SCHOOL OF
CIVIL ENGINEERING
INDIANA
DEPARTMENT OF HIGHWAYS

Interim Report

COMPUTATIONAL PACKAGE FOR PREDICTING
PILE STRESS AND CAPACITY

FHWA/IN/JHRP-84/16

Francisco Jose Tejidor

PURDUE UNIVERSITY

Interim Report

COMPUTATIONAL PACKAGE FOR PREDICTING
PILE STRESS AND CAPACITY

FHWA/IN/JHRP-84/16

Francisco Jose Tejidor

Digitized by the Internet Archive
in 2011 with funding from
LYRASIS members and Sloan Foundation; Indiana Department of Transportation

INTERIM REPORT
COMPUTATIONAL PACKAGE
FOR PREDICTING PILE STRESS
AND CAPACITY

To: H. L. Michael, Director
Joint Highway Research Project

August 23, 1984
Revised: February 1985
Project: C-36-36P

File: 6-14-16

From: C. W. Lovell, Research Engineer
Joint Highway Research Project

Attached is an Interim Report on the study, "Computational Package for Predicting Pile Stress and Capacity". The report is written by Francisco José Tejidor of our staff, who worked under my supervision.

The package PILE promises to be a highly useful one in the design and installation of highway bridge piles. It incorporates three elements: static analysis; pile driving equations; and wave equation analysis (WEAP). It is operational on the IDOH IBM computer.

An interim report is written to call the attention of interested parties to the existence of the computational package (PILE). The final project report will follow shortly, and will contain a number of elements. Most important of these will be: (a) additional comparisons of IDOH case studies with PILE predictions; (b) debugging of the program as it is used by IDOH; and (c) some modifications to make the program more useful and efficient in the routine IDOH operation.

Since the interim report is voluminous, it is recommended that only a limited number of copies be produced in full length. Chapters 3 through 6 are detailed summations of background material, of interest to only a few. Deletion of this portion of the report (for routine distribution) will reduce its length by about 150 pages.

Presentation of PILE listing, input form, and examples requires about 100 pages. It is presumed that any serious reader would wish to have all of the program in full detail.

The report is submitted for review, comment and acceptance in partial fulfillment of the referenced study.

Respectfully submitted,

CW Lovell

C. W. Lovell
Research Engineer

CWL/tb

cc:	A.G. Altschaeffl	G.K. Hallock	R.M. Shanteau
	J.M. Bell	J.F. McLaughlin	K.C. Sinha
	W.F. Chen	R.D. Miles	J.R. Skinner
	W.L. Dolch	P.L. Owens	C.A. Venable
	R.L. Eskew	B.K. Partridge	L.E. Wood
	J.D. Fricker	G.T. Satterly	S.R. Yoder
	W.H. Goetz	C.F. Scholer	

1. Report No. FHWA/IN/JHRP-84/16	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Computational Package for Predicting Pile Stress and Capacity		5. Report Date August 23, 1984 Revised: February 1985	
		6. Performing Organization Code	
7. Author(s) Francisco José Tejidor		8. Performing Organization Report No. FHWA/IN/JHRP 84/16	
9. Performing Organization Name and Address Joint Highway Research Project Civil Engineering Building Purdue University West Lafayette, Indiana 47907		10. Work Unit No.	
		11. Contract or Grant No.	
12. Sponsoring Agency Name and Address Indiana Department of Highways State Office Building 100 North Senate Avenue Indianapolis, Indiana 46204		13. Type of Report and Period Covered Interim Report	
		14. Sponsoring Agency Code	
15. Supplementary Notes Prepared in cooperation with the U.S. Department of Transportation, Federal Highway Administration. Part of the study titled "Computational Package for Predicting Pile Stress and Capacity."			
16. Abstract A comprehensive and versatile computer program for estimating driving stresses in piles, and load capacity of installed piles has been developed and named PILE. The program incorporates: (a) static analysis; (b) dynamic formulas; and (c) wave equation (WEAP) analysis. It is intended to provide the predictive capabilities needed by highway agencies in the design and construction of bridge piles. The report contains an intensive review of the state-of-the-art, as well as all the details of PILE, including example solutions.			
17. Key Words pile, computer, static capacity, driving formulas, wave equation, bridge foundations.		18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 180	22. Price

ACKNOWLEDGMENTS

The author would like to thank his major professor, Dr. C. W. Lovell, for the opportunity to participate in such an interesting and rewarding research project and for his invaluable guidance during the course of the research and in the preparation of this thesis.

A personal thanks is warmly extended to Prof. J. L. Chameau for his friendship and the inspiration his encouragements provided.

Much valuable guidance was derived from the disciplined approach toward the solution of computer algorithms as provided by Prof. V. J. Meyers in his Civil Engineering courses.

The author would also like to gratefully acknowledge the invaluable assistance of Mr. Steve Hull of the Indiana Department of Highways in the implementation of the computer program.

The financial support for this research was provided by the Indiana Department of Highways and the Federal Highway Administration. The research was administered through The Joint Highway Research Project, Purdue University, West Lafayette, Indiana.

INTERIM REPORT

COMPUTATIONAL PACKAGE
FOR PREDICTING PILE STRESS
AND CAPACITY

by

Francisco José Tejidor

Graduate Instructor in Research

Joint Highway Research Project

Project No: C-36-36P

File No: 6-14-16

Prepared for an Investigation

Conducted by the

Joint Highway Research Project
Engineering Experiment Station
Purdue University

in cooperation with the

Indiana Department of Highways

and the

U.S. Department of Transportation
Federal Highway Administration

The opinions, findings and conclusions expressed in this publication are those of the author and not necessarily those of the Federal Highway Administration.

Purdue University
West Lafayette, Indiana
August 23, 1984
Revised: February 1985

TABLE OF CONTENTS

	Page
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
LIST OF SYMBOLS AND NOMENCLATURE.....	ix
HIGHLIGHT SUMMARY.....	xvii
INTRODUCTION.....	1
CHAPTER	
1 - PILE-SOIL INTERACTION.....	3
Ultimate Load.....	4
Load Transfer.....	7
Residual Stresses.....	19
Installation Effects.....	25
Cohesionless Soils.....	25
Cohesive Soils.....	27
Negative Shaft Friction.....	30
2 - PILE-SOIL DESIGN CAPACITY.....	33
3 - PILE CAPACITY IN COHESIONLESS SOILS.....	38
Point Capacity.....	39
Meyerhof Model.....	39
Vesic Model.....	46
Shaft Capacity.....	50
Meyerhof Model.....	51
Vesic Model.....	55
Limit Unit Capacity.....	57
Reduction Factors.....	64
Recent Correlations.....	67
Group Capacity.....	76
Settlement of Piles.....	79
Meyerhof Method.....	79
Vesic Method.....	81

Project B-14321.....	209
Project B-12309.....	211
8 - SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS.....	213
Summary.....	213
General Criteria.....	213
Load Transfer.....	214
Static Analysis.....	215
Cohesionless Soils.....	216
Cohesive Soils.....	217
Dynamic Analysis.....	219
Conclusions.....	221
Recommendations for Further Study.....	222
LIST OF REFERENCES.....	225
APPENDICES	
Appendix A: PILE Program.....	233
Appendix B: PILE Input Form.....	317
Appendix C: Examples.....	320

LIST OF TABLES

Table	Page
1. Typical Values of Coefficient C_p	85
2. Adhesion Factor for Driven Piles	93
3. Variation of Alpha (α) with Soil Strength	97
4. Correction Factor F_L	98
5. Partial Safety Factors for Resistance to Axial Load	115
6. Probability of Failure under Normal Design Conditions	116
7. Summary of Pile Driving Formulas	126
8. Values of C_1 for Hiley Formula	127
9. Values of C_2 and C_3 for Hiley Formula	128
10. Values of Coefficient of Restitution (n)	129
11. Values of Hammer Efficiency	130
12. Recommended Values of Quake (Q)	181
13. Recommended Values of Damping Factors (J)	182
14. Properties of Cushion Materials	184
15. Properties of Pile Materials	185

LIST OF FIGURES

Figure	Page
1. Load Transfer From a Single Pile.....	8
2. Typical Simple Distributions of Load Transfer.....	10
3. Load Transfer From a Steel Pile Driven Through Compressible Silt to Rock.....	15
4. Load Transfer Analysis.....	17
5. Effects of Residual Loads on Load Distribution in Driven Piles in Sand at Arkansas River.....	21
6. Residual Load Distribution Effects.....	23
7. Bearing Capacity Factors and Critical Depth Ratios for Driven Piles.....	42
8. Relation Between Ultimate Point Resistance of Pile and Depth in Sand Stratum Beneath Weak Layer.....	44
9. Relation Between Ultimate Point Resistance of Pile and Depth in Thin Sand Layer Overlying Weak Soil.....	45
10. Assumed Failure Pattern Under Pile Point.....	48
11. Coefficient of Earth Pressure on Shaft of Piles Above Critical Depth in Sand.....	54
12. Approximate Relation Between Limiting Static Cone Resistance and Friction Angle of Sand.....	59
13. Ultimate Unit Point Resistance Versus D/B.....	69
14. Bearing Capacity Factor Versus D/B.....	71
15. Ultimate Unit Side Resistance Versus D/B.....	72
16. Lateral Earth Pressure Coefficient Versus D/B.....	73
17. Adhesion Factors for Case I.....	94

18.	Adhesion Factors for Case III.....	94
19.	Frictional Capacity Coefficient Versus Pile Penetration.....	106
20.	Particle and Wave Velocities During Driving.....	146
21.	Force and Particle Velocities at the Same Impedance of the Pile and the Hammer.....	149
22.	Effect of Skin Friction on the Force-Time Relationship at the Top of a Pile.....	154
23.	Idealization of Smith Model.....	161
24.	Soil Resistance Model.....	162
25.	Stress-Strain Behavior of Cushion Materials.....	164
26.	Plastic Deformation of the Soil.....	176
27.	Comparison Between Calculated Values from PILE and Measured Ultimate Load Capacity from Load Tests.....	205

LIST OF SYMBOLS AND NOMENCLATURE

A	= cross-sectional area
A'	= area of pile group enclosed in perimeter
A_p	= area of pile point
A_s	= pile surface area
a	= acceleration
B	= width of pile
B	= intercept of regression line
B'	= width of pile group
C	= coefficient of elastic deformation
C_p	= empirical coefficient dependent on soil type and construction
C_s	= empirical coefficient dependent on soil type and construction
$C_{(m,t)}$	= compression of internal spring
c	= cohesion
c	= stress wave velocity
c_a	= adhesion
c_h	= coefficient of radial consolidation of soil mass
c_m	= average undrained shear strength
c_u	= undrained shear strength
D	= depth of penetration
D_b	= depth of bearing stratum

D_c	= critical depth
D_r	= relative density
$D_{(m,t)}$	= element displacement
$D'_{(m,t)}$	= plastic displacement of external spring
d	= pile diameter
E_h	= pile driving hammer efficiency
E_p	= elastic modulus of pile
E_r	= manufacturer's hammer energy rating
E_s	= modulus of deformation of soil
$E^*_{s^s}$	= plane strain modulus of deformation
E_1	= energy reaching pile
E_2	= energy after impact
e	= coefficient of restitution
e	= relative displacement
e_f	= efficiency of hammer drop
e_{iv}	= efficiency factor for impact
F	= force
F_a	= friction force
F_C	= correction factor for soil strength
F_D	= correction factor for unit point capacity
F_{dyn}	= dynamic resistance
F_i	= initial wave force
F_L	= correction factor for pile penetration
F_r	= reflected wave force
F_s	= wave force on pile point
F_{SD}	= correction factor for unit shaft capacity
F_{stat}	= static resistance

