
LFTS - Low Frequency Transponder System

Figure 3. Representative Penetrators Tested by the Seabed Working Group (Talbert, 1984; Freeman and Burdett, 1986).

Figure 4. A Comparison of Predicted vs. Measured Impact Velocities for Penetrators in Tables 1 and 2.

Figure 5. A Comparison of Predicted vs. Measured Burial Depths for Penetrators in Tables 1 and 2.

model. FREEFALL appears to slightly overpredict of burial depth (calculated from the measured impact velocity). This may be partly due to a tendency of the measurement techniques to underestimate burial depth. Regardless of the cause of errors, the results are generally within +10% to -15% of the predictions.

Several of the penetrators launched in shallow water did not reach terminal velocity before impacting the bottom. FREEFALL predicted these impact velocities within 10%, an indication that the code does properly describe the acceleration phase of the water flight. This is further supported by the measurements of velocity taken from a MK-1A penetrator and compared with FREEFALL predictions in Figure 6 (Hickerson in ENEA, 1987).

If soil properties are described well enough by the soil model, the acceleration profile and velocity decay during penetration are predicted accurately. Figure 7 illustrates the agreement between predicted and measured velocities for a BRE Type HM penetrator and a Sandia MK-1A penetrator during the "HOCUS" Campaign near Antibes, France. The soil at this site was a very homogeneous, silty clay that linearly increased in strength with depth. In situ measurements of soil penetration resistance, q_c , and side-wall friction, f_s , were available down to 30 m (ENEA, 1987). Uniformity such as this is frequently not the case, however, because sediments often tend to be layered with material of different particle sizes. This can be compensated for by choosing soil constants that tend to average the properties of the layers. The predicted acceleration-time profile will not, however, relate well to the actual measurement made by the penetrator. An example is shown in Figure 8 in which an ISP penetrator (Table 1) encountered a known sand layer between 2 and 4 m deep and a suspected layer at about 14 m in a silty clay sediment (Hickerson in Boisson, 1986). This can be compared with the results in Figure 9 for a similar ISP vehicle that penetrated the fairly uniform sediments at the deep ocean site in the Nares Abyssal Plain (Freeman and Burdett, 1986).

Figure 6. A Comparison of Predicted vs. Measured Velocities after Launch for a MK-1A Penetrator.

Figure 7. A Comparison of Predicted vs. Measured Velocities during Soil Penetration for Sandia MK-1A and BRE Type HM Penetrators.

Figure 8. Decelerations during Penetration for an ISP Penetrator Encountering Sand Layers Compared with FREEFALL Predictions for a Uniform Sediment of Similar Average Strength.

Figure 9. Decelerations during Penetration for an ISP Penetrator in the Relatively Uniform Sediments of the Nares Abyssal Plain Compared with FREEFALL Predictions.

3. THE WATER FLIGHT MODEL

3.1 Mathematical Basis

The water flight model is a simple point-mass formulation that seeks solution to a one-dimensional statement of Newton's equation of motion:

$$m \frac{dv}{dt} = F_{\text{gravity}} - F_{\text{drag}} \quad (1)$$

where

m = mass of the vehicle

v = velocity

t = time

F = force, either due to gravity or hydrodynamic drag.

The model assumes only axial motion and allows for no trajectory deviations created by angle of attack, lift effects, or center-of-gravity perturbances.

Gravity forces accelerating the vehicle are created by its buoyant weight. Drag is found by summing the contributions of all n body components:

$$F_{\text{drag}} = \frac{1}{2} \rho v^2 \sum_{i=1}^n (C_D A)_i \quad (2)$$

where

ρ = the density of seawater, corrected for temperature

C_D = the drag coefficients of the n individual components

A = the appropriate area of each component.

Temperature, which affects the density and kinematic viscosity of seawater, is assumed to decline parabolically from a user-specified surface temperature to 2°C at a depth of 2500 m and then to remain constant to the bottom. Figure 10 shows a comparison between the temperature algorithm and a reference temperature profile over the Nares Abyssal Plain.

Figure 10. The Reference Temperature Profile for the Water Column over the Nares Abyssal Plain (Fuglister, 1960) Compared with the Temperature Algorithm in FREEFALL.

Drag coefficients are evaluated by semiempirical methods described by Hoerner (1965). Drag areas (the product of drag coefficient and the appropriate component area) are separated into pressure drag and surface-friction drag contributions and summed for each component. As indicated in Eq (2), component drag areas are then summed to give a total drag area for the penetrator.

Equation (1) is solved stepwise from time equals zero in 96 discrete time steps. After each step, the instantaneous acceleration is calculated and then numerically integrated with time to give velocity that, in turn, is numerically integrated to give distance traveled. Time steps, dt , are sized according to

$$dt = K (i^2) \quad (3)$$

where

K = a time constant defined by the user-specified flight time (typically 10 to 120 s) divided by 96^2

i = the step number (from 1 to 96).

This algorithm creates the smallest time increments at launch when conditions change most rapidly and increases the time step as the flight approaches equilibrium. As the penetrator falls, the code checks to see if the specified water depth has been exceeded, and if so back-calculates the impact conditions. These data are then stored as input to the penetration phase of the flight.

3.2 Drag Calculations

Accurate estimates of drag are required to calculate terminal and impact velocities of penetrator shapes and to correctly identify the contributions of each component. FREEFALL attempts to account for all major drag contributions and adjusts its calculations for the effect of temperature on water density, viscosity, and Reynolds numbers.

Penetrator drag is assumed to arise from the six vehicle components listed in Table 3 and is further broken down into pressure and skin friction

TABLE 3. SUMMARY OF DRAG CONTRIBUTIONS.

<u>Component</u>	<u>Pressure or Friction Drag</u>	<u>Related Component Area</u>
Nose	Pressure	Body cross-sectional area
	Skin friction	Nose surface area
Body tube	Skin friction	Body surface area
Tail cone	Pressure	Body cross-sectional area
	Skin friction	Tail-cone surface area
Stinger	Skin friction	Stinger surface area less base
Base	Pressure	Base cross-sectional area
Fins	Skin friction	Fin surface area

contribution. Pressure drag is created by the nose, tail cone, and base. Drag coefficients for each component are multiplied by the appropriate cross sectional area to produce the resultant component drag area. Skin friction drag is calculated for all wetted surfaces except the base of the penetrator, which lies perpendicular to the vehicle axis. Skin friction drag coefficients are multiplied by the surface area over which friction occurs to arrive at the appropriate drag area. Further modification of frictional drag is allowed by introducing a surface roughness factor that attempts to compensate for surface finish and the amount of fairing used to blend body transitions.

Surface friction drag coefficients are calculated as a function of Reynolds number, which is related to velocity through

$$R = \frac{v l \rho}{\mu} \quad (4)$$

where

l = a characteristic length, taken to be the length of the body tube

μ = the viscosity of seawater, corrected for temperature.

At low velocities and low Reynolds numbers, it is assumed that a laminar boundary layer exists and persists up to a velocity associated with a Reynolds number of 2×10^5 . Boundary layer transition to turbulent flow is assumed to occur at this point, converting to fully turbulent flow at a Reynolds number of 10^7 . Hoerner (1965) presents the observed relationship between friction drag coefficient for smooth flat plates and Reynolds number to which the following equations were fit and applied, in accordance with common practice, to friction surfaces of the components:

Laminar boundary layer ($R_n < 2 \times 10^5$)

$$C_f = \frac{1.328}{\sqrt{R_n}} \quad (5)$$

Turbulent boundary layer ($R_n > 10^7$)

$$C_f = \frac{0.046}{R^{1/6}} \quad (6)$$

Between these two regimes, where the transition from laminar to turbulent flow occurs, the value of C_f is assumed to be constant at 0.003. The resulting curve for the relationship of the friction drag coefficient over all three regimes is shown in Figure 11.

Average surface roughness is taken into account by multiplying the skin friction results by a roughness factor. Although this does not accurately represent the contribution of the effects of surface roughness with increasing Reynolds number, the procedure is valid for compensating for

Figure 11. Relationship between the Friction Drag Coefficient and the Reynolds Number Calculated by FREEFALL.

surface effects once velocities approach terminal speed and Reynolds numbers exceed 10^6 (Hoerner, 1965). The user must specify the value of the surface roughness factor, which typically varies from 1.0 for ideally smooth bodies to 1.2 for rough-machined, partially faired bodies, to 1.5 or higher for unfaired penetrators with very rough surfaces, ill-fitting joints, or surface protuberances. The User's Guide in Section 5 provides further information on this quantity.

Pressure drag for the nose and tail was obtained by choosing a curve that fit turbulent pressure drag data for afterbodies, as presented by Hoerner (Section 20-2, 1965):

$$C_p = \exp\left(\frac{-0.748 L}{D} - 3.17\right) \quad (7)$$

where

L = the length of the nose or tail

D = the diameter of the penetrator body.

Pressure drag created by the base was approximated by a curve fitted to data presented in Murray and Visintini (1985) for truncated cones ranging in angle from 10° to 25° , and then subtracting out the apparent afterbody contribution of 0.02. The resulting equation is stated as

$$C_p = 0.004 (180/\pi) \arctan\left(\frac{D-B}{2 L}\right) - 0.02 \quad (8)$$

where

B = the diameter of the base

L = the length of the tail cone.

This gives a drag contribution for the base that increases as the tail becomes blunter, reaching a maximum of 0.14 for a flat base with no tail cone. If a stinger is present, it is assumed that the influence of the tail cone on base drag [Eq (8)] decreases linearly with stinger length until it is negligible for a stinger of length equal to four base diameters:

$$C_p = C_{Eq (8)} + \frac{S}{4 B} (0.14 - C_{Eq (8)}) \quad (9)$$

where S is the stinger length. This results in a base drag coefficient of 0.14 for stingers of four base diameters in length, which is also assigned to the base of longer stingers.

4. THE BURIAL MODEL

The burial model parallels the functions of the water flight model. Newton's equation of motion is formulated for soil penetration, the force opposing motion is broken down into component contributions, and then time-stepped solutions are found for the variables related to acceleration, velocity, and depth of penetration:

$$m \frac{dy}{dt} = F_{\text{gravity}} - F_{\text{resistance}} \quad (10)$$

where

$$F_{\text{resistance}} = F_{\text{nose}} + F_{\text{body}} + F_{\text{fins}} + F_{\text{tail}} + F_{\text{stinger}} + F_{\text{base}}. \quad (11)$$

The burial model assumes that soil penetration by a seabed penetrator is analogous to penetration by a cone penetrometer such as that developed and employed by Delft Geotechnics (formerly the Delft Soil Mechanics Laboratory, in the Netherlands). The Delft piezocone shown in Figure 12 is typical of this type of instrument. A 60° cone at the tip is instrumented to measure areal bearing resistance, q_c (kg/cm²). Directly behind the cone is an instrumented steel sleeve that directly measures the sidewall friction stress, f_s (kg/cm²). In practice, the piezocone is mounted at the end of an assembly of rods and driven into the sediment by a hydraulic system, typically at a rate of 2 cm/s. Continuous measurements of q_c and f_s are recorded during the test, and penetrations up to 50 m are readily obtained.