$F_{(m,t)}$	= force in internal spring
FS	= factor of safety
f_l	= limit value of unit skin friction
$f_{\max}$	= peak soil-pile friction
$f_o(z)$	= local friction force
f_s	= unit shaft capacity
f_{s_c}	= calculated unit shaft capacity
f_{s_m}	= measured unit shaft capacity
G	= shear modulus
g	= gravitational acceleration
H	= thickness of weak deposit below pile point
H	= drop height of hammer
H_f	= thickness of fill penetrated by a pile group
H_s	= thickness of weak deposit penetrated by a pile group
I	= influence factor
I_{pp}	= dimensionless influence factor
I_{ps}	= dimensionless influence factor
I_r	= rigidity index
I_{rr}	= reduced rigidity index
J	= damping factor
$J_{(m)}$	= soil damping constant
K_o	= earth pressure coefficient at rest
K_s	= coefficient of lateral earth pressure
$K_{(m)}$	= spring constant for internal spring
$K'_{(m)}$	= spring constant for external spring
k	= axial stiffness of pile cushion
k_u	= empirical constant for Janbu formula
L	= length of pile shaft

l	= number of soil layers
M	= slope of regression line
m	= coefficient for soil type
m	= mass
m	= element number
N	= standard penetration resistance
N'	= corrected standard penetration resistance
N_c, N_q, N_γ	= bearing capacity factors
N'_s	= skin bearing capacity factor
N_σ	= bearing capacity factor
n	= soil density index
n	= number of piles in group
OCR	= overconsolidation ratio of clay
P	= shaft perimeter area
P	= best estimate of pile capacity
P_c	= calculated pile capacity from a dynamic formula
P_{n_f}	= additional pile load due to negative friction
p	= net foundation pressure
p_1	= pile perimeter
p_2	= perimeter of pile group
P_{su}	= ultimate shaft resistance
P_u	= ultimate pressure from pressuremeter test
Q_g	= ultimate group load
Q_{n_f}	= additional group load due to negative friction
Q_p	= ultimate point capacity
Q_{p_c}	= calculated point capacity
Q_{p_m}	= measured point capacity
Q_s	= ultimate shaft capacity

Q_u	= ultimate soil capacity
$Q(z)$	= local load transfer
$Q_{(m)}$	= element quake
q'	= normal shear stress
q_c	= limit static cone resistance
q_l	= limit value of unit point capacity
q_o	= limit point capacity for upper weak stratum
q_p	= unit bearing capacity
R	= total soil resistance
R_b	= reduction factor for point capacity in sands
R_d	= dynamic resistance of soil
R_s	= static resistance of soil
R_c	= reduction factor for point capacity in clays
R_u	= ultimate soil resistance
R_{u_f}	= final load capacity due to set-up
$R_{(m,t)}$	= external spring force
r	= center-to-center spacing between piles
S	= soil penetration
$S.D.$	= standard deviation
s	= total settlement of shallow foundation
sk_i	= set-up factor
T_h	= time factor
t	= time elapsed since pile driving
u	= hammer velocity after impact
u_p	= pile velocity after impact
u_p	= particle displacement
u^*	= relative pile-soil movement
v	= velocity of loading
$v_{(r)}$	= initial velocity of ram
$v_{(m,t)}$	= velocity of element

v	= hammer velocity before impact
v_h	= particle velocity in hammer
v_o	= initial hammer velocity
v_p	= particle velocity in pile
v_p	= pile velocity after impact
W	= weight of hammer
W_h	= weight of hammer
W_p	= weight of pile
W_{ram}	= ram weight
$W_{(m)}$	= weight of element
w	= settlement of a single pile
w'	= settlement of pile group
w_i	= displacement of pile element
w_o	= vertical displacement of pile top
w_{pp}	= settlement of pile point by load transmitted at point
w_{ps}	= settlement of pile point by load transmitted along shaft
w_s	= settlement due to axial deformation of pile shaft
w_z	= vertical displacement of pile shaft
z	= local depth
z	= impedance
z_h	= impedance of hammer
z_p	= impedance of pile
α	= adhesion factor
α_s	= coefficient of skin friction distribution

β	= skin friction factor
γ	= unit weight of soil
γ'	= effective unit weight of soil
γ_r	= partial safety factor for resistance
γ_s	= partial safety factor for load
Δ	= volume change parameter
ΔF	= net force
ΔL	= element length
ΔL	= differential length
ΔQ	= differential shaft load
ΔR_i	= immediate soil resistance
ΔS_{pp}	= plastic deformation of pile
ΔS_{ep}	= elastic deformation of pile
Δt	= time interval
δ	= angle of skin friction
η_g	= efficiency of pile group
ζ_g	= group settlement factor
λ	= stress-transfer factor
λ	= dimensionless coefficient
ν_s	= Poisson's ratio of soil
π_3	= pile-soil stiffness
ρ	= dimensionless factor
ρ	= mass density of pile
σ	= stress
σ_h	= horizontal stress in soil along shaft
σ'_m	= mean effective vertical stress
σ_v	= vertical stress in soil
σ'_v	= effective vertical stress

σ_{vp}	= vertical stress in soil at depth of pile point
σ'_{vp}	= effective vertical stress at pile point
τ_a	= pile-soil shear strength
ξ	= factor for elastic compression of soil and driving cushion
ϕ	= angle of internal friction
ϕ'	= effective angle of internal friction
$\frac{\partial}{\partial x}, \frac{\partial}{\partial t}$	= first partial derivative with respect to x, t
$\frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial t^2}$	= second partial derivative with respect to x, t

HIGHLIGHT SUMMARY

A computational package PILE has been developed in order to predict the ultimate bearing capacity of a pile and the dynamic stresses generated during installation. The ultimate bearing capacity of a pile which determines its static behavior is derived from analytical models which are dependent on subsurface conditions, pile type, method of installation, and other factors. The stresses generated during installation which will characterize the dynamic performance of a pile in-situ may be evaluated using a wave equation analysis. The main objective of the study is to afford the engineer a comprehensive means of effectively designing a pile foundation.

The ultimate bearing capacity of a pile in cohesionless soils is governed by the effective stress state of the surrounding soil. The capacity of a pile in cohesive soils is governed by the rate of pore pressure generation which affects the shear strength of the adjacent soil. If the rate of load application is such that pore pressure dissipation is rapid, then a drained-effective stress analysis may

be appropriate, otherwise an undrained-total stress analysis is conventionally employed. Since pile foundations are normally constructed as groups, the settlement and efficiency of a pile group also become pertinent variables.

The difficulties incumbent in the installation of piles may be alternately treated. The dynamic formulas have been found to be fundamentally flawed and consequent predictability can only be statistically improved on a regional basis. A wave equation analysis of pile driving, on the other hand, affords a more realistic assessment of ultimate load - final set relationship, driveability of a pile as a function of the equipment employed, and stresses generated during installation.

Two major conclusions may be drawn from initial use of the computational package. First, estimates of ultimate load bearing capacity are determined with reasonable accuracy. Second, depending on the variation of parameters, a wave equation analysis of pile driving will render, if only qualitatively, a correct assessment of difficulties concomitant with various installation procedures.

INTRODUCTION

Pile foundations are often used to transmit the required structural loads to a firm layer by penetrating the upper weak strata. The problem of pile foundations is then twofold. First, the designer must determine the depth at which the pile will have developed sufficient bearing capacity to sustain the design loads. The bearing capacity is developed both at the point and along the shaft of the pile. Second, the designer must choose an efficient and economical means of installation.

The foremost concern is the determination of ultimate bearing capacity. The present study will review recent advances in the field. The breadth of the study is determined by the intended generality of its final application. The recommended procedures must be sufficiently general so as to be applicable to most commonly encountered soil conditions, pile types, and construction practices.

Construction practices exert an important role in the determination of the final product. Conventional methods of corroborating design requirements in the field will be discussed. Furthermore, a detailed synthesis of improved analytical techniques will be presented. The advances afford a more realistic and therefore a more efficient and

economical assessment of common installation procedures.

In order to accomplish the task of a pile foundation design, a computer program will be devised which will incorporate in a comprehensive manner the analytical models necessary to investigate the static and dynamic behavior of a pile. The major objective in the formulation of the computer program will be to afford the designer the opportunity to vary parameters with relative ease, and to choose those models which have been extensively corroborated with field experience.

Finally, in order to instill confidence in the applicability of the computational package, it was necessary to test the program against a series of published pile load test results and case histories.

The computational package should therefore afford the designer a comprehensive and rational means of evaluating the static and dynamic performance of a pile.

The research was undertaken under the sponsorship of the Indiana Department of Highways (IDOH) and the Federal Highway Administration, and was coordinated through the Joint Highway Research Project, School of Civil Engineering, Purdue University. It is primarily intended to aid the Bridge Design Engineer and the Construction Engineer of the IDOH in the selection and placement of efficient and economic bridge foundations in the State of Indiana.

CHAPTER 1

PILE-SOIL INTERACTION

There are some analogous similarities between shallow foundations and deep foundations. In both cases, the foundation must possess sufficient safety against bearing and/or structural failure and sufficient protection against excessive deformations under working loads.

Although there are some obvious similarities in these two types of foundations, there are however some distinct differences. In the case of a shallow foundation, the soil under the foundation base is normally not disturbed, except for changes in effective stresses developed by excavation, construction of the footing, and possibly, backfilling. However, in the case of a deep foundation, the soil about the foundation is disturbed to some degree. The degree of the disturbance depends on the soil type and the method of placement of the foundation. For bored piles, most of the change occurs around the shaft, where a relatively narrow zone of soil surrounding the pile undergoes some remolding due to soil removal during construction. Depending also on the method of construction, some lateral stress relief may occur before the installation of the pile. In the case of driven piles, significant disturbance occurs in the soil

about the shaft and at the base of the pile. The effect of the disturbance is essentially dependent on soil type. Particular soil responses to disturbance will be discussed subsequently under "Installation Effects".

Ultimate Load

Therefore, as for shallow foundations, the problem is to determine the ultimate load (Q_u) which a deep foundation can sustain. The ultimate load is then that load which can cause either the structural failure of the foundation itself or the bearing capacity failure of the soil. Excluding buckling and bending under the action of lateral loads and failure caused by excessive stresses during pile driving, structural failure is assumed to occur when the maximum axial stress in the shaft exceeds the critical yield stress for the shaft material. This condition may govern the design of the pile for virtual point-bearing foundations. Otherwise, the ultimate load is determined from considerations of bearing capacity failure of the soil. It should be noted that an efficient design is one in which the ultimate structural capacity of the pile approximates the ultimate bearing capacity of the soil.

The determination of the ultimate bearing capacity of the soil is further complicated, since it is rarely clearly defined. Unlike shallow foundations for which the mode of shear failure (e.g. general shear, local shear, or punching shear failure) may vary with the soil type, rate of loading,

and other factors, deep foundations almost exclusively fail in punching shear under the foundation point, accompanied or preceded by direct shear failure of the soil along the foundation shaft (Vesic, 1970). However, there may be no visible collapse of the foundation and no clearly defined peak load. For this reason, several ultimate-load criteria have been proposed.

These criteria are often based on considerations of plastic (irrecoverable) or total (plastic and elastic) settlement of the pile under the test load. In order to properly apply the criteria, it is necessary to understand some of the essential mechanisms of load transfer between a pile and the surrounding soil. It has been found that full mobilization of shaft resistance requires a relative displacement between the pile shaft and surrounding soil of 0.25 to 0.40 in. (6 to 10 mm), regardless of pile size and length (Vesic, 1969). Furthermore, mobilization of ultimate point resistance of a pile requires a displacement of approximately 10 percent of the pile-tip diameter for driven piles and as much as 30 percent of the pile-tip diameter for bored piles (Vesic, 1969). In addition, in order to reliably apply the aforementioned criteria, the load-deformation curve must be in a "plunging" mode, that is, in a rapid constant rate of deformation under sustained load.

Fellenius (1980) discusses nine of the most widely adopted criteria. It should be noted that a particular criterion may be influenced by the pile load test conducted.

Therefore, the ultimate load criterion is dependent on various contributing factors, and not solely on absolute magnitude of plastic or total limit settlement.

For design purposes the ultimate load is conventionally separated into two components, the shaft or skin load (Q_s) and the base or point load (Q_p). The sum of the ultimate shaft and base resistances constitutes the ultimate load capacity (Q_u):

$$Q_u = Q_p + Q_s \quad (1.01)$$

The implicit assumption of this equation is that the shaft and base resistances are not interdependent. The assumption is not strictly correct but any resulting variance is insignificant for most practical purposes of normally proportioned piles (Poulos and Davis, 1980). Furthermore, this equation assumes that both the pile point and all points of the pile shaft have moved sufficiently with respect to adjacent soil to develop simultaneously the ultimate point and skin resistance of the pile. Full mobilization of the ultimate capacities necessitates that the load has been sufficiently transferred to the soil.

The approach, then, which makes use of the principles of soil mechanics with measured or assumed soil parameters is commonly termed the static method of ultimate load capacity of piles.

Load Transfer

The pile-soil system sustains its load capacity by transferring the load to the adjacent soil. The load transfer at ultimate capacity occurs simultaneously along the pile shaft and the bearing area of the pile point and involves a mass of soil extending to considerable distances from the pile. Therefore, in order to understand the mechanics of load transfer in each particular case of a pile foundation, it is necessary to consider the entire pile-soil system with all its peculiar features as they relate to the geological history of the soil strata, and the effects of installation during construction.

It is not surprising that the mechanics of load transfer between the pile and the adjacent soil represent a relatively complex phenomenon. Some parameters affecting load transfer are often difficult, if not impossible, to express in numerical terms. Nevertheless, some numerical assessment of load transfer characteristics of a pile-soil system is essential for settlement computations and for rational design of pile foundations.

The true distribution of the load transfer can only be assessed by direct measurements at discrete points along the length of the pile. If the measured axial force for a single pile is plotted with respect to depth (z) (Fig. 1), the function $Q(z)$ indicates the load transfer along the pile shaft. At a distance $z = D$, the transferred load is the pile point load (Q_p).

Figure 1
 LOAD TRANSFER FROM A SINGLE PILE
 (from Vesic, 1977)

Along the shaft, the difference $Q(z) - Q_p = Q_s$ represents the pile shaft load. The forces acting on a differential element along the shaft are comprised of both axial ($Q(z)$) and frictional forces ($f_o(z)$). Summation of forces in the vertical direction for a system in equilibrium yields:

$$\Sigma F_z = Q - (Q + dQ) - f_o \cdot P \, dz = 0$$

$$\frac{dQ}{dz} = - f_o \cdot P \quad (1.02)$$

Thus, the rate of change of the axial load is proportional to shaft resistance. Conversely, the slope of the function $Q(z)$ divided by pile perimeter length P yields the distribution of skin resistance (f_o) along the shaft (Vesic, 1970)

$$f_o = - \frac{1}{P} \frac{dQ}{dz} \quad (1.03)$$

It should be noted that as long as $Q(z)$ decreases with depth (z) the function (f_o) remains positive. Some hypothetical shapes of load distribution are shown in Fig. 2. For case (E) the axial load distribution partially increases with depth thus yielding a negative distribution of shaft resistance, which represents the case of negative skin resistance.

Likewise, from measured results of the curve $Q(z)$ it is possible to find the vertical displacement (w_z) of the pile shaft at any depth (z) below the ground surface, provided the cross-sectional area (A) and modulus of elasticity

Figure 2

Typical Simple Distributions of Load Transfer
(from Vesic, 1977)

(E_p) of the pile shaft are known and the vertical displacement of the pile top (w_o) is measured during the loading test. Assuming a linear stress-strain condition and constant A and E_p along the shaft, it can be shown that

$$w_z = w_o - \frac{1}{AE_p} \int_0^z Q(z) dz \quad (1.04)$$

The displacement can also be measured directly when strain rods are implemented. In a similar manner, a plot of the measured displacements versus depth as a function $w(z)$ yields the following

$$Q(z) = AE_p \frac{dw}{dz} \quad (1.05)$$

From the previous definition (Eq. 1.03), the distribution of shaft resistance can be determined from the rate of change of $Q(z)$ with depth.

Both strain gages and strain rods have been used extensively. The strain rods, which do not require sophisticated equipment, have been considered more reliable in the past when the use of electronic instrumentation on building sites posed many difficulties of practical order (Vesic, 1970). With improvements in efficiency and economy of strain gage instrumentation, the use of strain gages has become more common. They offer better accuracy and more efficient recording; in addition, they allow the measurement of residual stresses due to pile driving. The problem of residual stresses will be discussed subsequently. However, even when direct measurements cannot be made along the shaft, much

valuable information about the manner of load transfer can be had from simple tip measurements, using tell-tale data as proposed by the method of Leonards and Lovell (1978). A number of experimental studies have been conducted to test the basic principles of load transfer as previously delineated.

From the studies of piles in clay, the general findings are that piles driven in clay demonstrate a parabolic skin resistance similar to Fig. 2D (Vesic, 1970). The measured average skin resistance was about equal to the average undrained shear strength of the clay, and only small fractions of the load were transmitted by the pile point. Other studies (Seed and Reese, 1957) have also confirmed that the average skin resistance ultimately reached by the piles (in about a month after driving) was approximately equal to the average undrained strength of the undisturbed clay prior to pile driving.

Recent research has been focused on the role of residual stresses (Holloway et al., 1975; O'Neill et al., 1982). It is contended that the parabolic distribution of shaft friction is in part due to the presence of residual stresses, whereas in their absence a linear increasing distribution may be reasonable. In addition, a number of studies have contributed to the belief that the skin resistance of piles in clay is essentially of a frictional nature and, as such, may be predicted from drained strength characteristics of the soil (Burland, 1973; O'Neill et al., 1982).

Similarly for driven piles in sand, the observed load transfer curves indicate skin resistances which for piles in compression appear to be generally parabolic. Measurements of strains along the shaft produced load transfer curves which indicate very low or negative friction in the zone immediately above the base of the shaft. Measurements of normal stress along the skin indicate a clearly parabolic pressure distribution with an average coefficient of lateral pressure close to the active value. Later studies offered a rational explanation for the observed phenomenon (Vesic, 1970). The low skin friction above pile points in sand can be attributed to arching or stress relief in a zone extending to perhaps three pile diameters above the pile point. The interpretation of measured distributions of skin friction is furthered complicated by the presence of significant residual stresses (Holloway et al., 1975).

Nevertheless, the empirical studies have demonstrated, as previously stated, that both the pile point and all points of the pile shaft have moved sufficiently with regard to adjacent soil to develop simultaneously the ultimate point and skin resistance of the pile. The displacement needed to mobilize skin resistance is small, not exceeding 0.4 in. (10 mm), regardless of soil and pile type and pile dimensions. By contrast, the displacement needed to mobilize point resistance may be relatively large, particularly for very large piles, because it amounts to about 10 percent of pile-point diameter for driven piles and as much as 30

percent of pile-point diameter for bored piles. Thus, it can be shown that for very stiff piles, where the pile-point displacement is approximately the same as the displacement of the pile head, the ultimate skin resistance is mobilized much sooner than the point resistance. Furthermore, it has been demonstrated (Vesic, 1970) that the fraction of the total load carried by point resistance is much less at working loads than at the ultimate load. In other words, had the ultimate point resistance been mobilized simultaneously with shaft resistance, a constant relationship would be evident in the measured tests results.

The manner of load transfer is more complicated for relatively flexible piles, in which the displacement of the pile head can be considerable greater than that of pile tip, thus mobilizing the skin resistances in the upper portion of the pile much earlier than those in the lower portion of the pile. In such a case, a substantial load can be carried exclusively by the pile shaft, particularly if the pile is relatively long. A substantial load may need be applied to produce the required relative displacement between the point and adjacent soil before any portion of the total load can be transferred at the tip. This mechanism is illustrated in Fig. 3 which shows the load transfer of a relatively flexible steel-tube pile. It can be seen that all of the applied load was taken by shaft resistance. A load of 40 kips (180 kN) was required to overcome shaft friction and begin to transfer load at the point.

Figure 3

Load Transfer From a Steel Pile Driven Through Compressible Silt to Rock
(from Vesic, 1970)

A number of analytical methods have been proposed in order to make some numerical assessment of the observed phenomenon of load transfer. It is an assessment which is at best approximate, but essential for a rational design of pile foundations. The methods can be categorized into two basic approaches; the transfer function approach and the elastic solid approach (Fig. 4) (Vesic, 1977).

The transfer function provides a unique relationship between the load transferred from an element and the displacement of that element. It contains an implicit assumption that the displacements along any element are not affected by skin loads transferred by other elements, except through the pile itself. Thus, the model is comprised of a set of nonlinear springs which represent the soil, supporting the pile at mid-points of each element and entirely independent of each other. However, as it was previously demonstrated that any applied load (Q) transmitted to the soil must affect other points along the shaft to some distance from the considered element, the concept of a unique transfer function is in obvious contradiction with reality.

The elastic-solid approach attempts to circumvent this inconsistency by considering the effects of transmitted shaft loads (ΔQ) on other points along the shaft. It also allows the possibility that the displacement of a pile element (w_1) may be different from the displacement of the adjacent soil. It should be noted that fundamental to this approach is the assumption that the soil conforms to a

linear stress-strain relationship as a homogeneous, elastic, isotropic solid defined by the two deformation characteristics: modulus of deformation (E_s) and Poisson's ratio (ν_s) (Vesic, 1970)

The principal advantage of using this approach is that it considers the adjacent soil as a continuum. However, there are still a number of limitations. The fundamental assumption of only two deformation characteristics (E_s and ν_s) may not adequately model the soil response. Most soils show stress-strain history, and time-dependent response to loading. In particular, cohesionless soils exhibit increasing E -values as a function of the mean normal stress. In addition, the elastic-solid approach assumes the same E in tension as in compression, whereas most soils can take little stress, if any, in tension.

Recent research with finite elements has attempted to obviate a number of these limitations. The stress-strain response of the soil can be modeled as bilinear and stress dependent (Poulos and Davis, 1980). In addition, the finite-element method (by its nature) allows the introduction of arbitrary inhomogeneties in the soil mass, such as layers and lenses of different characteristics. Some researchers have introduced special joint elements at the pile interface to allow for slip (Ellison et al., 1971; Desai, 1974). Comparisons of finite element analysis with the elastic-solid approach indicate that analysis based on the latter approach gives results of adequate accuracy for

practical purposes, provided that severe variations in sub-soil conditions do not occur along the pile (Lee, 1973).

Residual Stresses

Detailed studies of load transfer from instrumented load tests indicate the presence of residual loads. These residual stresses have been shown to significantly alter the pile response to subsequent load, particularly in the case of relatively deformable piles. Residual stresses are stresses which have developed along the pile shaft and point due to strains, produced from previous load cycles, which were constrained from rebounding elastically.

The generation of residual loads is a function of the load-unload mechanisms of pile installation and appear to depend on the pile-soil system only, independent of the impact pile-driving apparatus used. The pile shaft will tend to return to its initial length upon removal of a compressive load at the head of the pile. With this tendency, the upper portion of the shaft will normally move sufficiently upward against the adjacent soil to develop negative skin friction, which is counterbalanced by residual skin friction in the lower portion of the shaft and, if the applied compressive load was high enough, by residual point load on the pile (Vesic, 1977). In particular for driven piles whose installation consists of periodical loading and unloading of the pile head by dynamic impulses, the result is a generation of substantial residual loads, the existence

of which has an effect on the subsequent load-settlement response of the pile.

These effects were first confirmed quantitatively in field studies at the Arkansas River Project (Mansur and Hunter, 1970) where residual loads up to 50 tons (45 metric tons) were recorded in 53 ft (16 m) long steel pipe and H-piles. As shown in Fig. 5, the presence of residual load causes an apparent concentration of skin friction in the upper part of the shaft, if the pile is loaded in compression. Conversely, for piles loaded in tension after being loaded in compression or immediately after driving, the apparent load distribution shows a tensile load at the pile tip equal in magnitude to the residual point load.

It should be noticed that the magnitude of residual point load (compression) would be measured in tension since the instrumentation was zeroed at the start of the compression load test. Furthermore, the actual residual load distribution cannot be determined unless the instrumentation is zeroed before installation of the load test pile. Consequently, the shapes of unit skin friction distribution ($f_0(z)$) and load transfer distribution ($Q(z)$) curves at failure in both compression and tension remain indeterminate. However, corrected values of point capacity and total shaft capacity can be obtained from subsequent tension load test data, provided load-strain measurements are made near the pile point.