The models describing cone penetration are well developed and have been shown to be applicable to objects the size of penetrators (Sanglerat, 1972; Schmertmann, 1975; Durgunoglu and Mitchell, 1975). Furthermore, these models and the experiments of Young (1981) show that cone angle has little effect on sediment penetration, allowing the direct use of measurements of bearing resistance, q_c , from standard 60° cones to predict the behavior of penetrators of varying nose geometries. Since sidewall friction, f_s , is measured by a steel sleeve on the piezocone, should also be directly applicable to the surfaces of a metal penetrator, it remains only to add the

Figure 12. The Delft Piezocone (ENEA, 1987; Hickerson et al., 1987; Sanglerat, 1972).

algorithms that express the variation of these two soil constants with velocity and depth, and to include the dynamic pressures that are induced at the tail and base.

Gravitational forces tending to accelerate a penetrator are created by its buoyant weight in the sediment. Resisting forces opposing its progress are assumed to have three sources:

- Bearing forces acting on component cross sections normal to the axis
- Frictional forces acting tangentially to component surfaces
- Dynamically induced negative pressures at the tail cone and the base acting normal to the axis.

Bearing forces are calculated by directly multiplying the local value of q_c , adjusted for strain rate, by the cross-sectional area. As mentioned earlier, q_c , is expressed in the model as varying linearly with depth, according to

$$q_c = q_{c(\text{surface})} + mz \quad (12)$$

where m is the user-specified average slope of the q_c -vs.-depth measurements and z is the distance below the sediment surface. To this is added a velocity-dependent, strain-rate adjustment taken from a curve fitted through piezocone data for a saturated cohesive clay tested by Smits (1987):

$$q_{c2} = q_{c1} + 3.13 (v_2^{0.2} - v_1^{0.2}) \quad \text{kg/cm}^2 \quad (13)$$

The subscripts indicate values obtained at penetration velocities 1 and 2. The model assigns the initial values, v_1 and q_{c1} , to the cone penetrometer data and assumes that it was collected at the standard rate of 2 cm/s. A significant improvement in the accuracy of the code might be obtained if strain rate data for the sediment of interest were used, but these data are rarely available at this time.

The sidewall friction factor, f_s , is defined as a fraction of the penetration resistance, q_c . This fraction ranges from 0.02 to 0.07, as

taken from Sanglerat (1972) and defined in Table 4 of the User's Guide in Section 5. The total sidewall friction on the penetrator surfaces is obtained by multiplying the local value of f_s by the component surface area.

The nose, body, and fins make use of the bearing and friction formulations described above. The resisting forces of these components are expressed as follows:

$$F_{\text{nose}} = q_c A_{\text{xsexn}} + f_s A_{\text{nose}} \quad (14)$$

$$F_{\text{body}} = f_s A_{\text{body}} \quad (15)$$

$$F_{\text{stinger}} = f_s A_{\text{stinger}} \quad (16)$$

$$F_{\text{fins}} = q_c A_{\text{bear}} + f_s A_{\text{fins}} \quad (17)$$

where

A_{xsexn} = the cross-sectional area of the penetrator

A_{nose} , A_{body} , etc. = the surface area of the components

A_{bear} = the bearing area of the fins created by their radial extension beyond the diameter of the body tube.

The resisting force created by the base is analogous to the base drag formulation described in the section on water flight modeling. Suction on the tail is approximated by assuming that the sediment behaves as a dense fluid describable by the Bernoulli principle for incompressible flow (Streeter, 1962). In the manner of Dzwilewski and Karnes (1982; see also Karnes in Burdett and Karnes, 1986), this force is given by

$$F_{\text{base}} = (\rho_s v^2/2 - \rho_s g z) A_{\text{base}}, \quad (18)$$

where ρ_s is the density of the sediment and g is the acceleration of gravity.

The resistance forces created by the tail cone arise from the friction of the sediment and the suction force described above. Given that Eq (18) describes the maximum suction created by a flat base, a tapered tail would reduce the pressure gradient and create less suction per unit area. Assuming that suction forces will scale in the same manner as the drag of tail sections described by Hoerner (1965, pp. 20-2), Eq (18) is modified for tail cones to reflect the degree of taper and the presence of sidewall friction, resulting in

$$F_{\text{tail}} = A_{\text{tail}}^* (\rho_s v^2 / 2 - \rho_s g z) \left(1 - \frac{L}{2.5 D} \right)^{1/2} + f_s A_{\text{tail}}, \quad (19)$$

where L is the length of the tail cone and A_{tail}^* is the projected area of the tapered portion of the tail cone.

The relations given above describe the forces acting on a fully submerged penetrator. During the first stages of penetration, these forces must be modified to reflect the transition from water flight to soil entry and the diminishing influence of water drag. Except for the tail section and base, this is done by adding soil forces and eliminating water drag in proportion to the amount of submergence of the individual components. Tail and base suction are treated slightly differently. The results of Dzwilewski and Karnes (1982) indicate that complete closure of the sediment behind the tail may take three to four body-diameters of travel; therefore, the sediment suction forces are modeled to build linearly from zero to the values given in Eqs (18) and (19) during the first three diameters of tail submergence.

5. A USER'S GUIDE TO FREEFALL.PAS

5.1 General Guidance

Certain restrictions apply to the judicious use of FREEFALL. Because the drag contributions of various vehicle components are based on theory and on semiempirical data for ideal shapes, one would not expect FREEFALL to be accurate for penetrator geometries that deviate significantly from the input geometry that is, a cylindrical body truncated by an ogival nose and flat base or conical tail. For example, to be analyzed by FREEFALL, an optimized penetrator with the continuous contours described by Hembise et al. (1987) must be approximated by a short cylindrical shape with a long nose and tail.

Similarly, the soil model used for the sediment penetration phase assumes the seafloor to be a cohesive clay of linearly increasing strength. Thus, the accelerations and burial depth predictions are only as accurate as this formulation allows. Layered sediments of differing strengths or sandy soils are not accurately represented by this model. Sand layers are especially troublesome and will create accelerations far in excess of those predicted by such a soil model (Figure 9).

The soil model requires soil-bearing and friction constants, q_c and f_s , as would be measured in situ by the Delft piezocone (Sanglerat, 1972). This type of data, although thought to be the most accurate for making penetration predictions (Hickerson, and Freeman et al. in ENEA, 1987), is rarely available for seafloor soils. Instead, soil strength is usually stated as undrained shear strength, c_u , measured in the laboratory on samples recovered from cores. According to Sanglerat (1972), the conversion between these two is approximately $q_c = 10 c_u$ when both are expressed in identical units. This is supported by data taken from shear tests on cores and penetrator experiments (Freeman et al., 1987). In deep ocean sediments, bearing resistance, q_c , expressed in kg/cm^2 is about $0.1 c_u$, expressed in kPa. This conversion has been found adequate for other clays and silty clays, as indicated by the agreement between predictions and experiments presented in Table 2 and Figure 5.

Once q_c has been determined, the value of the friction constant, f_s , is readily established for different soil or sediment types by the use of Table 4, taken from Sanglerat (1972). For example, the predicted value of $f_s = 0.05 q_c$ taken from the table has been found appropriate for deep ocean clays of the Nares and Great Meteor East areas (sediment descriptions in Shephard et al, 1987), while the near-shore silty clays near Antibes, France, were found to have a value of $f_s = 0.02 q_c$ by direct piezocone measurements (Smits in ENEA, 1987).

TABLE 4. RELATIONSHIP BETWEEN f_s AND q_c AS DETERMINED BY PARTICLE SIZE AND SOIL TYPE (After Sanglerat, 1972).

f_s/q_c	Soil Type	Percentage of Particles Under 16 μ m in Diameter
0.012 to 0.014	Coarse sand	0
0.015 to 0.017	Fine sand	0
0.018 to 0.025	Silty sand	5
0.025 to 0.032	Silt-clay-sand	15 to 35
0.032 to 0.034	Loam	35 to 45
0.035 to 0.045	Clay-loam	45 to 75
0.045 to 0.055	Clay	75 to 95
0.055 to 0.070	Fine clay	95 to 100

The strain rate adjustment the code makes to soil constants may not be accurate for a wide range of clays, and is probably not accurate for sandy soils. The piezocone data on which it is based were provided by Smits (1987) and were obtained from a saturated French clay tested at penetration rates of 0.001 to 1 m/s. Although the algorithm appears adequate for most sediments, predictions for coarser sediments should be viewed with caution.

General surface finish and the degree of fairing between body contours are accounted for by the use of a roughness factor. This is specified by the user with the aid of the guidelines given in Table 5. Values given in Table 5 have been arrived at based on data presented by Hoerner (1965) and by analyzing data from the penetrator tests discussed earlier. A typical well-designed penetrator will exhibit a roughness factor between 1.1 and

1.2. Optimized vehicles will produce a roughness factor from 1.0 to 1.1. Very crude vehicles have been found to exhibit a roughness factor of 1.5.

TABLE 5. GUIDANCE FOR SELECTING ROUGHNESS FACTORS.

<u>Roughness Factor</u>	<u>General Surface Condition</u>
1.00	Polished, mirror-like surfaces, faired transitions between all body components, no joints showing
1.05	Polished surfaces, faired transitions between all body components, joints well fitting
1.10	Finish-machined and painted surfaces, faired transitions between all body components, small surface defects
1.20	Painted rough-machined or hot rolled and sand-blasted surfaces, faired nose and tail transitions, joints showing slight mismatches, surface irregularities due to instrumentation ports, fins bolted on without fairings to tail or body surfaces
1.30	As above, but with more surface irregularities such as protuberances or large joint mismatches
1.50	Rough, dirty, unpainted and rusted surfaces, unsmoothed weldments, unfaired transitions, blunt fin edges, gross protuberances or large unfaired ports, large joint mismatches

5.2 Operational Hints

The code is not completely protected from invalid data entry by the user. If this occurs for example, by entering a character, a zero, or a negative number where a positive number is expected the code may fail. In that event, Turbo Pascal automatically sends the program into the "edit" state and indicates where the error was detected within the code. If the

cause was improper data entry, then the user ignores this condition, exits the editor, and restarts the run.