- (1) Measured compression load distribution assuming no stress in pile at start of test.
- (2) Measured compression load distribution after compression test assuming no stress in pile at start of test.
- (3) Measured tension load distribution assuming no stress in pile at start of test.
- (4) Measured tension load distribution after tension test assuming no stress in pile at start of test.
- (5) Tension load distribution adjusted by subtracting Curve 4 from Curve 3.
- (6) Compression load distribution adjusted by adding Curve 4 to and subtracting Curve 2 from Curve 1.

Figure 5

Effects of Residual Loads on Load Distribution in Driven Piles in Sand at Arkansas River

(from Mansur and Hunter, 1970)

In order to better appreciate the effect of residual loads on the mechanism of load transfer, hypothetical distributions of load transfer behavior in compression and tension are presented in Fig. 6 as provided by Holloway et al. (1975). A triangular distribution of skin friction at failure is assumed for the sake of simple analysis as in Fig. 6b. Consequently, the distribution of load transfer ($Q(z)$) can be determined from the previously described load transfer relationships.

Curves 0 and 0' describe the residual load-shear stress distribution state at the start of the compression and tension test, respectively. The shapes of these curves were arbitrarily assumed based on preliminary static analyses of cyclic load test behavior (Holloway et al., 1978). As assumed, the load distribution curve 1 is parabolic for a linearly increasing skin friction. The differences between curves 1 and 0 are plotted as curve 1M. If the instrumentation were zeroed after installation and prior to the load test, the curve 1M would describe the load transfer ($Q(z)$) and distribution of skin resistance ($f_o(z)$) mobilized in the pile-soil system at the current load level. In fact, most load tests make the implicit assumption of a stress-free pile at the start of the test. Therefore, curve 1M would represent the measured load transfer ($Q(z)$) and skin resistance ($f_o(z)$) along the shaft, assuming no residual loads. As previously stated, reported load test results often exhibit a similar parabolic distribution.

Figure 6

Residual Load Distribution Effects

(from Holloway et al., 1975)

Similarly for the tension test, curves 2 present what would be the actual load distribution. Curve 2M represents the load distribution obtained from measured data. For conventional load tests, this would represent the mobilized resistance assuming no stress in the pile at the start of the test. Note that the measured tension force at the pile tip corresponds to the residual compression load at the time the gage was zeroed. This phenomenon was also evidenced in the results of the Arkansas River Project.

There are several important implications concerning the effect of residual loads. As previously stated, curves 1M describe the shapes of the (hypothetical) measured load and computed shaft resistance curve, uncorrected for the residual load distribution. It should be noted that the unit shaft resistance curve 1M shows a linear increase of shear stress only at shallow depth, after which it drops to a new constant value. In addition, the curve 1M defines a greater area beneath the curve and, thus, a higher shaft capacity than is actually present. Likewise, the measured (uncorrected) value of point bearing capacity underestimates the true value by the magnitude of the residual point load.

The aforementioned hypothetical case, which gives at best only a qualitative assessment of the effect of residual loads, nevertheless yields the following observations (Holloway et al., 1975). By neglecting residual stresses in analyzing pile load tests results one usually (1) overestimates pile shaft capacity, (2) underestimates point

capacity, and (3) incorrectly determines the actual resistance distribution at failure.

Analytical techniques have been proposed which incorporate the effects of residual loads in approximating pile-soil interaction behavior throughout installation and load testing (Holloway et al., 1978). A caveat at this point may be warranted. The incorporation of residual load effects may greatly enhance present analytical techniques of predicting pile driveability and accompanying structural stresses. However, more research should be conducted before recommending new correlations for ultimate design capacities.

Installation Effects

The importance of residual stresses in piles, induced during installation, on the mechanism of load transfer has been demonstrated. There are, however, also other installation effects which may significantly alter the load response of the pile.

Cohesionless Soils

A prominent effect of driven piles in cohesionless soils is the compaction of the soil by displacement and vibration, resulting in permanent rearrangement and some crushing of the particles. The distinct advantage of driven piles in loose soils is that the load capacity of the pile is increased as a result of the increase in relative density caused by driving. The extent of compaction of sand and the

increase in relative density around the pile have been studied by Meyerhof (1959) and Robinsky and Morrison (1964).

The results of both investigations were found to be in broad agreement. It was found that in an initially very loose sand (relative density $D_r = 17\%$), soil movement extended 3 to 4 pile diameters from the side of the pile and 2.5 to 3.5 diameters below the pile tip. In a medium sand ($D_r = 35\%$), the movement extended to some 4.5 to 5.5 diameters from the side and 3.0 to 4.5 diameters below the tip. Although both results are in general agreement, Meyerhof (1959) predicts the amount of compaction near the tip to be greater, and that near the top of the shaft to be less.

The extent of compaction also effects the load behavior of pile groups. When groups of piles are driven into a loose sand, the soil around and between the piles becomes highly compacted, and if the pile spacing is sufficiently close (less than about six diameters), the ultimate load capacity of the group may be greater than the sum of the capacities of the individual piles, that is, the efficiency of the group is greater than 1 (Poulos and Davis, 1980). However, in very dense sand, pile driving may cause loosening rather than compaction, yielding a group efficiency of possibly less than 1. This might be expected due to the tendency of dense sand for dilation. These volume increases would result in increases in the void ratio which means a decrease in density. Furthermore, dynamic loads due to pile driving may cause local liquefaction. Cyclic loads can cause a

build up of pore pressures in medium to dense saturated sands, and hence induce measurable strains in the soil which would ordinarily exhibit a dilative response under static loads. However, the actual behavior of the single pile or the pile group will depend on a number of contributing factors which are difficult to measure.

The installation procedures for a bored pile are likely to disturb and loosen the soil to some extent. Also, some loosening is liable to occur at the bottom of the pile as a result of baling or "shelling-out" the hole, and when this

is done under water, the upward surge on withdrawal of the baler or shell can loosen the soil for several feet below and around the pile (Poulos and Davis, 1980). Consequently, based on empirical studies, Meyerhof (1976) recommends certain reductions in ultimate shaft and point capacities.

Cohesive Soils

The effects of pile driving in clays have been classified into four major categories by de Mello (1969):

- (a) Remolding or partial structural alteration of the soil surrounding the pile.
- (b) Alteration of the stress state in the soil in the vicinity of the pile.
- (c) Dissipation of the excess pore pressures developed around the pile.
- (d) Long-term phenomena of strength regain in the soil.

Based on observations from load tests to failure carried out on piles at different times after their installation, it has been found that the undrained strength of a clay is initially decreased considerably because of driving, but that significant regain of strength occurs with time. Hence, it may be generally expected that the driving of piles into clay will initially cause some (or even considerable) loss in undrained strength of the clay because of remolding at constant water content. Subsequently, the strength will usually increase because of a combination of

two factors : thixotropic regain of undrained strength as the structural bonds destroyed by remolding are at least partially restored, and increase resulting from local consolidation of the clay produced by dissipation of excess pore-water pressures that arise from the increase in stress in the soil surrounding the pile (Poulos and Davis, 1980).

The available evidence (de Mello, 1969) suggests that immediately after driving, the amount of remolding decreases from about 100% at the pile-soil interface to virtually zero at about 1.5 to 2.0 diameters from the pile surface. The rate of increase of soil strength subsequent to pile driving, other than for thixotropic regain, is related to the rate of dissipation of excess pore pressure (Soderberg, 1962). Measurements of pore pressure at the pile face indicate that the excess pore pressures may become equal to or even greater than the effective overburden stress. However, the induced excess pore pressures decrease rapidly with distance from the pile and generally dissipate very rapidly at all points. Furthermore, as a consequence of excess pore pressures, significant negative friction is developed as a result of reconsolidation of the soil around the pile (Fellenius and Broms, 1969).

Heave of the soil surrounding the pile is another significant effect of pile driving in clay. The movement (heave) caused by pile driving may have a significant effect on adjacent structures and may also cause previously installed piles to displace. In such cases, it may be

necessary to redrive earlier piles or to initially use bored piles.

From measurements of displacements resulting from pile driving in clays, it has been found that the soil displacements are less for piles driven in sensitive clays than for corresponding piles in insensitive clays, and that when piles penetrate alternating strata of fine-grained soil and granular materials, the observed surface-heave may be much less than that which would have occurred in insensitive clay soils (Hagerty and Peck, 1971). Concerning lateral movements, it was found that driven piles tended to be displaced away from subsequent driving, with movements continuing for a considerable length of time after completion of driving.

The adhesion between pile and soil is significantly altered by the installation of bored piles. The adhesion has been found to be less than the undrained cohesion before installation, mainly because of softening of the clay immediately adjacent to the soil surface. Poulos and Davis (1980) contend that softening may be caused by :

- (a) Absorption of moisture from the wet concrete (when present).
- (b) Migration of the water from the body of the clay toward the less highly stressed zone around the borehole.
- (c) Water used in the boring process to facilitate operation of the cutting tool.

Some of these causes may be ameliorated by good drilling technique.

Disturbance and softening of the clay beneath the pile base may result in increased settlements, in particular for belled piers, for which the base carries a major portion of the load. However, base disturbance and softening should have a negligible effect on the ultimate bearing capacity of the base because of the comparatively large mass of clay involved in the shear failure of the clay (Skempton, 1959). The friction resistance developed in the clay along the pile is probably restricted to the narrow softened zone. Nevertheless, due to construction effects (e.g., entrapped drilling mud along the shaft), no reliance should be placed on load transfer developing within three shaft diameters of the surface or one diameter of the base.

Negative Shaft Friction

Positive skin friction is mobilized by relative displacements of the pile with respect to the soil. Negative friction, on the other hand, is mobilized by relative displacements of the surrounding soil with respect to the pile. Load transfer $Q(z)$ is characterized as increasing with depth. Observations indicate that a relative movement of 0.6 in. (15 mm) of the soil with respect to the pile may be sufficient to mobilize full negative skin friction (Vesic, 1977). According to Varde et al (1969), this relative displacement need not be more than 10 mm to develop the full negative skin friction. The downward movement of the soil with respect to the pile shaft is due to other surface loads or any other

cause such as pumping of water from one of the aquifer strata in the profile, which produces an increase of effective stress and settlement of adjacent soil. Under such movements, a negative friction is exerted against the pile shaft by the soil. The negative friction may be assumed as a superimposed axial load which must be transmitted to a deeper stratum. Consequently, this can cause excessive settlements of the piles, with severe damage or even collapse of the structure supported by the piles.

Measurements of magnitude and distribution of negative skin friction show its dependence on the effective ground stress, similar to the formulations for positive skin friction (Johannessen and Bjerrum, 1965; Endo et al., 1969; Fellenius and Broms, 1969). However, the negative skin friction develops only along the portion of the pile shaft where the soil settlement exceeds the downward displacement of the pile shaft. A neutral point is established, whose position coincides with the point of no relative movement between the pile and the adjacent soil (Endo et al. 1969). Some studies have measured the neutral point at approximately 75 percent of pile length (Fellenius and Broms, 1969; Walker and Darvall, 1973). However, other investigators (Bozozuk, 1972; Cognon, 1972) show that the neutral point can be located at a higher or lower position. It is influenced by factors such as relative compressibility of the pile shaft and underlying soil with respect to the surrounding soil, relative magnitude of axial load in the pile with respect to

effective stress change that causes settlement of surrounding soil, as well as the position of the most compressible stratum in the overall soil profile (Bozozuk, 1972; Buisson et al. 1960). Below the neutral point, friction remains positive and the load transmitted to the pile point may be significantly less than the maximum axial load in the pile.

A number of methods have been proposed to reduce negative skin friction. The pile shaft can be isolated from the settling soil by use of an outer casing. Alternatively, the pile can be placed in a pre-drilled hole of larger diameter than the pile shaft, filling the gap with bentonite slurry, which in view of its low friction characteristics, limits the negative friction to a relatively low value (Vesic, 1977). In soft soils, some success has been achieved by coating the pile with a bitumen layer.