Penetrators are sometimes required to pull a communication cable or wire from the surface. An accurate prediction of impact velocity can be obtained by entering the cable as a stinger equal in diameter to the base and long enough to create the same surface area as the cable at the time of impact with the bottom. This will result in a good prediction of the impact velocity, although the early velocity-time profile will probably be inaccurate. If the cable is dispensed from the penetrator as it falls, then no pulling forces arise and it can be disregarded in the calculation.

Velocities of penetrators that tow floats have also been found by the following procedure. The float is modeled as a second penetrator and given an apparent buoyant weight that represents a first guess at a towing force required to reach the anticipated terminal velocity. The calculated terminal velocity of the float is then compared with the calculated terminal velocity of the towing penetrator after subtracting this same towing force from its buoyant weight and modeling the tow cable as a stinger as discussed above. Iterative adjustments are then made in the towing force (and apparent buoyant weights) of each vehicle until the terminal velocities match. Because deceleration after impact with the bottom is so rapid, the presence of the tow cable and float adds little additional drag and can be neglected for the penetration calculation.

It has been found that if very slow vehicles are analyzed for very long flight times, a program failure may occur near the end of the water flight calculations. This is probably because the time steps as calculated by the parabolic algorithm become very large. It appears that small changes in acceleration are then able to create large changes in other variables as a result of the size of the time step, and occasionally cause the appearance of negative or out-of-range numbers. A possible solution to this problem would be to change the constant "maxnum" appearing in the initial CONST declaration for the code from 96 to a much larger number, such as 300. This would keep the time steps small, but would significantly increase the calculational time. Because the procedures that print out the tabular results

print every sixth value, this interval would also have to be increased to prevent these tables from filling more than one screen.

It was mentioned earlier that penetrators with continuously changing contours must be approximated by a similar shape with an ogive nose, a tubular body, and a conical tail. The tubular body section must be at least 25 to 30% of the total length, because the code calculates the Reynolds number for the friction drag coefficient based on the length of this body section. Calculated drag areas and the frontal area drag coefficient will tend to be higher than actual measurements. For example, FREEFALL applied in this manner to the penetrator of Hembise et al. (1987) predicts a frontal area drag coefficient of 0.10 to 0.11; actual tank tests have shown it to be 0.08. This results in terminal velocity predictions that tend to be slightly lower than measurements.

Occasionally the code will predict that a very streamlined penetrator with a very large ogive radius (a slender, tapered nose) will accelerate briefly during the initial stages of penetration into a very soft sediment. This is due to the inability of the code to model the drag effects occurring during the transition from water flight to the entry of the "soupy" bottom material. In these cases the code tends to overpredict burial depths, usually by a distance that is about equal to the distance over which this anomalous acceleration occurs.

5.3 Running the Code

In addition to the code itself, the user needs the following files available from Borland International (1985a,b):

- Turbo Pascal Compiler Version 3.0 or later
- Turbo Graphix Toolbox Version 1.0 or later.

The FREEFALL.PAS makes use of the compiler and of the following "include" files from the Toolbox:

- typedef.sys
- graphix.sys
- kernel.sys
- windows.sys
- findwrld.sys
- axis.hgh
- polygon.hgh.

These files are called in this order in the first statements of the program.

Once the program is called, a series of screens appears, as shown in Appendix A. The following describes the user/program interactions during a typical run. Pressing any key will advance the program to the next screen once a given function is completed.

Screen 1 is the title screen giving the name and uses of the program.

Screen 2 specifies that dimensions to be provided later are to be entered in millimeters, and the fin area in square meters

Screen 3 first displays a graphic illustration of a penetrator separated into components. The user is then prompted to provide the nose ogive radius (in mm), followed sequentially by the dimensions for the other features. Fin area should be given in square meters for one side of one fin only. The surface roughness parameter is the last item requested and can be estimated based on the data in Table 5. The example in Appendix A shows the screen after all data have been entered.

Screen 4 lists a summary of the data entered and the default settings for the other parameters needed for the water flight calculations. The program prompts the user to check and correct any item in the list by entering its identification number and then the new value after the next prompt.

Changes may be made to as many items as desired. Entering "0" starts the calculations for the water flight.

Screen 5 presents an abbreviated summary of the drag calculations for each component. Every sixth value is presented from the 96 data points calculated. The stinger drag is added to the data in the column for the tail section.

Screen 6 summarizes the acceleration, velocity, depth, and total drag for the vehicle by listing every sixth value calculated and allows the user to return to Screen 4 and modify the input data if desired. Otherwise, the program continues to Screen 7.

Screen 7 Informs the user (by means of a transient screen) that Screen 8 will summarize the input data, and that the user will be allowed to provide a title and date as a heading to the data appearing in Screen 8.

Screen 8 first prompts the user for a title and then a date. This may be omitted by striking the "enter" or "return" key. The screen then fills with the last input data as in Screen 4.

Screen 9 summarizes the component area calculations and water column settings and gives the principal results for drag coefficient and impact conditions.

Screen 10 asks for the title of the velocity versus depth curve to be presented in the next screen. This may be omitted by striking the "enter" or "return" key.

Screen 11 presents a graph of velocity versus depth. A legend lists the maximum velocity shown on the graph (usually not the impact velocity, because impact may not have been achieved if the time of flight was too short, or impact may have

occurred before the flight time elapsed and while the velocity was still changing).

Screen 12 presents the input values for the burial calculation. If impact did not occur because the flight time was too short, an alternate screen appears informing the user of this and stating that burial calculations will be omitted. The user is then given the option of terminating the session or returning to Screen 4 of the water flight section of the code. If Screen 12 appears, the user may adjust any of the input parameters for the burial calculation, including the impact velocity. Default values are given for the fin height above the body and the fin thickness that may also be modified. A table in the upper third of the screen provides a reminder of the typical ranges for the soil parameters needed by the code.

Screen 13 presents results of the burial calculations as a table of acceleration, velocity, depth, and time after impact. The final burial values are listed at the bottom of the screen. If the penetration time selected in Screen 12 was too short and the vehicle did not come to rest, then the final values appear as zeros and subsequent graphic screens appear blank because of data out of range. Because about 15 seconds are required for the calculations on an IBM AT (expect a longer time for a PC), an interim screen appears during the calculational period inform the user that the program is calculating and to wait for the results.

Screen 14 asks if the user wishes to run another penetration case. If the answer to the prompt is "yes" (Y), then the code returns to Screen 13 and the input data can be adjusted. If the answer is "no" (N), then the code proceeds to the graphic screens.

Screen 15 asks for the title of the plot of burial data versus time. This may be omitted by pressing "enter" or "return."

Screen 16 presents the plots of acceleration, velocity, and depth versus time after impact. If these data are negative because of data entry errors, the screen appears blank. Striking the "enter" or "return" key causes the code to proceed to the next screen.

Screen 17 asks for the title of the plot of burial data versus depth of penetration. This may be omitted by striking the "enter" or "return" key.

Screen 18 presents a plot of acceleration and velocity versus depth of penetration. A legend gives the burial depth attained and the maximum acceleration recorded.

Screen 19 asks if the user wishes to return to Screen 4 to begin a new flight and penetration case or if the user wishes to terminate the session. Answering "yes" (Y) allows the user to repeat the whole series of calculations from Screen 4 onward. Answering "no" (N) terminates the code and returns the user to Turbo Pascal system level.

Any of the text screens may be copied to the printer with the standard computer "Print Screen" utility. Copying the graphic screens requires graphic screen-printing software such as GRAFPLUS by Jewell Technologies (1984) if the computer operating system does not permit direct graphic screen printing.

REFERENCES

- Boisson, J. Y. (1986). ACIS: Rapport de Mission, 14-19 Avril 1986 (OCDE/AEN Seabed Working Group). Reference 192/86/DPT/UGRA/ARMOR/JYB/CD, Commissariat al'Energie Atomique, Gif-sur-Yvette, France. 74 pp.
- Borland International (1985a). Turbo Pascal Version 3.0 Reference Manual. Borland International Inc., 4585 Scotts Valley Drive, Scotts Valley, CA, 95066. 376 pp.
- Borland International (1985b). Turbo Graphix Toolbox Version 1. Borland International Inc., 4585 Scotts Valley Drive, Scotts Valley, CA, 95066. 256 pp.
- Burdett, J. R. F. and C. H. Karnes (1986). Report of the Fifth Interim Meeting of the Seabed Working Group Engineering Studies Task Group. SAND85-1563, Sandia National Laboratories, Albuquerque, NM. 46 pp.
- Calloway, T. M. (1984). Depth of Penetration Achieved by Instrumented Seabed Penetrators During Initial Systems Demonstration Tests. SAND83-0060, Sandia National Laboratories, Albuquerque, NM. 19 pp.
- Colp, J. L., W. N. Caudle, and C. L. Schuster (1975). "Penetrator System for Measuring In Situ Properties of Marine Sediments." IEEE/MTS Oceans '75 Proceedings, September 22-25, 1975, p.405.
- Durgunoglu, H. T. and J. K. Mitchell (1975). Static Penetration Resistance of Soils: I - Analysis; II - Evaluation of Theory and Implications for Practice. In In Situ Measurement of Soil Properties, Volume I, Proceeding of the Conference, June 1-4, 1975, North Carolina State University. Published by the American Society of Civil Engineers, New York. 554 pp.

- Dzwilewski, P. T. and C. H. Karnes (1982). Dynamic Hole Closure Behind a Deep Ocean Sediment Penetrator. In Subseabed Disposal Program Annual Report, January to September, 1981, Vol. II, Appendices Part 2 of 2, pp. 335, SAND82-0664, Sandia National Laboratories, Albuquerque, NM. 972 pp.
- ENEA (1987). "HOCUS" Cruise Report. ENEA Environmental Studies Series, La Spezia, Italy, in press.
- The ESOPE Scientific Party (1987). The ESOPE Cruise Report. To be published by the Nuclear Energy Agency of the Organization for Economic Cooperation and Development for the Seabed Working Group, Paris. 300 pp.
- Freeman, T. J. and J. R. F. Burdett (1986). Deep Ocean Model Penetrator Experiments. EUR 10502, Commission of the European Communities, Luxembourg. 92 pp.
- Freeman, T. J., N. Cooper, and A. K. R. Bromley (1987). The Results of Seatrials of Instrumented Penetrators 1983-86. Department of the Environment, UK, in press.
- Fuglister, F. C. (1960). Atlantic Ocean Atlas of Temperature and Salinity Profiles and Data from the International Geophysical Year of 1957-1958. Woods Hole Oceanographic Insitute, Woods Hole, MA.
- Hauser, G. C. and J. Hickerson (1987). An Explosive Acoustic Telemetry System for Seabed Penetrators. SAND87-0952, Sandia National Laboratories, Albuquerque, NM. 30 pp.
- Hembise, O., P. Guevel, P. R. Bardey, J. L. Viaud, J. Y. Boisson, F. Vacherand (1987). Identification of Marine Soils by Ballistic Penetro-metry. Paper OMAE 87-554 presented at the Sixth International Symposium on Offshore Mechanics and Arctic Engineering, Houston, March 1987.