However, these methods could be expensive. Hence, an estimate of the additional load due to negative friction should be established and taken into account in the design. A wide variety of methods has been proposed for that purpose. Kézdi (1975) suggested (as a rough estimate) to assume that negative friction on the contact between the pile shaft and soft soil is equal to the pile perimeter multiplied by the thickness of the soft layer and by the average value of the undrained shear strength (c_u) of the undisturbed soft soil.

Bowles (1982) suggested the following procedures to make the required estimate:

1. For a fill layer overlying a compressible soil deposit:

$$P_{n_f} = \frac{p_l K_o \gamma' H_s^2}{2} \quad (1.04)$$

where:

P_{n_f} = additional pile load due to negative friction

p_l = pile perimeter

γ' = effective unit weight of compressible layer

H_s = thickness of compressible layer

K_o = coefficient of lateral earth pressure at rest

2. For cohesive fill overlying cohesionless soil:

$$P_{n_f} = \frac{p_l K_o \gamma' H_f^2}{2} \quad (1.05)$$

where:

H_f = thickness of fill layer

3. For piles at small spacings, the negative friction forces may effectively act on the block perimeter, rather than on individual piles. Two cases should then be investigated:

a. The negative skin friction as the sum of the individual piles:

$$Q_{n_f} = n \cdot P_{n_f} \quad (1.06)$$

where:

Q_{n_f} = additional group load due to negative friction

n = number of piles in a group

b. The block skin friction is:

$$Q_{n_f} = r H_s p_2 + \gamma' H_s A' \quad (1.07)$$

where:

r = center-to-center spacing between piles

H_s = thickness of very soft layers or recently placed fill penetrated by the piles

A' = area of pile group enclosed in perimeter

p_2 = perimeter of pile group

The maximum value from Eq. (1.06) or (1.07) should be taken. Usually the value of Q_{n_f} obtained from Eq. (1.07) is the critical one. The value of Q_{n_f} should be added to the column load, and modifications in the number of piles or in spacings may be required to account for the effect of negative skin friction.

There are also a number of rather complex methods to calculate the downward drag due to negative friction (e.g., Poulos and Davis, 1980). However, Kézdi (1975) had stated that a very refined analysis is in general not justified, since the numerical values of the input parameters are only approximately known.

Therefore, negative skin friction can be expected above the point of the pile where the relative downward displacement of the soil with respect to the pile exceeds 0.6 in. (150 mm). Its magnitude is proportional to the effective vertical stress and is similar to positive friction in computation. As long as the downdrag force is less than the live load, its negative effect on pile foundation performance is minimal.

CHAPTER 2

PILE-SOIL DESIGN CAPACITY

As previously stated, the summation of the point and shaft resistances constitutes the ultimate static load capacity of the pile, that is,

$$Q_u = Q_p + Q_s \quad (2.01)$$

where

Q_p = ultimate point resistance

Q_s = ultimate shaft resistance

The fundamental assumption concerning load transference of this expression is that ultimate point and shaft resistances are mobilized simultaneously.

The ultimate point resistance (Q_p) is generally evaluated analogously with respect to bearing capacity theory as

$$Q_p = A_p (cN_c + \sigma_{v_p} N_q + 1/2 \gamma d N_\gamma) \quad (2.02)$$

where

A_p = area of pile point

c = cohesion of soil

σ_{v_p} = vertical stress in soil at depth

of pile point

γ = unit weight of soil

d = pile diameter

N_c, N_q, N_γ = bearing capacity factors, which are primarily functions of the angle of internal friction (ϕ) of the soil. They refer to the influence of soil cohesion, surcharge, and frictional resistance induced by weight, respectively.

The shaft resistance (Q_s) is determined by integration of the pile-soil shear strength (τ_a) over the surface of the shaft. Shear strength (τ_a) is characterized by the Coulomb expression

$$\tau_a = c_a + \sigma_h \tan \phi_a \quad (2.03)$$

where

τ_a = pile-soil shear strength

c_a = adhesion

σ_h = horizontal stress in soil along pile shaft

ϕ_a = angle of friction between pile and soil

The horizontal stress (σ_h) is related to the vertical stress (σ_v) as

$$\sigma_h = K_s \sigma_v \quad (2.04)$$

where

K_s = coefficient of lateral pressure

Making the necessary substitutions, the expression for pile-soil shear strength becomes

$$\tau_a = c_a + K_s \sigma_v \tan \phi_a \quad (2.05)$$

and integrating along the length of the pile shaft, the shaft resistance is evaluated as

$$Q_s = \int_0^L P \tau_a dz$$

$$Q_s = \int_0^L P (c_a + K_s \sigma_v \tan \phi_a) dz \quad (2.06)$$

where

P = pile perimeter

L = length of pile shaft

Therefore, a general expression for the ultimate capacity of a single pile can be developed from a combination of Eqs. 2.01, 2.02, 2.06,

$$Q_u = A_p (cN_c + \sigma_v N_q + 1/2 \gamma N_\gamma) + \int_0^L P (c_a + K_s \sigma_v \tan \phi_a) dz \quad (2.07)$$

Equation 2.07 is a general expression for the ultimate static load capacity of a single pile. The general expression will subsequently be modified to accommodate particular theories and soil conditions. Nevertheless, certain soil parameters are required to evaluate the above equation.

The ultimate load capacity for undrained or short-term conditions is computed on the basis of the soil parameters c , ϕ , c_a , and γ which are appropriate for undrained conditions. These variables may be readily determined through laboratory testing or assumed from past performance of similar soils. The vertical stresses, σ_v and σ_{v_p} should be evaluated as total stresses for undrained condition. The ultimate load capacity for long-term conditions should be determined by the drained values for the soil parameters. The vertical stresses in this case should be evaluated as the effective vertical stresses σ'_v and σ'_{v_p} . The vertical stresses are usually taken to be the overburden or surcharge stresses. There is, however, considerable evidence to suggest a limiting value for vertical stress due to overburden (Vesic, 1967; Meyerhof, 1976).

A number of theoretical and semiempirical methods have been proposed to analyze point bearing capacity. These methods are often based on limit equilibrium solutions of classical plasticity theory. Common assumptions for soil behavior as delineated by Holloway et al. (1975) are:

- (a) Mohr-Coulomb failure criterion
- (b) Shear strength independent of strain
- (c) Elastic deformations negligible with respect to plastic deformations
- (d) Volume changes due to shear and

normal stresses are negligible.

These conditions describe a
rigid-plastic, Mohr-Coulomb solid.

The limit equilibrium methods analytically approximate the behavior of the physical system with a representative kinematic failure mechanism and associated boundary and discontinuity conditions. These methods ultimately provide approximate solutions to idealized continuum mechanic problems.

Notice

Chapters 3 through 6 (pages 38 through 191) are omitted from this copy of the report. They may be obtained, at the cost of reproduction, from:

Joint Highway Research Project
School of Civil Engineering
Purdue University
West Lafayette, Indiana

47907

CHAPTER 7

PILE PROGRAM AND APPLICATIONS

A computer program PILE was written to facilitate the computation of ultimate pile capacity and the concomitant driveability analysis. The program is designed therefore to conduct both static and dynamic analyses of the pile-soil system. The models incorporated into the structure of the program were chosen essentially for the following reasons: extensive corroboration of the models with empirical observations and practicality of the required parameters with reference to subsurface investigations. An implicit criterion of the chosen models is also the generality of application to difficult soil conditions and pile configurations. Whereas, a number of models have been discussed which have proven to be equally reliable, they are often constrained to a certain data field or by impracticality of pertinent parameters.

In general, the program is structured in accordance with most of the requirements and recommendations of the Meyerhof (1976) model for static analysis of ultimate capacity. For cohesive soils, it was necessary to present alternate models (Meyerhof, 1976; Tomlinson, 1971) which would reflect critical assumptions concerning shear strength

response of soils under design loads.

The dynamic methods employed in the program consist of a number of pile-driving formulas and/or a wave equation analysis of the pile-soil system. The pile-driving formulas incorporated were chosen on the basis of conventional construction specifications and reliability of predictions. They include the equations conventionally used by highway agencies to select bridge foundations. The wave equation analysis of the pile-soil system is based on the WEAP program developed by Goble and Rausche (1976). The data required to run a wave equation analysis have been reduced and simplified by Goble and Associates, Inc. (1980). However, it should be recognized that the modifications limit the analysis to conventional pile-soil systems. More special applications (i.e., partially embedded piles, point bearing on rock) would require the use of the original WEAP program. Other limitations will become apparent with the subsequent discussion of parameters.

Some of the side topics like residual stresses and rate of pore pressure dissipation (for settlement computations) are not included in the program since they have not been well established yet. Some research still needs to be done in these topics. Hence, they will be handled in the final report.

User's Manual

The program software was written specifically for the VAX computer system at Purdue University. It has also been slightly modified for the IDOH IBM 370 system. It is opera-

tive on both systems. Further pertinent details are given in: Appendix A, PILE Program; Appendix B, Pile Input Form; and Appendix C, Examples.

It should be noted that a static analysis may be conducted independent of either dynamic methods. However, neither

response of soils under design loads.

The dynamic methods employed in the program consist of a number of pile-driving formulas and/or a wave equation analysis of the pile-soil system. The pile-driving formulas incorporated were chosen on the basis of conventional construction specifications and reliability of predictions. They include the equations conventionally used by highway agencies to select bridge foundations. The wave equation analysis of the pile-soil system is based on the WEAP program developed by Goble and Rausche (1976). The data required to run a wave equation analysis have been reduced and simplified. However, it should be recognized that the modifications limit the analysis to conventional pile-soil systems. More special applications (i.e., partially embedded piles, point bearing on rock) would require the use of the original WEAP program. Other limitations will become apparent with the subsequent discussion of parameters

Users' Manual

The program software was written specifically for the VAX computer system at Purdue University. It has also been slightly modified for the IDOH IBM 370 system. It is operative on both systems. Further pertinent details are given in: Appendix A, PILE Program; Appendix B, Pile Input Form; and Appendix C, Examples.

It should be noted that a static analysis may be conducted independent of either dynamic methods. However, neither

dynamic analysis may be performed in this program without having previously modelled the pile-soil system as per the requirements of the static methods. The required parameters for the computer program will be subsequently discussed.

The first line of data characterizes the global conditions of analysis of the pile-soil system.

NSTRAT	The number of soil layers or strata penetrated or expected to be penetrated by the pile. A maximum of 10 soil layers may be inputted.
B	Diameter of the base of the pile. It is used to determine the critical depth. (ft)
APNT	Area at the base of the pile. Required to determine point capacity. (ft ²)
ASHFT	Unit surface area of the shaft. This quantity will be generated internally for tapered piles. (ft ² /ft)
LEN	Length of the pile. This parameter is only used to assess the ultimate soil resistance mobilized for a given pile length. (ft)
QPRMT	Ultimate design capacity of pile. This parameter is used to determine the necessary length of the pile required to develop the prescribed ultimate design capacity. (ton)
SHEAR	This parameter will default the corresponding model of analysis. Input 'DC' for drained shear strength-long term capacity. This condition is generally applicable for piles in cohesionless soils or for an effective stress stress analysis (β -method) of piles in

cohesive soils. Input 'UU' for undrained shear strength-short term capacity. Undrained conditions correspond to the α method of analysis.

PRCNT The percent of pile taper. It is only required for tapered piles. It is assumed that the pile is tapered along its entire length.

INSTL The method of pile installation. As previously discussed the method of installation can significantly alter the bearing capacity of the pile. One of four possible installation techniques must be inputted: 'driven', 'bored', 'bulbous', or 'h pile'.

NEQN Number of dynamic equations for which pertinent parameters will be supplied. There are five possible equations. Any combination of the 5 equations may be used.

WAV Wave equation analysis. Input '1' for option to use WEAP. More parameters will subsequently be required. Input '0' if only a static analysis is desired.

As previously noted, the parameter SHEAR essentially determines the analytical model for computation. The following input parameters are required if SHEAR is defined as 'DC'. In effect, the following input data characterize the soil conditions.

SOIL This parameter is used to describe three basic soil types, sands, silts, and clays. It is, of course, an idealized model of the soil strata. In-situ soil conditions may not exclusively fall into one particular category but may exhibit some combination of the three basic types. The operator must then determine which soil type will predominantly govern the behavior of a particular stratum. For example, some glacial tills may exhibit bearing capacities similar to sands. A

soil type is then defined in terms of three possible abbreviations:
 'ARN' for predominantly sandy stratum
 'LIM' for predominantly silty stratum
 'ARG' for predominantly clayey stratum
 Input is required for all layers.

- XL This parameter defines the thickness of a particular stratum. A maximum of 100 ft per stratum is permissible. Input required for all layers.
 (ft)
- N Blow count obtained from the Standard Penetration tests. Input required for all layers. Input when possible for cohesive layers.
 (blows/ft)
- NCORR A corrected standard penetration resistance may be inputted, otherwise the uncorrected value is defaulted. The corrected value will only be used in the determination of the limit point resistance.
- PHI Angle of internal friction. Input required in degrees for first layer. It serves primarily for classification purposes for subsequent layers.
- NQ Bearing capacity factor. Pertinent values may be obtained from Fig. 7. Input required for first layer. It is optional for subsequent layers since conventional bearing capacity theory in terms of N'_q may not be applicable with reference to the critical depth as previously demonstrated.
- CU Undrained shear strength of a cohesive layer. It serves as the limit shaft resistance for cohesive layers. Input is required for all cohesive layers in a stratified soil condition. This parameter will activate the β method for a effective stress analysis.
 (ton/ft²)
- GAM Unit weight of the soil. Input is required for first layer, optional for subsequent layers.
 (lb/ft³)

ZC	critical depth ratio. Input is required for first layer. Appropriate values may be obtained from Fig. 7. A default of 10 is assumed for subsequent layers.
KS	Coefficient of earth pressure on the shaft. Input is required for first layer only, unless GAM is defined for that layer. It is only required for cohesionless layers. Values are determined internally for cohesive layers.
DEL	angle of skin friction. Input is required for first layer only, unless GAM is defined for all layers. It is also only required for cohesionless layers.
CONSL	Type of consolidation history. This parameter is defined as 'NC' for normally consolidated clays, or 'OC' for overconsolidated clays. It is required only for cohesive layers. It is used internally to determine the coefficient of earth pressure for cohesive layers.
OCR	Ratio of overconsolidation. This input is required whenever CONSL is defined as 'OC'. It is used to modify the internal values of the coefficient of earth pressure. However, if a value of OCR cannot be obtained directly from field or laboratory tests, do not provide an assumed value. When omitted, an empirical factor will be applied internally.

In addition, if an analysis for ultimate soil resistance in terms of the undrained shear strength of the clay is desired, then the following parameters are required.

ALPHA	Coefficient of adhesion. Required for all cohesive layers.
NC	Bearing capacity factor. Required for all cohesive layers. It is generally taken to be 9.

The input variable NEQN will call the dynamic formula subroutine. The numerical value of NEQN will determine the number of data lines required. A total of five formulas may be called up, in any combination. There are a number of variables which are common to several formulas. However, each formula has a particular combination of these variables with some possibly additional parameters. The operator must input those variables which are relevant only to a particular formula.

EQN	Name of the dynamic formula. This variable defines the equation to be used and is therefore required for each line of data. The number of data lines cannot exceed the number defined in NEQN. The following dynamic formulas may be called. Input: 'ENR' 'HILEY' 'JANBU' 'DANISH' 'GATES'
W	Weight of hammer. Input is required for all defined equations. (lb)
H	Height of hammer drop. Input is required for all defined equations. (in.)
C	Factor for energy losses. Input is required only for 'ENR'.
EF	Efficiency factor for hammer. Input is required only for the following dynamic formulas: 'HILEY', 'DANISH', and 'GATES'.
N	Coefficient of elastic restitution. Input is required only for the 'HILEY' formula.
WP	Weight of pile. Input is required

only for the following formulas:
 'HILEY', and 'JANBU'.
 (1b)

- C1 Elastic compression of pile head and cap. Input is required only for 'HILEY' formula.
 (in.)
- C3 Elastic compression of soil, quake. Input is required only for 'HILEY'.
 (in.)
- EP Modulus of elasticity of pile. Input is required only for the following formulas: 'HILEY', 'JANBU', and 'DANISH'.
 (lb/in.²)
- FS Factor of safety. Input is required for all formulas. This variable differs from other previously applied safety factors in that a number of the formulas have been developed with an implicit value. For example, the 'ENR' formula has an implicit value of 6 for its safety factor. Typical values for other formulas may range between 3 and 4. However, since the static analysis for ultimate soil resistance implicitly assumes a safety factor, the input value for FS must be a partial factor of the value required for a particular formula. For example, if a safety factor of 2 is assumed for the static assessment of ultimate resistance, then FS = 3 is required if the 'ENR' formula is defined.

The unmodified version of the WEAP program allows for two alternate data input forms, a short and a long form. In essence, the short form, although very practical due to the simplicity of the data requirements, is only applicable for relatively simple pile-soil models. Due to the simplicity of the pile-soil model, it is able to internally discretize

the model into acceptable elements for analysis. One disadvantage of the automatic model discretization is that pile segments may tend to straddle different soil layers. However, the wave equation analysis can only consider one type of soil response per pile element. In order to avoid this situation in stratified soils, the operator would have to manually generate the discretized pile-soil model. Similarly, any other exception or model peculiarities would necessitate the manual generation of the discretized model. This process would entail the long form of the input data, and can be quite cumbersome.

In the present form of the WEAP subroutine, the long input form has been simplified by internally computing a number of the required input quantities, such as, pile element stiffness, pile segment weights, and relative ultimate resistances. One clear advantage of the modification is that the relative ultimate resistances will conform to the resistance distribution as determined by a previous static analysis. This obviates the need to prescribe relative percentage distributions along the pile.

Therefore, the operator is only required to simply discretize the pile with reference to soil layers and define certain parameters relevant to the pile, soil, and driving system. The following parameters are required to run a wave equation analysis of the pile-soil model.

NSEG

Number of pile segments. The pile must be discretized into a finite

number of pile segments of acceptable lengths. It assumes a fully embedded pile.

NLAYER	Number of soil layers. The number of soil layers penetrated by the pile must be defined.
ATP	Area at top of pile. (in. ²)
E	Modulus of elasticity of the pile (ksi)
RP	Unit weight of pile material

The following variables define certain parameters for each discrete pile segment.

NELM	Segment number. Each segment must be numbered in sequential order.
NSTRAT	Soil layer number. Each pile segment corresponds to a particular soil layer. Input is required for first segment of each layer, and optional for subsequent segments of that layer.
ALPH	Length of segment. The pile must be discretized in segment lengths characterized by whole numbers. Input is required for first segment of each layer, and optional for subsequent segments of that layer unless a change in length of the segments occurs for that layer. (ft)
A	Area of element. Input is required for first element of each layer. Average area per segment is allowed for tapered piles. (in. ²)
QS	Soil quake. Input is required for first segment of each layer and optional thereafter. A value is also required for the pile tip. (in.)

SJ Soil damping factor. Input required for first segment of each layer and optional thereafter. A value must also be defined for the pile tip if different from the previous segment.

The following variables define certain characteristics of the driving system.

IHAMR Type of hammer. The WEAP program package has an extensive inventory of conventional hammer data. Each hammer is defined by a numeric code.

IPEL This variable defines how one arrives at the pile segment parameters. The modified version presently requires that IPEL = 2 since the pile masses and stiffnesses will be determined internally.

N Number of pile segments. The number of pile segments must be defined again.

NCROSS This variable, for most cases, is set to 0 to define a uniform pile.

IBEDAM The pile material damping value in percent of pile critical damping. Recommended values are for steel (1), concrete (3), and timber (5). For zero damping set IDEDAM = -1. These values are usually of minor importance and sufficient information for their choice is not yet available.

ISMITH This variable defines the type of damping factor to be used, whether Smith or Case. If set equal to 1, the damping parameters are assumed to be of the Smith type, (i.e., dimension sec/ft). If set equal to 0, the damping parameters are assumed to be of the viscous type but non-dimensionalized by dividing by EA/c, Case damping.

ITYS Type of skin friction distribution.

For the present modified version, the variable must be set equal to -2 since several variables, such as quake, damping, resistance distribution, will have been pre-defined.

The following variables define certain parameters of the driving train which include the capblock, helmet, and cushion. The term CAP will include the striker plate, capblock, and helmet. A cushion atop the pile may also be specified.

WT.CAP	Weight of CAP. This is essentially the weight of the helmet. (kips)
STIFF.CAP	Stiffness of capblock. (k/in.)
STIFF.CUSHION	Stiffness of cushion. This quantity must be manually computed based on the elastic modulus of the material. (k/in.)

If either the capblock or cushion is not present, its stiffness should be 0.

Four coefficients of restitution have to be specified for Anvil, Cap, Pile Top, and Pile Cushion for diesel hammers or Cap, Assembly, Pile Top, and Pile Cushion for air/steam hammers. If they are not specified, a value of 0.85 will be assumed except for the cushion when not present. The coefficient of restitution of the cushion will be set to 1.0 when not present. It is recommended to use 0.85 for steel-to-steel impact.

The efficiency of the hammer may also be specified.

EFF Efficiency of hammer. A hammer efficiency of 0.95 for diesel hammers will be used when not specified. For air/steam hammers no specification will result in the use of an internal value from hammer tables.

Reliability of PILE Program

The reliability of the PILE program was examined by the author by comparing the computed results from the program with published load test data. The studied cases were taken from the published literature, and comprise a considerable range in both subsurface conditions and pile type. It was however necessary to exclude certain pile types (e.g., step-tapered) due to program limitations. Nevertheless, the data field is comprised of a variety of steel pipe, H-pile, precast concrete, and tapered timber piles. The subsurface conditions also exhibited considerable variability in stratification. In total, some 50 cases which had accompanying load test results were programmed.

The results of the reliability study are encouraging. This is illustrated in Fig. 27, where both calculated and measured ultimate load capacities are plotted. Points along the 45 degree line would indicate perfect correlation between calculated and measured ultimate loads. The amount of variability in the correlations would manifest itself in the manner in which the plotted points scatter about the 45 degree line. For the present study, the points tend to be

Figure 27.

Comparison Between Calculated Values from PILE
and Measured Ultimate Load Capacity from Load Tests

aligned fairly close to the line, with some concentration under the 45° line. Statistically, this distribution represents an average ratio of calculated to measured capacities (Q_{uc}/Q_{um}) of 0.968. Furthermore, the coefficient of correlation was found to be 0.75. Perfect correlation would have rendered a ratio of 1.0. The variability of the distribution is characterized by a coefficient of variation of 32%. Therefore, on the average, the PILE program will tend to slightly underestimate the pile capacity.

It should be noted that the variability evident in Fig. 27 is not exclusively the result of inherent deficiencies in the analytical models employed in the PILE program. The variability may also be influenced by the quality of the published load test results and subsurface investigations. The degree to which these factors affect the distribution of results is not presently available, but the extent to which the quality of these factors can be controlled should yield a reduced coefficient of variation. Furthermore, the correlation of the present study may be viewed even more favorably in light of the breadth of the data field.

Only the results of static analyses were considered in the reliability study. For piles in predominantly cohesionless soils, the Meyerhof method was followed, employing the necessary parameters as discussed in Chap. 3. For piles in cohesive soils, the alpha method was used as discussed in pg. 90. In cases where there was insufficient data concerning the cohesive strength of the soil, a drained strength

condition was assumed, and the beta method was employed for the analysis as detailed in Chap. 4 starting on pg. 98.

Case Histories from IDOH

Several case histories of unusual or difficult pile-driving conditions were provided by Indiana Department of Highways (IDOH). A number of these case histories were restudied with use of the PILE program in order to determine if any of these conditions could have been foreseen. The case histories chosen for study are considered representative of the type of problems which may be encountered under present pile-driving practice. For all cases studied, the results of subsurface investigations provided were assumed to be reasonably representative of in-situ soil conditions.