- Hickerson, J., T. J. Freeman, J.-Y. Boisson, N. J. Murray, F. Gera, H. Nakamura, J. D. Nieuwenhuis, K. H. Schaller (1987). Subseabed Disposal of High-Level Waste: Volume 4: Engineering. Nuclear Energy Agency of the Organization for Economic Cooperation and Development, Paris, in press; also published as SAND87-2223, Sandia National Laboratories, Albuquerque, NM. 240 pp.
- Hinga, K. R., G. R. Heath, D. R. Anderson, and C. D. Hollister (1982). "Disposal of High-Level Radioactive Wastes by Burial in the Sea Floor," Environmental Science and Technology. January. p.28.
- Hoerner, S. F. (1965). Fluid-Dynamic Drag. Published by the Author, from Hoerner Fluid Dynamics, P.O. Box 342, Brick Town, NJ. 200 pp.
- Instituto di Biologia del Mare (1985). Relazione Sulla Campagna "Bannock 1985": Fossa Jabuka. National Research Council, Venice, Italy.
- James, L. T. (1983). Gun-Launched Instrumented Seabed Penetrators: Initial Field Tests of the ISP-1 and ISP-2 Systems. SAND83-0095, Sandia National Laboratories, Albuquerque, NM, 19 pp.
- Jewell Technologies (1984). GRAFPLUS User Manual. Jewell Technologies Inc., Seattle, WA. 31 pp.
- Murray, C. N. and L. Visintini (1985). Parametric Analysis of Performances of Free-Fall Penetrators in Deep-Ocean Sediments. IEEE Journal of Oceanic Engineering, Vol. OE-10, No. 1, January. p.38.
- Sanglerat, G. (1972). The Penetrometer and Soil Exploration. Elsevier Publishing Company, New York. 464 pp.
- Schmertmann, J. H. (1975). Measurement of In Situ Shear Strength. In In Situ Measurement of Soil Properties, Volume II, Proceedings of the Conference, June 1-4, 1975, North Carolina State University. Published by the American Society of Civil Engineers, New York. 393 pp.

Seabed Programs Division (1983). The Subseabed Disposal Program: 1983 Status Report. SAND83-1387, Sandia National Laboratories, Albuquerque, NM. 179 pp.

Shephard, L. E. et al. (1987). Feasibility of Disposal of High-Level Radioactive Wastes into the Seabed: Volume 4: Geoscience Characterization. Nuclear Energy Agency of the Organization for Economic Co-operation and Development, Paris, in press. Also published as SAND87-1913, Sandia National Laboratories, Albuquerque, NM.

Smits, F. (1987). Personal communication of penetrometer data taken on a French clay covering penetration rates from 0.001 to 1 m/s.

Streeter, V. L. (1962). Fluid Mechanics. McGraw-Hill Book Company Inc., New York. 555 pp.

Talbert, D. M. (1984). Report of the Fourth Interim Meeting of the Seabed Working Group Engineering Studies Task Group, 3-6 October 1983, Rijks Geologische Dienst, Haarlem, The Netherlands. SAND83-2333, Sandia National Laboratories, Albuquerque, NM. 59 pp.

Thigpen, L. (1971). Water-Entry Technology - A Review. SC-DR-710196, Sandia Laboratories, Albuquerque, NM. 19 pp.

Young, C. W. (1981). An Empirical Equation for Predicting Penetration Depth into Soft Sediments. Oceans 81, Vol. II, pp. 674-677.

APPENDIX A

Principal FREEFALL Video Screens

SCREEN 1:

WELCOME TO FREEFALL

A Seafloor Penetrator Program

To Calculate:

Reynolds Numbers

Drag Coefficients

Drag Areas

Freefall Velocities

Sediment Penetration

press "enter" to continue

SCREEN 2:

The next screen will prompt you for
BASIC PENETRATOR DIMENSIONS

* Please provide linear dimensions
in MILLIMETERS !

* Give the "FIN AREA"
in SQUARE METERS for
one surface of one fin only

press "enter" to continue

SCREEN 3:

SCREEN 4:

YOU HAVE ENTERED THE FOLLOWING DATA,

1	Ogive Radius:	2030.0 mm
2	Body Diameter:	273.0 mm
3	Nose Length:	723.0 mm
4	Body Length:	2604.0 mm
5	Tail Cone Length:	464.0 mm
6	Stinger Length:	0.0 mm
7	Tail Base Diameter:	102.0 mm
8	Fin Area:1 fin,side:	0.0786 sq. m
9	Number of Fins:	4
10	Weight (dry):	1476 kg
11	Buoyant Weight:	1308 kg
12	Roughness Factor:	1.20
13	Maximum Fall Time:	10.0 sec
14	Surface Water Temp:	20 C
15	Water Depth:	1000 m

ENTER THE NUMBER OF THE VALUE YOU WISH TO CHANGE
IF NO CHANGES, ENTER " 0 "
>>> 15
ENTER THE NEW VALUE : 260

SCREEN 5:

DRAG AREAS FOR COMPONENTS							
VELOCITY (m/s)	R No.	TOTAL (sq.m)	NOSE (sq.m)	BODY (sq.m)	FINS (sq.m)	TAIL (sq.m)	BASE (sq.m)
0.38	9.46E+05	0.0139	0.0018	0.0080	0.0023	0.0016	0.00018
1.53	3.78E+06	0.0139	0.0018	0.0080	0.0023	0.0016	0.00018
3.44	8.45E+06	0.0139	0.0018	0.0080	0.0023	0.0016	0.00018
6.09	1.49E+07	0.0136	0.0018	0.0079	0.0022	0.0016	0.00018
9.44	2.29E+07	0.0127	0.0017	0.0073	0.0021	0.0015	0.00018
13.39	3.23E+07	0.0121	0.0016	0.0069	0.0019	0.0015	0.00018
17.82	4.26E+07	0.0116	0.0016	0.0066	0.0019	0.0014	0.00018
22.53	5.33E+07	0.0112	0.0015	0.0064	0.0018	0.0014	0.00018
27.30	6.38E+07	0.0109	0.0015	0.0062	0.0017	0.0014	0.00018
31.88	7.36E+07	0.0107	0.0015	0.0060	0.0017	0.0013	0.00018
36.04	8.21E+07	0.0105	0.0014	0.0059	0.0017	0.0013	0.00018
39.64	8.90E+07	0.0104	0.0014	0.0058	0.0016	0.0013	0.00018
42.57	9.42E+07	0.0103	0.0014	0.0058	0.0016	0.0013	0.00018
44.84	9.77E+07	0.0102	0.0014	0.0057	0.0016	0.0013	0.00018
46.50	9.98E+07	0.0102	0.0014	0.0057	0.0016	0.0013	0.00018
47.65	1.01E+08	0.0102	0.0014	0.0057	0.0016	0.0013	0.00018

press enter to continue :

SCREEN 6:

***** WATER FLIGHT RESULTS *****				
TIME (sec)	DEPTH (m)	VELOCITY (m/sec)	ACCELERATION (m/sq.(sec))	DRAG AREA (sq.m)
0.039	0.01	0.38	9.81	0.0139
0.156	0.13	1.53	9.80	0.0139
0.352	0.65	3.44	9.76	0.0139
0.625	2.01	6.09	9.64	0.0136
0.977	4.84	9.44	9.41	0.0127
1.406	9.90	13.39	9.04	0.0121
1.914	18.02	17.82	8.48	0.0116
2.500	30.09	22.53	7.72	0.0112
3.164	46.93	27.30	6.79	0.0109
3.906	69.22	31.88	5.73	0.0107
4.727	97.42	36.04	4.63	0.0105
5.625	131.76	39.64	3.57	0.0104
6.602	172.21	42.57	2.63	0.0103
7.656	218.57	44.84	1.84	0.0102
8.789	270.53	46.50	1.23	0.0102
10.000	327.70	47.65	0.78	0.0102

DO YOU WISH RUN ANOTHER CASE ? (Y OR N) : N

SCREEN 8:

TITLE : MK-1A, ANTIBES SITE
DATE : 12 JANUARY 1988
Program FREEFALL

1	Ogive Radius:	2030.0 mm
2	Body Diameter:	273.0 mm
3	Nose Length:	723.0 mm
4	Body Length:	2604.0 mm
5	Tail Cone Length:	464.0 mm
6	Stinger Length:	0.0 mm
7	Tail Base Diameter:	102.0 mm
8	Fin Area:1 fin,side:	0.0786 sq. m
9	Number of Fins:	4
10	Weight:	1476 kg
11	Buoyant Weight:	1308 kg
12	Roughness Factor:	1.20
13	Maximum Fall Time:	10.0 sec
14	Surface Water Temp:	20 deg C
15	Water Depth:	260 m

press ENTER to continue

SCREEN 9:

Nose Area : 4.1328E-01 sq. m
Body Area : 2.2333E+00 sq. m
Tail Area : 2.7792E-01 sq. m
Fin Area : 6.2880E-01 sq. m
Base Area : 8.1713E-03 sq. m
X-section : 5.8535E-02 sq. m
Total Area: 3.5533E+00 sq. m
(less base)

Frontal Area Drag Coef.: 0.1743
Impact Time : 8.5625 s
Impact Velocity : 46.2 m/s
Impact Drag Area : 0.0102 sq.m
Water Depth : 260 m
Surface Temperature : 20 deg C
Bottom Temperature : 2 deg C

press ENTER to continue

SCREEN 10:

ENTER THE GRAPH TITLE

MK-1A, ANTIBES SITE

SCREEN 11:

SCREEN 12

TYPICAL SEDIMENT PROPERTIES in kg/sq.cm

	Surface	Slope:	
	Qc	Qc/z	Fs/Qc
SNAP site (soft)	0.40	0.15	0.05
Antibes site (hard)	0.00	1.15	0.02

Qc(kg/sq.cm) = 0.1 Cu (kPa)

THE DEFAULT VALUES OF THE PENETRATION VARIABLES

1	Impact Velocity :	46.22 m/s
2	Fin Height above Body:	100.0 mm
3	Fin Thickness:	10.00 mm
4	Soil Density :	1600.0 kg/cu.m
5	Slope of Qc with Depth :	1.150 kg/sq.cm/m
6	Ratio of Fs/Qc:	0.020
7	Surface value of Qc :	0.000 kg/sq.cm
8	Maximum Penetration Time:	1.800 s