Project B-14265

This case, in many ways, is typical of frequent difficulties in field operations. A tip elevation has been predetermined at which the desired bearing capacity would be developed. The problem, then, is to choose a driving system which can deliver efficiently the necessary energy to penetrate the pile to the desired tip elevation.

In this case, 14 in. diameter steel shell piles with an allowable load of 40 tons were required to be driven to a predetermined tip elevation. Installation was initiated with a Vulcan II hammer and completed with a Vulcan I. Upon termination of the driving sequence, the bearing value in tons of the pile was determined by use of the Engineering News formula.

The PILE program yielded the following results. First, assuming that the subsurface data were representative of the in-situ conditions, the ultimate static capacity of the pile was approximately 42 tons. The ultimate bearing capacity is considerably lower than would be anticipated for an allowable bearing capacity of 40 tons.

Secondly, the results of a dynamic analysis indicate that the use of a Vulcan II would result in possible hard driving conditions. The analysis was initially conducted assuming the manufacturer's rated energy. At the efficiency level of the rated energy, the resulting blow counts (60 blows/ft) were not excessive, but were close to the upper limit of acceptable driving criteria. However, as previously demonstrated, the actual efficiencies in the field may vary considerably. At these lower efficiency levels, the Vulcan II becomes incapable of efficiently driving to the desired tip elevation. The Vulcan I hammer, according to the analysis, is able to drive the pile to the designated tip elevation, even at the lower spectrum of efficiency levels. However, it should be recognized that the soil resistance at the stated tip elevation is only approximately 42 tons. If the intended ultimate soil resistance was 80 tons (assuming a safety factor of 2), it is doubtful whether the Vulcan I would have been able to efficiently drive the pile.

Thirdly, the bearing values determined from the Engineering News formula are indeed erroneous in their

implications. If piles of a particular type are driven to virtually the same tip elevation (assuming no significant change in the soil profile), then their bearing capacity is determined by the resistance developed in the surrounding soil and independent of the driving system employed. For this case, the driving records indicate that the bearing value of the piles driven with a Vulcan I are nearly double the amount computed for those driven with a Vulcan II. The discrepancy is inherent in the dynamic equation, which is primarily a function of an impact energy and not of the resistance developed in the surrounding soil.

Project B-14321

An alternate method for termination of pile-driving is also often employed. In this case, the criterion for termination is based on a required final set which would correspond to a desired bearing value of the pile as computed by the Engineering News formula. The total length of the pile is dependent on the dynamic resistance encountered during driving.

In this case, two pile types of different allowable bearing capacity were utilized. A 14 in. diameter steel shell was required to be driven to a depth at which a 40 ton capacity could be developed. Similarly, an H-pile (12HP53) was to be driven to attain a final set representative of 50 ton capacity. A diesel hammer, MKT DE-20/30 was used for the driving system. As previously stated, the final set was

predetermined as per the Engineering News formula.

Installation procedures under this criterion will experience much variability for basically two reasons. First, cohesive soils, in particular, exhibit a reduced resistance during driving due to excess pore pressures generated. Therefore, if termination of driving is made dependent on dynamic resistance, the pile may be driven to a depth considerably greater than required to develop the required static bearing capacity. Second, any change in the energy of the driving system would be reflected in the final length of the pile. If the efficiency of the system is reduced during driving (as is often the case with some types of diesel hammers), the pile driving would be terminated at shallower depths not because of any increased soil resistance but because the energy is not available for the same rate of penetration. In either case, the result would be piles of variable length but allegedly similar bearing capacities.

This condition is evident in the driving records of this case. There is considerable disparity in the total lengths of the piles of comparable recorded bearing values. An analysis using the PILE program demonstrates that these variable lengths represent similar variability in ultimate bearing capacity. It was found that some of the 14 in. shells may have a static capacity as low as 36 tons, whereas capacities of 118 tons were computed for some of the H-piles. In addition, this criterion can result in overdriven

piles which often produces considerable pile damage. A dynamic analysis of the H-piles indicated the the possibility of a highly stressed condition near the tip. Stresses as high as 24 ksi were found in the vicinity of the pile point.

Project B-12309

In this case, the PILE program was not used for several reasons. First, there is insufficient real subsurface data, and presumptive values would be required. Secondly, and more importantly, the driven pile will carry the applied loads virtually in point bearing. In order to accomplish this, the pile needs to be driven through stiff clay and some layers of soft and hard shale. The fundamental assumption of the point capacity models is that there is sufficient deformation of the pile point to mobilize ultimate point resistance. The problem may be circumvented by modelling the rock (shale) layers with comparatively high 'N' values. Similarly, rock layers of variable hardness may be modelled to reflect anticipated in-situ conditions

This case is an example for which the unmodified version of WEAP might be used. The unmodified WEAP would prove more versatile in that the distribution of ultimate soil resistance may be varied to reflect a virtually point bearing pile. The total resistance may be specified at approximately 100% at the point (or any other reasonable estimate). A driving system may then be chosen which could efficiently

penetrate through intermittent layers of hard shale.

It should be noted, however, that the purpose of the WEAP analysis, in this case, is to determine an efficient driving system for a hypothetical resistance model, a model which may not reflect the actual subsurface conditions. In other words, the analysis will be used to determine which system can most effectively penetrate intermittent hard layers in order to reach bedrock. Furthermore, the analysis will yield valuable information concerning the dynamic stresses developed during driving, which in this case may be roughly doubled. The analysis should be scrutinized for possible stresses in excess of the yield criteria of the pile material.

CHAPTER 8

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

The present study presumes that piles are the most appropriate foundation type based upon structural and loading considerations, subsurface conditions, estimated performance, and economy of construction. Thus, this study reviews recent advances in this field and presents a comprehensive method of analysis in the form of a computer program. It was necessary to design the program in such a manner that it would be generally applicable to various pile types and soil conditions, viz., bridge piles in the State of Indiana.

Summary

General Criteria

In some essential respects, the design of deep foundations is analogous to shallow foundations. The critical problem is to determine the ultimate load which a deep foundation can sustain. The ultimate load is then that load which can cause either the structural failure of the foundation itself or the bearing capacity failure of the soil. For most situations, ultimate load is determined by soil failure. The failure mechanism is characterized by punching

shear under the point, accompanied or preceded by direct-shear failure along the shaft.

Load Transfer

The ultimate load of the pile-soil system is sustained by transferring the load to the adjacent soil. The load transfer at ultimate capacity occurs simultaneously along the pile shaft and the bearing area of the pile point and involves a mass of soil extending to considerable distances from the pile. The displacement needed to mobilize skin resistance is small compared to that for point resistance. Ultimate skin resistance is reached much sooner than point resistance and the portion of the load carried by the point is smaller at working conditions than at failure.

The manner in which the ultimate load is transferred to the adjacent soil can be significantly altered by the presence of residual stresses in the pile. The generation of residual stresses is a function of the cyclic loading mechanisms of pile installation. Residual stresses are produced by shaft friction in preventing elastic rebound of an unloaded pile, thus entrapping elastic strains. The presence of residual stresses may alter the distribution of shaft friction from a triangular mode of transfer to a parabolic mode. Neglect of residual stresses in analyzing pile load tests results in usually overestimating pile shaft capacity, underestimating point capacity, and incorrectly determining the actual resistance distribution at failure.

Pile type and installation effect may also alter the load response of the pile. The prominent effect of driven piles in cohesionless soils is the compaction of the soil by displacement and vibration, resulting in permanent rearrangement and some crushing of the particles. The principal effect of driven piles in cohesive soils is the structural alteration of the soil surrounding the pile. The alterations generate considerable excess pore pressures around the pile with subsequent strength regain with the dissipation of the pore pressures.

Negative skin friction is mobilized by relative displacement of the surrounding soil with respect to the pile. The negative friction may be considered as a superimposed axial load. Neglect of this possible condition can cause excessive settlements of the piles, with severe damage or even collapse of the structure supported by the piles. Several methods are available to reduce negative skin friction, including installing piles within casings or coating them with bitumen.

Static Analysis

For design purposes, the ultimate load is separated into two components: the point load, and the shaft or skin load. The fundamental assumption is that ultimate point and shaft resistances are mobilized simultaneously. The analytical models chosen for design purposes are distinguished with respect to soil type.

Cohesionless Soils

Ultimate load capacity in cohesionless soils in terms of both point and shaft resistance is expressed as a function of overburden pressure. However, the analytical relationships hold only when the pile point is above a certain critical depth. Below the critical depth, the point resistance and average skin friction remain practically constant in a homogeneous sand deposit, due to effects of soil compressibility, crushing, arching, and other factors. A method is proposed by Meyerhof which determines the point and shaft capacities in terms of limit values. The model proposed by Vesic requires detailed knowledge of the strength and deformation characteristics of the soil strata and also of the variation of density and water content within those strata. The limit values are directly determined from such field tests as the static-cone penetration test or standard penetration tests.

Certain recommendations are made to modify the estimates of ultimate load capacity to account for the effects of pile type and installation procedures. In addition, a reduction factor is introduced to account for scale effects of large diameter piles.

Recent correlations have suggested that both ultimate point and shaft resistance may not reach a limit value but indeed continue to increase with the mean normal stress, albeit at a considerably slower rate. However, the curves developed for the new correlations indicate an inherent

inconsistency in the results. Furthermore, the new correlations are not supported by more recent studies. A statistical study of the results of instrumented load tests confirms, if only qualitatively, the occurrence of a limit value for both unit point and shaft resistance.

Pile foundations are normally constructed as groups of closely spaced piles. Pile spacing should be such that the group capacity is not less than the sum of the capacities of the individual piles. However, the ultimate load of a group may be less than the sum depending on soil type, size and shape of the group, spacing and length of piles, and construction procedures.

The allowable design load for groups is often determined from settlement considerations. The settlement analysis of pile foundations is somewhat similar to that of shallow foundations. However, distinct differences may exist as a result of installation procedures. In addition, the settlement of a group is usually more than that of a single pile with comparable working load. Predictions of total settlement can be considerably improved by experience obtained from observations on a regional basis.

Cohesive Soils

Deep foundations in clay are more critically affected by the rate of loading of the pile. Significant pore pressures may be generated during the loading stage. The rate of pore pressure dissipation implicitly determines the analytical method chosen for an assessment of bearing

capacity. If significant pore pressures are expected to be induced due to the nature of the soil, an undrained, short-term ultimate load capacity should be computed. However, if pore pressures induced for normal rates of load application dissipate fairly rapidly, then an effective stress-drained analysis may be more appropriate.

For point capacity, an analysis based on drained conditions is performed similarly to the method developed for cohesionless soils. Otherwise, a semi-empirical bearing capacity factor is applied to the undrained cohesion of the soil at the pile base.

Several methods are available to determine the ultimate shaft capacity of piles in clay. Not all of the methods are universally applicable and best results are obtained with discrete applications of the models with respect to soil type and loading rate.

The alpha method is conventionally used to evaluate the ultimate shaft capacity in clay in undrained conditions. The shaft friction is a function of the undrained strength of the clay modified by the alpha factor. The factor is a ratio of the pile-soil adhesion and the undrained shear strength of the clay.

The beta method may be employed when it is determined that the rate of pore pressure dissipation is to be so rapid that for normal rates of load application, drained conditions will generally prevail in the soil near the pile shaft. The beta factor modifies the effective overburden pressure.

Impressive results have been obtained with the use of the lambda method. It has been extensively used to predict the shaft capacity of heavily loaded pipe piles for offshore structures. The lambda coefficient was found to be a function of pile penetration.

The alpha method is highly sensitive to measured variation in undrained shear strength, whereas variations of this parameter may be eclipsed by the effective overburden pressure explicit in the lambda method. The beta and lambda methods have been found to give better results for normally consolidated soils. The alpha and lambda methods provide reasonable correlations for overconsolidated soils.

In order to avoid introducing significant scale effects of pile diameter on the ultimate point resistance in stiff fissured clay, a reduction factor is also recommended.