ENTER THE NUMBER OF THE VALUE YOU WISH TO CHANGE
IF NO CHANGES, ENTER " 0 "

>>> 8

ENTER THE CORRECTED VALUE: .8

SCREEN 13:

***** PENETRATION RESULTS *****

TIME (sec)	DEPTH (m)	VELOCITY (m/s)	ACCELERATION (g)
0.0001	0.00	46.22	0.11
0.0031	0.14	46.22	0.10
0.0125	0.58	46.22	-0.05
0.0281	1.30	46.17	-0.51
0.0500	2.31	45.97	-1.27
0.0781	3.59	45.42	-2.62
0.1125	5.13	43.87	-5.84
0.1531	6.84	41.09	-7.61
0.2000	8.66	37.16	-9.17
0.2531	10.48	31.91	-10.68
0.3125	12.15	25.21	-12.05
0.3781	13.49	17.02	-13.14
0.4500	14.32	7.44	-13.84
0.5281	14.47	0.00	0.00
0.6125	14.47	0.00	0.00
0.7031	14.47	0.00	0.00
0.8000	14.47	0.00	0.00
0.5147	14.47	<= Final Values at Arrest	

SCREEN 14:

DO YOU WISH TO RUN ANOTHER PENETRATION CASE ?
(Y or N): N

SCREEN 15:

ENTER THE GRAPH TITLE
for the time base curve

MK-1A, ANTIBES SITE

SCREEN 16:

SCREEN 17:

ENTER THE GRAPH TITLE
for the depth base curve

MK-1A, ANTIBES SITE

SCREEN 18:

SCREEN 19:

DO YOU WISH TO TERMINATE THIS SESSION ?
(Y or N): Y

APPENDIX B

A FREEFALL.PAS Program Listing


```

PROGRAM FREEFALL (input,output);

(* Include with "$I" prefix  typedef.sys *)
(* Include with "$I" prefix  graphix.sys *)
(* Include with "$I" prefix  kernel.sys  *)
(* Include with "$I" prefix  windows.sys *)
(* Include with "$I" prefix  findwrld.hgh *)
(* Include with "$I" prefix  axis.hgh    *)
(* Include with "$I" prefix  polygon.hgh *)

CONST
    maxnum = 96;                (* maximum no. of values calculated *)
    g = 9.807;                  (* acceleration of gravity, m/s*s  *)
    pi = 3.1416;

TYPE
    Number = array [0..maxnum] of real;
VAR
    i,N,flag1,flag2,flag3,setswitch,flagx,flagz,Switch,Branch : integer;
    xsexn,totarea,Cd,maxtime,roughness,R,D,L,L1,L2,L3,L4,B,F : real;
    surftemp,constant,distance,density,viscosity,stinger : real;
    depth,Rn,v,nose,body,tail,fins,base,drag,buoyantwt : real;
    weight,areaCd,impacttime,impactvel,impactCdA : real;
    finalt,finalz,finxsexn,Gravforce,soilQ,DragWater,soildensity: real;
    tmax,finheight,finthickness,Qzero,slope1,slope2,Resistance : real;
    amax,timeconst : real;
    Rnum,vel,S,a,z,t,CdA : Number;
    value,value1,value2,value3 : PlotArray;

PROCEDURE WELCOME;

    var
        ch: char;
        j : integer;

    begin
        textmode (1);
        textbackground (7);
        textcolor (1);
        clrscr;

        for j := 1 to 3 do writeln;
        writeln ('          WELCOME TO FREEFALL          ');
        writeln;
        writeln ('      A Seafloor Penetrator Program      ');
        writeln ('          To Calculate:          ');
        writeln;
        writeln ('          Reynolds Numbers          ');
        writeln ('          Drag Coefficients          ');
        writeln ('          Drag Areas          ');
        writeln ('          Freefall Velocities          ');
        writeln ('          Sediment Penetration          ');
        for j := 1 to 2 do writeln;
        textcolor (4);

```

```

writeln ('press "enter" to continue');
readln (ch);

clrscr;
textcolor (1);
for j := 1 to 8 do writeln;
writeln ('The next screen will prompt you for');
writeln ('      BASIC PENETRATOR DIMENSIONS      ');
writeln;
writeln;
writeln;
writeln (' * Please provide linear dimensions ');
writeln ('      in MILLIMETERS !                ');
writeln;
writeln (' * Give the "FIN AREA"                    ');
writeln ('      in SQUARE METERS for                ');
writeln ('      one surface of one fin only            ');
for j := 1 to 5 do writeln;
textcolor (4);
writeln ('press "enter" to continue');
readln (ch);

end;      (* WELCOME *)

```

PROCEDURE INDATA;

```

const
  pi = 3.1416;

var
  ch : char;

begin
  InitGraphic;
  SetBackgroundColor (0);
  SetForegroundColor (14);
  DrawBorder;
  DefineWorld (1,0,1000,1000,0);
  DefineWindow (1,3,5,XMaxGlb-3,YMaxGlb-5);
  DefineHeader (1, 'PENETRATOR DIMENSIONS');
  SetHeaderOn;
  SelectWorld (1);
  SelectWindow (1);
  DrawBorder;
  (* DRAW THE PENETRATOR SHAPES *)

  DrawLine (400,175,600,175);      (* Draw Body *)
  DrawLine (600,175,600,325);
  DrawLine (600,325,400,325);
  DrawLine (400,325,400,175);
  DrawLine (620,175,825,225);      (* Draw Tail *)
  DrawLine (825,225,825,275);

```



```

DrawLine (825,275,620,325);
DrawLine (620,325,620,175);

DrawLine (620,350,825,300);          (* Draw Fins *)
DrawLine (825,300,810,400);
DrawLine (810,400,680,400);
DrawLine (680,400,620,350);
DrawLine (620,150,825,200);
DrawLine (825,200,810,100);
DrawLine (810,100,680,100);
DrawLine (680,100,620,150);

DrawLine (380,175,250,200);          (* Draw Nose *)
DrawLine (250,200,180,225);
DrawLine (180,225,150,250);
DrawLine (150,250,180,275);
DrawLine (180,275,250,300);
DrawLine (250,300,380,325);
DrawLine (380,325,380,175);

DrawLine (840,225,970,225);          (* Draw Stinger *)
DrawLine (970,225,970,275);
DrawLine (970,275,840,275);
DrawLine (840,275,840,225);

SetLineStyle (4);                    (* Draw Dimension Lines *)
DrawLine (20,175,120,175);
DrawLine (20,325,120,325);
DrawLine (150,450,150,580);
DrawLine (390,450,390,580);
DrawLine (610,450,610,580);
DrawLine (825,450,825,580);
DrawLine (970,450,970,580);
DrawLine (940,225,940,65);
DrawLine (940,65,900,65);
DrawLine (940,275,940,350);
DrawLine (225,270,350,80);
DrawLine (350,80,370,80);

DefineWindow(2,3,5,XMaxGlb-3,YMaxGlb-5); (* Set Window for Text *)
SelectWindow (2);

Delay (500);

GotoXY(32,4);                        (* Read Ojive Radius *)
write('R : ');
read(R);

GotoXY (6,8);                        (* Read Body Diameter *)
write ('D : ');
read (D);

GotoXY (17,14);                      (* Read Nose Length *)
write ('L1 : ');
read (L1);

```

```

GotoXY (35,14);          (* Read Body Length *)
write ('L2 : ');
read (L2);

GotoXY (52,14);          (* Read Tail Cone Length *)
write ('L3 : ');
read (L3);

GotoXY (66,14);          (* Read Stinger Length *)
write ('L4 : ');
read (L4);

GotoXY (63,4);           (* Read Tail Base Diameter *)
write ('B : ');
read (B);

GotoXY (10,18);          (* Read Fin Area (one side only) *)
write ('Fin Area : ');
read (F);

GotoXY (10,19);          (* Read Number of Fins *)
write ('Number of Fins : ');
read (N);

GotoXY (10,20);          (* Read the Dry Weight *)
write ('Dry Weight (kg) : ');
read (weight);

GotoXY (10,21);          (* Read the Buoyant Weight *)
write ('Buoyant Weight (kg) : ');
read (buoyantwt);

GotoXY (10,22);          (*Read the Surface Roughness Factor *)
write ('Roughness Factor (1.0 smooth, 1.5 rough) : ');
read (roughness);

repeat until KeyPressed;
Delay (500);
Leavegraphic;

TextMode (3);
textbackground (7);
textcolor (1);
clrscr;

(* DEFAULT VALUES OF CONSTANTS *)
maxtime := 30;           (* Initial Value of Freefall time (s) *)
depth := 1000;           (* Initial Value of Water Depth (m) *)
surftemp := 20;          (* Initial Value of Surface Water Temp (C) *)
impacttime := 0;
impactvel := 0;
impactCdA := 0;
distance := 0;
flag1 := 22;
flagx := 1;

```

```

soildensity := 1600; (* soil density, kg/cu.m *)
slope1 := 0.150; (* slope of Qc vs. depth curve, kg/sq.cm/m *)
slope2 := 0.05; (* ratio of Fs/Qc *)
Qzero := 0.40; (* surface value of Qc *)
finheight := 100; (* height of fins above the body *)
finthickness := 10; (* thickness of fins *)
tmax := 1.8; (* maximum penetration time *)

end; (* INDATA *)

```

PROCEDURE CHANGE;

```

var
  j : integer;
  num : real;

Begin
  textbackground (7);
  textcolor (1);
  clrscr;

  while (flag1 <> 0) do
    begin
      TextColor (1);
      writeln;
      writeln ('      YOU HAVE ENTERED THE FOLLOWING DATA, ');
      writeln;
      writeln ('      1   Ogive Radius:      ',R:10:1,' mm   ');
      writeln ('      2   Body Diameter:     ',D:10:1,' mm   ');
      writeln ('      3   Nose Length:       ',L1:10:1,' mm   ');
      writeln ('      4   Body Length:       ',L2:10:1,' mm   ');
      writeln ('      5   Tail Cone Length:   ',L3:10:1,' mm   ');
      writeln ('      6   Stinger Length:     ',L4:10:1,' mm   ');
      writeln ('      7   Tail Base Diameter: ',B:10:1,' mm   ');
      writeln ('      8   Fin Area:1 fin,side:',F:10:4,' sq. m ');
      writeln ('      9   Number of Fins:     ',N:10,' ');
      writeln ('     10   Weight (dry):       ',weight:10:0,' kg');
      writeln ('     11   Buoyant Weight:     ',buoyantwt:10:0,' kg');
      writeln ('     12   Roughness Factor:   ',roughness:10:2,' ');
      writeln ('     13   Maximum Fall Time: ',maxtime:10:1,' sec ');
      writeln ('     14   Surface Water Temp: ',surftemp:10:0,' C');
      writeln ('     15   Water Depth:       ',depth:10:0,' m');

      for j := 1 to 2 do writeln;
      Textcolor (4);
      writeln ('      ENTER THE NUMBER OF THE VALUE YOU WISH TO CHANGE ');
      writeln ('      IF NO CHANGES,      ENTER " 0 " ');
      write ('      >>> ');
      readln (num);
      flag1 := trunc (num);
      if (flag1 <> 0) then write ('      ENTER THE NEW VALUE : ');
      CASE flag1 of
        1: readln (R);

```

```

2: readln (D);
3: readln (L1);
4: readln (L2);
5: readln (L3);
6: readln (L4);
7: readln (B);
8: readln (F);
9: readln (N);
10: readln (weight);
11: readln (buoyantwt);
12: readln (roughness);
13: readln (maxtime);
14: readln (surftemp);
15: readln (depth);
end;      (* CASE *)

clrscr;
end;      (* while *)

      (* Convert millimeter dimensions to meters *)

if L2 = 0 then L2 := (L1 + L3)/20;      (* prevents out of range values *)
R := R/1000;
D := D/1000;
L1 := L1/1000;
L2 := L2/1000;
L3 := L3/1000;
L4 := L4/1000;
B := B/1000;
flag2 := 0;
end;      (* CHANGE *)

```

PROCEDURE AREA;

```

const
  pi = 3.1416;

var
  ch: char;
  i : integer;
  theta,sintheta,costheta,surflength : real;

begin
      (* Calculate the area of the nose *)
  theta := arctan(L1/(R-D/2));
  nose := 2*pi*R*R*(sin(theta) - theta*cos(theta));

      (* Calculate the area of the body *)
  body := pi*D*L2;

      (* Calculate the area of the tail cone *)
  surflength := sqrt(sqr(D - B)/4 + L3*L3);
  tail := pi*surflength*(D + B)/2;

```

```

                                (* Calculate the area of the stinger *)
stinger := pi*B*L4;

                                (* Calculate the area of the fins *)
fins := F*N*2;

                                (* Calculate the area of the base *)
base := pi*(B*B)/4;

xsexn := pi*D*D/4;

totarea := nose + body + tail + fins;

clrscr;

end;                                (* INDATA *)


PROCEDURE SEAWATER;

var
    factor1,factor2,temp,focus : real;

begin
    Textmode (3);
    textbackground (7);
    textcolor (1);
    focus := sqr(surftemp - 2)/(2*2500);          (* focus of parabola *)
    if distance < 2500 then
        temp := surftemp - sqrt(2*focus*distance)  (* parabolic temp decay *)
    else temp := 2;
    factor1 := - (Exp(1.56*Ln(temp/779)));
    density := 1027.9*(Exp(factor1));

    factor2 := exp(-1.56863*Ln(1 + temp/47.15));  (*Find Kinematic Viscosity *)
    viscosity := 0.00187945*factor2/density;

end;                                (* SEAWATER *)


PROCEDURE DRAGAREA (var drag,Rn : real);

    (*****)
    (*Drag area is calculated as the sum of all component Cd*A *)
    (*)
    (*The individual drag coefficients are as follows: *)
    (*)

```

```

(*)  Nose : pressure drag      = 0.05                      *)
(*)      skin friction drag    = f(Rnum)                  *)
(*)                                                                    *)
(*)  Body : skin friction drag = f(Rnum) for 3 regimes:    *)
(*)              0 < R < 2E5      *)
(*)              2E5 < R < 1E7    *)
(*)              1E7 < R < 1E12   *)
(*)                                                                    *)
(*)  Tail Cone :
(*)      pressure drag        = f(L3/D)                  *)
(*)      skin friction drag    = f(Rnum) as for body      *)
(*)                                                                    *)
(*)  Stinger : skin friction drag = f(Rnum) as for body    *)
(*)                                                                    *)
(*)  Base: pressure drag      = f(diameter,body friction) *)
(*)                                                                    *)
(*)  Fins: skin friction drag = f(Rnum) as for body        *)
(*)  (*****)

```

```

var
  noseCd,tailCd,baseCd : real;
  beta,expon,denom : real;
  dragnose,dragbody,dragfins,dragtail,dragstinger,dragbase : real;

begin
  if flag3 = -1 then
    begin
      clrscr;
      writeln;
      writeln (' DRAG AREAS FOR COMPONENTS');
      writeln;
      writeln (' VELOCITY      R No.      TOTAL      NOSE      BODY      FINS      TAIL      BASE ');
      writeln (' (m/s)              (sq.m)   (sq.m)   (sq.m)   (sq.m)   (sq.m)   (sq.m) ');
      writeln;
      flag3 := 1;          (* set printing flag initial value *)
    end;

    if (Rn < 2.0E5) then
      Cd := 1.328/(sqrt(Rn));
    if (Rn >= 2.0E5) then
      if (Rn <= 1.0E7) then
        Cd := 0.003;
      if (Rn > 1.0E7) then
        begin
          expon := 1/6;
          denom := exp (expon*(ln(Rn)));
          Cd := 0.046/denom;
        end;

        (* Calculate the Drag Areas of the Components *)
        noseCd := exp(-0.748*L1/D - 3.17);      (* frontal area pressure drag *)
        tailCd := exp(-0.748*L3/D - 3.17);      (* tail cone pressure drag *)
        if L3 = 0 then baseCd := 0.14
        else
          (* base drag estimated from *)
          (* Nick Murray's curves in *)

```

```

        beta := arctan((D-B)/(2*L3))*180/pi;      (* Oceanic Engr. Jan.'85, *)
        baseCd := 0.004*beta - 0.02;              (* page 38 *)
        if L4 > 0 then
            baseCd := baseCd*(1 + L4*(0.14 - baseCd)/(4*B));
        if L4 > 4*B then baseCd := 0.14;
        end;

        dragnose := noseCd*xsexn + roughness*nose*Cd;
        dragbody := roughness*body*Cd;
        dragfins := roughness*fins*Cd;
        dragstinger := roughness*stinger*Cd;
        dragtail := tailCd*(xsexn-base) + roughness*tail*Cd + dragstinger;
        dragbase := baseCd*base;
        drag := dragnose + dragbody + dragfins + dragtail + dragbase;

        if setswitch = 1 then
            writeln (v:8:2,' ',Rn:8,drag:9:4,dragnose:9:4,dragbody:9:4,dragfins:9:4,dragtail:9:4,dragbase:9:4);
        end;      (* DRAGAREA *)

```

PROCEDURE IMPACT;

```

var
    deltaS,dS,Ratio : real;      (* Calculate Impact Values *)
begin
    if distance > depth then      (* of time, velocity, and *)
        if interpolation of data then
            begin
                deltaS := S[i]-S[i-1];
                dS := depth - S[i-1];
                Ratio := dS/deltaS;
                impacttime := t[i-1] + Ratio*(t[i] - t[i-1]);
                impactvel := vel[i-1] + Ratio*(vel[i] - vel[i-1]);
                impactCdA := CdA[i-1] + Ratio*(CdA[i] - CdA[i-1]);
                flag2 := 1;
            end;
        end;      (* IMPACT *)
    end;

```

PROCEDURE FLIGHT;

```

var
    deltaT : real;

begin
    constant := maxtime/(sqr(maxnum));      (* time constant, time := constant*sqr(maxnum) *)
    vel[0] := 0.0;      (* initialize velocity *)
    S[0] := 0;      (* initialize distance *)
    CdA[0] := 1;      (* initialize drag area *)
    t[0] := 0;

    for i := 1 to maxnum do

```

```

begin
    t[i] := constant*i*i;                                (* absolute time *)
    deltaT := t[i] - t[i-1];                             (* time delta *)
    a[i] := g - density*vel[i-1]*vel[i-1]*(CdA[i-1])/(2*buoyantwt);
    vel[i] := vel[i-1] + a[i]*deltaT;                    (* new velocity *)
    S[i] := S[i-1] + vel[i]*deltaT;                      (* new distance *)
    distance := S[i];
    v := vel[i];
    Rnum[i] := v*(L2)/viscosity;    (* calculate new Reynolds No. *)
    Rn := Rnum[i];
    DRAGAREA (drag,Rn);    (* calculate new drag area *)
    CdA[i] := drag;        (* Provide the new drag area *)

    if flag2 = 0 then IMPACT;

        (* Calculate new values of viscosity and water density *)
    SEAWATER;

        (* Set the printing switch in procedure DRAGAREA *)
    setswitch := 0;
    flag3 := flag3 + 1;
    if (flag3/6 = 1) then
        begin
            setswitch := 1;
            flag3 := 0;
        end;

    end;

    textcolor (red);
    writeln;
    write (' press enter to continue : ');
    repeat until Keypressed;
end;    (* FLIGHT *)

```

PROCEDURE PRINTOUT;

```

var
    j,flag4 : integer;
    yesno : char;

Begin
    clrscr;
    flag4 := 0;
    writeln;
    writeln ('          *****   WATER FLIGHT RESULTS   ***** ');
    writeln;
    writeln ('          TIME      DEPTH    VELOCITY    ACCELERATION    DRAG AREA  ');
    writeln ('          (sec)      (m)      (m/sec)    (m/sq.(sec))    (sq.m)    ');
    writeln;
    for i := 1 to maxnum do
        begin

```



```

        flag4 := flag4 + 1;
        if (flag4/6) = 1 then
            writeln (t[i]:10:3, S[i]:10:2, vel[i]:9:2, a[i]:12:2, CdA[i]:14:4);
        if flag4 = 6 then
            flag4 := 0;
        end;

textcolor (4);
writeln;
write (' DO YOU WISH RUN ANOTHER CASE ? (Y OR N) : ');
readln (yesno);
if yesno = 'Y' then flag1 := 33;
if yesno = 'y' then flag1 := 33;
if yesno = 'N' then flag1 := 0;
if yesno = 'n' then flag1 := 0;

if flag1 = 33 then
begin
    R := R*1000;
    D := D*1000;
    L1 := L1*1000;
    L2 := L2*1000;
    L3 := L3*1000;
    L4 := L4*1000;
    B := B*1000;
end; (* 'IF' *)
if flag1 = 0 then
begin
    Textmode (1);
    TextBackground (7);
    TextColor (4);
    clrscr;
    for j := 1 to 4 do writeln;
    write (' THE FOLLOWING IS THE CURRENT INPUT');
    TextColor (1);
    delay (1500);
    writeln;
    writeln;
    writeln (' PROVIDE THE TITLE AND DATE');
    writeln (' OF THIS RUN ');
    delay (2500);
    TextMode (3);
end;
clrscr;

end; (* PRINTOUT *)

```

PROCEDURE SUMMARY;

```

type
    dataname = string[40];

var

```

```

yesno : char;
j : integer;
title, date : dataname;

begin
  clrscr;
  TextBackground (7);
  TextColor (1);
  R := R*1000;
  D := D*1000;
  L1 := L1*1000;
  L2 := L2*1000;
  L3 := L3*1000;
  L4 := L4*1000;
  B := B*1000;
  areaCD:= CdA[maxnum]/xsexn;

  clrscr;
  writeln;
  write (' TITLE : ');
  readln (title);
  write (' DATE : ');
  readln (date);
  writeln (' Program FREEFALL');
  writeln;
  writeln ('      1   Ogive Radius:      ',R:10:1,' mm ');
  writeln ('      2   Body Diameter:      ',D:10:1,' mm ');
  writeln ('      3   Nose Length:        ',L1:10:1,' mm ');
  writeln ('      4   Body Length:        ',L2:10:1,' mm ');
  writeln ('      5   Tail Cone Length:    ',L3:10:1,' mm ');
  writeln ('      6   Stinger Length:      ',L4:10:1,' mm ');
  writeln ('      7   Tail Base Diameter:  ',B:10:1,' mm ');
  writeln ('      8   Fin Area:1 fin,side: ',F:10:4,' sq. m ');
  writeln ('      9   Number of Fins:      ',N:10,' ');
  writeln ('     10   Weight:              ',weight:10:0,' kg');
  writeln ('     11   Buoyant Weight:      ',buoyantwt:10:0,' kg ');
  writeln ('     12   Roughness Factor:    ',roughness:10:2,' ');
  writeln ('     13   Maximum Fall Time:   ',maxtime:10:1,' sec ');
  writeln ('     14   Surface Water Temp:  ',surftemp:10:0,' deg C');
  writeln ('     15   Water Depth:        ',depth:10:0,' m');
  writeln;
  write (' press ENTER to continue ');
  repeat until KeyPressed;
  clrscr;
  writeln;
  writeln;
  writeln ('      Nose Area : ',nose:10,' sq. m');
  writeln ('      Body Area : ',body:10,' sq. m');
  writeln ('      Tail Area : ',tail:10,' sq. m');
  writeln ('      Fin Area : ',fins:10,' sq. m');
  writeln ('      Base Area : ',base:10,' sq. m');
  writeln ('      X-section : ',xsexn:10,' sq. m');
  writeln ('      Total Area: ',totarea:10,' sq. m');
  writeln ('      (less base) ');
  writeln;

```

```

writeln ('Frontal Area Drag Coef.: ',areaCd:10:4);
writeln ('      Impact Time : ',impacttime:10:4,' s');
writeln ('      Impact Velocity : ',impactvel:7:1,' m/s');
writeln ('      Impact Drag Area : ',impactCdA:10:4,' sq.m');
writeln ('      Water Depth : ',depth:5:0,' m');
writeln ('      Surface Temperature : ',surftemp:5:0,' deg C');
writeln ('      Bottom Temperature :      2 deg C');
writeln;
write (' press ENTER to continue ');
repeat until KeyPressed;
end;          (* SUMMARY *)

```

PROCEDURE SETARRAY;

```

begin
  for i := 1 to (maxnum) do
    begin
      value[i,1] := S[i];
      value[i,2] := vel[i];
    end;
end;          (* SETARRAY *)

```

PROCEDURE DRAWGRAF;

```

var
  header : array[1..30] of char;
  j : integer;
  temp : real;

begin
  clrscr;
  for j := 1 to 3 do writeln;
  writeln ('      ENTER THE GRAPH TITLE ');
  writeln;
  write ('      ');
  readln (header);

  clrscr;
  InitGraphic;
  SetBackgroundColor(0);
  SetForegroundColor(14);

  DrawBorder;
  DefineWindow(1,trunc(XMaxGlb/10),trunc(YMaxGlb/10),trunc(XMaxGlb/1.1),trunc(YMaxGlb/1.1));

  Defineheader(1, header);
  SelectWindow(1);
  SetHeaderOn;

  FindWorld(1,value,maxnum,1.05,1.05);

```

```

with World[1] do
begin
    temp := Y1;
    Y1 := Y2;
    Y2 := temp;
end;

SelectWorld(1);
SelectWindow(1);
DrawBorder;

SetLineStyle(0);
DrawAxis(6,-6,0,0,0,0,0,false);
DrawPolygon(value,1,maxnum,-9,3,0);

GotoXY(2,12);                                (* labelling the axes *)
Write(' Vel ');
GotoXY(2,13);
Write('(m/s)');
GotoXY(35,24);
Write(' Depth (m) ');

GotoXY(40,18);
Write(' Max. Velocity = ',vel[maxnum]:4:1,' m/s');

GotoXY(40,19);
Write('Frontal Area Cd = ',areaCd:6:3);
Repeat until Keypressed;
LeaveGraphic;

TextMode (1);
clrscr;
if impactvel = 0 then writeln (' NO IMPACT, BURIAL CALCULATIONS ABORTED');
Repeat until KeyPressed;
Textmode (3);

end;      (* DRAWGRAF *)

PROCEDURE START;

var
    yesno : char;
    choice,j : integer;
    num : real;                                (* dummy input *)

begin
    TextMode (3);
    TextBackground (7);
    TextColor (1);
    clrscr;

    finalt := 0;                                (* default value of penetration time *)
    finalz := 0;                                (* default value of penetration depth *)

```

```

choice := 1;
amax := 0;

while (choice <> 0) do
begin
  writeln (' TYPICAL SEDIMENT PROPERTIES in kg/sq.cm');
  writeln;
  writeln ('          Surface      Slope: ');
  writeln ('          Qc          Qc/z      Fs/Qc ');
  writeln (' SNAP site (soft)  0.40      0.15      0.05 ');
  writeln (' Antibes site (hard) 0.00      1.15      0.02 ');
  writeln;
  writeln (' Qc(kg/sq.cm) = 0.1 Cu (kPa) ');
  writeln;
  writeln (' THE DEFAULT VALUES OF THE PENETRATION VARIABLES ');
  writeln;
  writeln (' 1 Impact Velocity :      ', impactvel:10:2, ' m/s');
  writeln (' 2 Fin Height above Body: ', finheight:10:1, ' mm');
  writeln (' 3 Fin Thickness:      ', finthickness:10:2, ' mm');
  writeln (' 4 Soil Density :      ', soildensity:10:1, ' kg/cu.m');
  writeln (' 5 Slope of Qc with Depth : ', slope1:10:3, ' kg/sq.cm/m');
  writeln (' 6 Ratio of Fs/Qc:      ', slope2:10:3);
  writeln (' 7 Surface value of Qc : ', Qzero:10:3, ' kg/sq.cm');
  writeln (' 8 Maximum Penetration Time: ', tmax:10:3, ' s');
  writeln;
  TextColor (4);
  writeln (' ENTER THE NUMBER OF THE VALUE YOU WISH TO CHANGE ');
  writeln (' IF NO CHANGES, ENTER " 0 " ');
  write ('          >>> ');
  readln (num);
  choice := trunc (num);

  if (choice <> 0) then write (' ENTER THE CORRECTED VALUE: ');

  CASE choice of
    1: readln (impactvel);
    2: readln (finheight);
    3: readln (finthickness);
    4: readln (soildensity);
    5: readln (slope1);
    6: readln (slope2);
    7: readln (Qzero);
    8: readln (tmax);
  end;
  (* CASE *)
  clrscr;
  Textcolor (1);
end;
(* while *)

clrscr;
TextColor (4);
for j := 1 to 4 do writeln;
writeln(' CALCULATING, PLEASE WAIT ');
Textcolor (1);

```

```

finxsexn := N*finheight*finthickness;
L1 := L1/1000;      (* convert millimeter dimensions to meters *)
L2 := L2/1000;
L3 := L3/1000;
L4 := L4/1000;
D := D/1000;
B := B/1000;
finxsexn := finxsexn/1.0E6;
L := L1+L2+L3+L4;
slopel := slopel*g*1.0E4;
Qzero := Qzero*g*1.0E4;

(* Setting initial Values *)
timeconst := tmax/(sqr(maxnum)); (* note: t[i] := timeconst*sqr(i) *)
z[0] := 0.0;
t[0] := 0;
a[0] := 0;
vel[0] := impactvel;
Branch := 1;
end;      (* START *)

PROCEDURE GRAVITYFORCE;
var
  SG,bw : real;
begin
  SG := soildensity/1000;
  bw := buoyantwt*g;      (* buoyant force in N *)
  if z[i] < L then
    Gravforce := bw*(1 - z[i]/L) + (bw/SG)*z[i]/L
  else
    Gravforce := bw/SG;
end;      (* GRAVITYFORCE *)

PROCEDURE WATERDRAG;
const
  rhowater = 1030;
begin
  if z[i] < L then
    DragWater := 0.5*(rhowater*vel[i]*vel[i]*impactCdA)*sqr(1 - z[i]/L)
  else
    DragWater := 0.;      (* Units of Newtons *)
end;      (* WATERDRAG *)

FUNCTION Qc : real;
var
  vell,SRF : real;
begin
  if vel[i] < 0.001 then
    vell := 0.001
  else

```

```

    vel1 := vel[i];
    SRF := 3.13*(exp(0.2*ln(vel1))-0.457); (* DELFT rate adjustment *)
    Qc := (slope1*z[i] + SRF + Qzero);
end; (* Qc *)

```

```

FUNCTION Fs : real;
begin
    Fs := Qc*slope2;
end; (* Fs *)

```

```

PROCEDURE SOILRESIST;
var
    Qshaft, Qnose, Qtail, Qfin, Qfinbear, Qstinger, Qbase, rho, suction: real;
    deltaQ, zbar, vsqr : real;
    areafactor, ratio : real; (* reduction factor for tail areas *)
begin
    Qshaft := 0;
    Qtail := 0;
    Qfin := 0;
    Qfinbear := 0;
    Qstinger := 0;
    Qbase := 0;

    rho := soildensity; (* abbreviate for convenience *)
    areafactor := (nose - pi*D*D/2)/nose; (* equivalent conical area *)
    if L3 < 2.5*D then
        suction := (xsexn-base)*(1-sqrt(L3/(2.5*D))) (* tail suction factor *)
    else suction := 0.0;
    vsqr := vel[i]*vel[i]; (* abbreviate for convenience *)

    if z[i] <= L1 then
        Qnose := (Qc*xsexn + Fs*nose*areafactor)*z[i]/L1;
    if z[i] > L1 then
        begin
            Qnose := Qc*xsexn + Fs*nose*areafactor;
            if z[i] <= (L1+L2) then
                Qshaft := Fs*pi*D*(z[i] - L1)
            else
                Qshaft := Fs*body;
            if z[i] > (L1+L2) then
                begin
                    zbar := L1+L2 + (z[i]-(L1+L2))/2;
                    if L3 > 0 then
                        ratio := (z[i]-L1-L2)/L3
                    else
                        ratio := 0;
                    deltaQ := Qc - Qzero;
                    Qtail := rho*(vsqr/2-g*zbar)*ratio*suction + Fs*tail*ratio;
                    Qfin := Fs*fins*ratio;
                    Qfinbear := (Qzero + deltaQ*((z[i]-(L1+L2))/z[i]))*finxsexn;
                end;
            end;
        end;
end;

```

```

if z[i] > (L1+L2+L3) then
begin
  zbar := L1 + L2 + L3/2;
  if L4 > 0 then
    Qstinger := Fs*stinger*(z[i]-(L1+L2+L3))/L4
  else Qstinger := 0;
  Qtail := (rho*vsqr/2 - rho*g*zbar)*suction + Fs*tail;
  Qfin := Fs*fins;
  if z[i] < (L1+L2+4*L3) then
    Qtail := Qtail*(z[i]-(L1+L2+L3))/(3*L3);
end;
if z[i] > L then
begin
  Qstinger := Fs*stinger;
  Qbase := (rho*vsqr/2 - rho*g*(z[i]-L))*base;
  if z[i] < (L+3*B) then
    Qbase := Qbase*(z[i]-L)/(3*B);
end;

Resistance := Qnose + Qshaft + Qtail + Qfin + Qfinbear + Qstinger + Qbase;
if vel[i] <= 0 then
  Resistance := Gravforce;
end;
(* SOILRESIST *)

```

```

FUNCTION FORCE : real;
begin
  FORCE := Gravforce - DragWater - Resistance;
end;
(* FORCE *)

```

```

PROCEDURE DECELERATION;
var
  deltaT, vell : real;
begin
  deltaT := t[i] - t[i-1];
  a[i] := FORCE/weight; (* acceleration in m/s*s *)
  vell := vel[i] + a[i]*deltaT;
  if a[i] < amax then
    amax := a[i];
  if vell < 0 then
    vel[i] := 0
  else
    vel[i] := vell;
  z[i] := z[i] + vel[i]*deltaT;
  if Branch = 1 then
    begin
      if vell <= 0 then
        begin
          finalt := t[i];
          finalz := z[i];
          Branch := 3;
        end;
    end;

```



```

        end;
end;                                (* DECELERATION *)

PROCEDURE TABLE;
var
    flag6,j : integer;
    gees : real;
    yesno : char;

begin
    clrscr;
    flag6 := 0;
    writeln;
    writeln ('          *****  PENETRATION  RESULTS  ***** ');
    writeln;
    writeln ('          TIME          DEPTH          VELOCITY  ACCELERATION  ');
    writeln ('          (sec)          (m)          (m/s)          (g)          ');
    writeln;
    gees := a[1]/g;
    writeln (t[1]:12:4,z[1]:12:2,vel[1]:12:2,gees:14:2);
    for i := 1 to maxnum do
        begin
            flag6 := flag6 + 1;
            if (flag6/1) = 6 then
                begin
                    gees := a[i]/g;
                    writeln(t[i]:12:4,z[i]:12:2,vel[i]:12:2,gees:14:2);
                    flag6 := 0;
                end;
            end;
        writeln;
        textcolor (4);
        writeln (finalt:12:4,finalz:12:2,'      <= Final Values at Arrest');
        Repeat Until KeyPressed;

        clrscr;
        textcolor (1);
        for j := 1 to 3 do writeln;
        writeln ('          DO YOU WISH TO RUN ANOTHER PENETRATION CASE ? ');
        write ('          ( Y or N ): ');
        readln (yesno);
        if yesno = 'Y' then Switch := 1;
        if yesno = 'y' then Switch := 1;
        if yesno = 'N' then Switch := 0;
        if yesno = 'n' then Switch := 0;
        L1 := L1*1000;          (* convert millimeter dimensions to meters *)
        L2 := L2*1000;
        L3 := L3*1000;
        L4 := L4*1000;
        D := D*1000;
        B := B*1000;
        finxsexn := finxsexn*1.0E6;
        slopel := slopel/(g*1.0E4);
        Qzero := Qzero/(g*1.0E4);

```

```
end; (* TABLE *)
```

PROCEDURE SETARRAY2;

```
begin
  for i := 1 to (maxnum) do
    begin
      if Switch = 2 then
        begin
          value1[i,1] := z[i];
          value1[i,2] := a[i];
          value2[i,1] := z[i];
          value2[i,2] := vel[i];
        end;
      if Switch = 1 then
        begin
          a[i] := -a[i]/g; (* accel => decel in g's *)
          value1[i,1] := t[i];
          value1[i,2] := a[i];
          value2[i,1] := t[i];
          value2[i,2] := vel[i];
          value3[i,1] := t[i];
          value3[i,2] := z[i];
        end;
      end;
    end;
end; (* SETARRAY2 *)
```

PROCEDURE DRAWGRAF2;

```
var
  header : array[1..50] of char;
  j : integer;
  temp : real;

begin
  clrscr;
  for j := 1 to 3 do writeln;
  writeln ('      ENTER THE GRAPH TITLE ');
  if Switch = 1 then
    writeln ('      for the time base curve ')
  else
    writeln ('      for the depth base curve ');
  writeln;
  write ('      ');
  readln (header);
  clrscr;
  InitGraphic;
  SetBackgroundColor(0);
  SetForegroundColor(14);

  DrawBorder;
  DefineWindow(1, trunc(XMaxGlb/9), trunc(YMaxGlb/10), trunc(XMaxGlb/1.07), trunc(YMaxGlb/1.1));
```

```

Defineheader(1, header);
SelectWindow(1);
SetHeaderOn;

FindWorld(1,value2,maxnum,1.05,1.05);
with World[1] do
  begin
    temp := Y1;
    Y1 := Y2;
    Y2 := temp;
  end;

SelectWorld(1);
SelectWindow(1);
DrawBorder;

SetLineStyle(0);
DrawAxis(6,-6,0,0,0,0,0,0,true);
DrawPolygon(value1,1,maxnum,-9,5,0);
AxisGlb := TRUE;
DrawPolygon(value2,1,maxnum,-1,1,0);
if Switch = 1 then
  begin
    AxisGlb := TRUE;
    DrawPolygon(value3,1,maxnum,-0,1,0);
  end;
GotoXY(2,6); (* labelling the axes *)
Write(' a ');
GotoXY(2,7);
Write(' (g) ');
GotoXY(2,8);
Write('.....');
GotoXY(2,11);
Write(' V ');
GotoXY(2,12);
Write(' (m/s) ');
GotoXY(2,13);
Write('+++++');
if Switch = 1 then
  begin
    GotoXY(2,16);
    Write(' Z ');
    GotoXY(2,17);
    Write(' (m) ');
    GotoXY(2,18);
    Write('-----');
    GotoXY(35,24);
    Write(' TIME (s) ');
  end;
if Switch = 2 then
  begin
    GotoXY(35,24);
    Write(' DEPTH (m) ');
  end;

```

```

        GotoXY(16,15);
        Write('Burial Depth: ',finalz:4:1,' m');
        GotoXY(16,16);
        amax := -amax/g;
        Write('Max. Decel.: ',amax:4:1,' g');
    end;
    Repeat until Keypressed;
    LeaveGraphic;
end;      (* DRAWGRAF2 *)

```

PROCEDURE RETURN;

```

var
    yesno : char;

begin
    clrscr;
    for i := 1 to 3 do writeln;
    writeln ('      DO YOU WISH TO TERMINATE THIS SESSION ? ');
    write ('          ( Y or N ): ');
    readln (yesno);
    if yesno = 'Y' then flagz := 0;
    if yesno = 'y' then flagz := 0;
    if yesno = 'N' then flagz := 1;
    if yesno = 'n' then flagz := 1;
    impacttime := 0;
    impactvel := 0;
    impactCdA := 0;
    distance := 0;
    flag1 := 22;
    flag2 := 0;
    flagx := 1;
end;      (* RETURN *)

```

```

Begin      (* MAIN *)

    flagz :=1;
    WELCOME;
    INDATA;
    while (flagz <> 0) do
        begin
            repeat
                flag3 := -1;
                CHANGE;
                AREA;
                SEAWATER;
                FLIGHT;
                PRINTOUT;
            until flag1 = 0;
            SUMMARY;
            SETARRAY;
            DRAWGRAF;
            if impactvel <> 0 then

```

```

begin
Switch := 1;
while (Switch <> 0) do
begin
START;
for i := 1 to maxnum do
begin
t[i] := timeconst*i*i;
a[i] := a[i-1];
vel[i] := vel[i-1];
z[i] := z[i-1];
GRAVITYFORCE;
WATERDRAG;
SOILLRESIST;
DECELERATION;
end;
TABLE
end;      (* while *)
Switch := 1;
SETARRAY2;
DRAWGRAF2;
LeaveGraphic;
Switch := 2;
SETARRAY2;
DRAWGRAF2;
end;      (* if *)
RETURN;
end;      (* while *)
End.

```

DISTRIBUTION

US Department of Energy
Office of Civilian Radioactive Waste Management
Attn: N. Eisenburg, RW-24
Washington, DC 20545

US Department of Energy
Albuquerque Operations Office
Attn: D. G. Jackson
P. O. Box 5400
Albuquerque, NM 87115

Building Research Establishment
Attn: Dr. Timothy Freeman
Garston
Watford DW2 FJR
Hertz, UNITED KINGDOM

Commission of the European Communities
Joint Research Centre
Attn: Dr. Nicholas Murray
I-21020 Ispra (Varese)
ITALY

Commissariat a L'Energie
Attn: Dr. Jean-Yves Boisson
ISPN DPT SEPD - CEN/FAR
B.P. No. 6
92265 Fontenay aux Roses
FRANCE

Principia Recherche Developpment

Attn: Dr. Philippe Bardey

Sophia Antipolis

06560 Valbonne

FRANCE

0400	C. H. Karnes
1551	C. E. Hailey
5146	G. C. Hauser
5165	W. J. Paterson
6300	R. W. Lynch
6330	W. D. Weart
6334	D. R. Anderson (5)
6334	R. D. Klett
9120	M. M. Newsom
9121	L. T. James
9122	R. H. Braasch
9122	M. J. Forrestal
9122	C. W. Young
9122	J. Hickerson (3)
9127	C. W. Childers
3151	W. L. Garner (3)
3154-1	C. H. Dalin (8)
	(for DOE/TIC Unlimited Release)
3141	S. A. Landenberger (5)
8524	P. W. Dean