A set of partial safety factors are proposed in order to more directly assess the influence of various factors on the design process. The overall safety factor is expressed as a function of the partial factors for soil resistance and load. The partial safety factors may be reassessed if an unsatisfactory probability of ultimate failure results.

Dynamic Analysis

The need to corroborate the load capacity of driven piles in the field has led historically to the development of a number of dynamic formulas. The rate of pile penetration afforded a means of relating driving resistance to

ultimate capacity. The fundamental assumption of all the formulas is that the dynamic resistance encountered during driving is equal to the load capacity of the pile under static loading. Most formulas are based on the principles of a kinetic energy converted to the work of penetrating the soil a set distance. Not all of the formulas are equally reliable and only experience on a regional basis can improve their predictability.

With the advent of the computer, it was possible to analyze the mechanics of pile driving using wave theory. The wave equation analysis examines the transmission of elastic waves along the length of the pile. The main objective in using the wave equation approach is to obtain a better relationship between ultimate pile load and pile set, as well as providing a means of assessing the driveability of a pile with a particular set of equipment. The approach also enables a rational analysis to be made of the stresses in the pile during driving. The hammer-pile-soil system is discretized and represented by a system of springs, dash-pots, and mass elements. The second order differential wave equation is expressed in finite difference form. The resulting equations are solved simultaneously for each element for each time interval considered. A number of parameters are required with varying degrees of importance to the results of the analysis. The reliability of the analysis is dependent on the quality of the hammer-pile-soil model used to represent the in-situ conditions.

A computer program PILE was also developed to integrate both the static and dynamic analyses of pile foundations. The static analysis consists of various methods previously discussed and the dynamic analysis is comprised of a set of dynamic formulas and the wave equation analysis. The reliability of the program was tested on published results of load tests, and on case histories of difficult driving conditions provided by IDOH.

Conclusions

A computational package PILE has been developed which enables the designer to analyze pile foundations statically and dynamically in an integrated and comprehensive manner. The analytical models for static analysis were chosen on the basis of extensive empirical corroboration and the facility with which the pertinent and required parameters may be obtained. For the dynamic analysis, pile driving formulas are provided to accomodate present design practice. In addition, by integrating the wave equation analysis into the package, it was possible to simplify and reduce the amount of input data previously required to successfully perform a wave equation analysis.

As a whole, the following conclusions may be stated concerning the computational package (PILE):

- (1) Estimates of ultimate load capacities have been shown to be of reasonable accuracy for some 50 cases, including a variety of soils and pile

types.

- (2) Results of the dynamic formulas can be predicted only within wide limits but can be improved on a regional basis.
- (3) The wave equation has proven to be a most versatile tool. A detailed analysis of case histories has demonstrated that the wave equation analysis affords a useful means of assessing the driveability of a pile with a particular set of equipment, and is the only means of rationally evaluating the stresses generated during driving. Finally, the analysis renders at least a qualitatively correct picture of the driving mechanism.
- (4) The computational package is unique in its comprehensive integration of static and dynamic pile analyses. This package is expected to be of considerable value to pile design in general, and to highway bridge designers in particular.

Recommendations for Further Study

The present study by no means exhausts the possibilities of improving the analysis and design of pile foundations. To complete the topic of this research, the following additional areas of detailed study are recommended:

- (1) A synthesis of the state-of-art for pile load tests.

A number of new techniques have been introduced fairly recently that offer a faster and more economical determination of ultimate capacity.

- (2) A synthesis of the state-of-art for dynamic measurements. Since the introduction of dynamic measurements, significant advances have been made in the quality of the measurements, and these are useful adjuncts to wave equation predictions.
- (3) Implementation and debugging of PILE program. Extensive use of the PILE program may dictate certain modifications in order to more closely serve the design and construction procedures of the IDOH and other highway agencies.
- (4) Additional correlations with IDOH case histories. Such comparisons will more clearly define the usefulness of the package in its present form.

In order to further enhance present knowledge of pile design and construction, the following topics are recommended as content for future research projects:

- (1) Improved dynamic analysis.
The capacity to make dynamic measurements and apply them to the design of bridge piles should be developed for the IDOH.
- (2) New load testing procedures.
Modern and rapid procedures are now available for load testing. Development of a suitable

version of these procedures for the IDOH would make it economically feasible to run far more of them. This would in turn have a very favorable impact on present practice.

(3) Pile group behavior.

Present analytical models are based on the mechanics of a single pile and extended to pile groups. More research is needed in the in-situ performance of groups.

(4) Point bearing in hard rock.

The failure mechanisms of piles founded on rock need to be more closely investigated.

LIST OF REFERENCES

LIST OF REFERENCES

- Agerschou, H. A. 1962. "Analysis of the Engineering News Pile Formula." J. Soil Mech. & Found. Des., ASCE, vol. 88, no. SM5 : 1 - 11.
- Al-Awkati, Z., Vesic, A. S. 1972. "Effect of Lateral Pressure on Penetration Resistance and Bearing Capacity of Piles." ASCE Annual Meeting, Houston, Texas.
- Al-Awkati, Z. 1975. "On Problems of Soil Bearing Capacity at Depth." Ph.D. dissertation, Duke Univ., Durham, N.C.
- Authier, J., Fellenius, B. H. 1982. "Wave Equation Analysis and Dynamic Monitoring of Pile Driving." Civil Engineering for Practicing and Design Engineers, vol. 2 : 1 - 30.
- Bazaraa, A. R. 1967. "Use of the Standard Penetration Test for Estimating Settlements of Shallow Foundations in Sand." Ph. D. Thesis, Univ. of Illinois, Urbana, Illinois.
- Bender, C. H., Lyons, C. G., Lowery, L. L. 1969. "Applications of Wave Equation Analysis to Offshore Pile Foundations. 1st Offshore Tech. Conf., Houston, OTC 1055.
- Bowles, J. E. 1982. Foundation Analysis and Design. New York : McGraw-Hill.
- Bozozuk, M. 1972. "Downdrag Measurements on a 160-ft Floating Pipe Test Pile in Marine Clay." Canadian Geot. J., vol. 9, no. 2, May : 127 - 136.
- Bredenberg, H. 1978. "Steel Piles to Rock." Byggmastaren, vol. 57, no. 6 : 17 - 20.
- Broms, B. B., Bredenberg, H. 1980. "Applications of Stress-Wave Theory on Pile Driving - A State-of-the-Art Report." Royal Inst. Tech., Stockholm, Sweden : 1 - 31.
- Buisson, M., Ahu, J., Habib, P. 1960. "Le frottement negatif." Annales de l'Institut Technique du Batiment et des Travaux Publics, vol. 13, no. 145, Jan. : 29 - 46.
- Burland, J. B. 1973. "Shaft Friction of Piles in Clay - A Simple Fundamental Approach." Ground Eng., vol. 6, no. 3,

May : 30 - 42.

Chellis, R. D. 1961. Pile Foundations. New York : McGraw-Hill.

Chen, D. D. S., Toto, J. V., Wong, I. H. 1979. "Field Evaluations of Hammer Efficiency and Pile Driving Criteria." Symp. Deep Foundations, ASCE, Oct. : 64 - 74.

Cognon, J. M. 1972. "Measure in situ du frottement negatif." Annales de l'Institut Technique du Batiment et des Travaux Publics, no. 293, May.

Coyle, H. M., Costello, R. R. 1981. "New Design Correlations for Piles in Sand." J. Geot. Eng. Div., ASCE, vol. 107, no. GT7 : 965 - 986.

Dayal, V., Allen, J. H. 1975. "The Effect of Penetration Rate on the Strength of Remolded Clay and Sand Samples." Can. Geot. J., vol. 12 : 336 - 348.

De Beer, E. E. 1971. "Methodes de Deduction de la Capacite Portante d'un Pieu a Partir de Resultats de Essais de Penetration." Annales de Travaux Publics de Belgique, vol. 72, nos. 4 - 6 : 1 - 142.

De Mello, V. F. B. 1969. "Foundation of Buildings on Clay." State of the Art Report, Proc. 7th Int. Conf. Soil Mech. & Found. Eng., Mexico City : 49 -136.

De Mello, V. F. B. 1981. "The Standard Penetration Test." Proc. 4th Panamer. Conf. Soil Mech. & Found. Eng., San Juan : 1 - 86.

Dennis, N. D., Olson, R. E. 1983. "Axial Capacity of Steel Pipe Piles in Sand." Geot. Practice in Offshore Eng., ASCE, April : 389 - 401.

Desai, C. S. 1974. "Numerical Design - Analysis for Piles in Sands." J. Geot. Eng. Divn., ASCE, vol. 100, no. GT6 : 613 - 635.

Desai, C. S., Christian, J. T. 1977. Numerical Methods in Geotechnical Engineering. New York : McGraw-Hill.

Ellison, R. D., D'Appolonia, E., Thiers, G. R. 1971. "Load-Deformation Mechanism for Bored Piles." J. Soil Mech. & Found. Eng., ASCE, vol. 97, no. SM4 : 661 - 678.

Emrich, W. J. 1970. "Performance Study of a Soil Sampler for Deep-Penetration Marine Borings." ASTM, STP 483 : 30 - 50.

Endo, M., Minou, A., Kawasaki, T., Shibata, T. 1969. "Negative Skin Friction Acting on Steel Pipe Pile in Clay." Proc. 7th Int. Conf. Soil Mech. & Found. Eng., vol 2., Mexico City : 85 - 92.

Fellenius, B. H., Broms, B. B. 1969. "Negative Skin Friction for Long Piles Driven in Clay." Proc. 7th Int. Conf. Mech. Found. Eng., vol. 2, Mexico City : 93 - 98.

Fellenius, B. H. 1980. "The Analysis of Results from Routine Pile Load Tests." Ground Eng., vol. 13, no. 6, Sept. : 19 - 31.

Flaate, K. S. 1964. "An Investigation of the Validity of Three Pile Driving Formulas in Cohesionless Material." Pub. no. 56, N. G. I., Oslo, Norway.

Forehand, P. W., Reese, J. L. 1963. "Pile Driving Analysis Using the Wave Equation." M. S. Thesis, Princeton Univ., Princeton, N. J.

Fox, E. N. 1932. "Stress Phenomena Occurring in Pile Driving." Engineering, vol. 134 : 263 - 265.

Goble, G. G., Cochran, D., Marcucci, F. 1982. "Foundation Design and Evaluation for Winnemucca Viaduct." Piling 82, Transp. Res. Rec. 884, Washington : 37 - 45.

Goble, G. G., Rausche, F. 1976. "Wave Equation Analysis of Pile Driving; WEAP Program." vol. 1 - Background, vol. 2 - User's Manual, vol. 3 - Program Documentation, U.S. Dept. Transp., FHWA, Off. Res. & Dev., vol. 76-14.1, 76-14.3, Washington : 389 pp.

Goble, G. G., Rausche, F., Likins, G. E. 1980. "The Analysis of Pile Driving. A State-of-the-Art", Proc. International Seminar on the Application of Stress Wave Theory on Piles, Stockholm: 131 - 162.

Goble, G. G., Rausche, F., Moses, F. 1970. "Dynamic Studies on the Bearing Capacity of Piles. Phase III, Report No. 48, vol. 1 and 2, Div. of Solid Mechanics, Structures, and Mechanical Design, Case Western Reserve University, Cleveland: 325pp.

Hagerty, D. J., Peck, R. B. 1971. "Heave and Lateral Movements Due to Pile Driving." J. Soil Mech. & Found. Des., ASCE, vol. 97, no. SM118 : 1513 - 1537.

Heerema, E. P. 1981. "Dynamic Point Resistance in Sand and in Clay for Driveability Analysis." Ground Eng., vol. 14, no. 6, Sept. : 30 - 46.

- Hirsch, T. J., Carr, L., Lowery, L. L. 1976. "Pile Driving Analysis. Wave Equation User's Manual. TTI Program." vol. 1 - Background, vol. 2 - Computer Program, vol. 3 - Program Documentation, U.S. Dept. Transp., FHWA, Off. Res. & Dev., Washington : 308 pp.
- Holloway, P. M., Clough, G. W., Vesic, A. S. 1975. "The Mechanics of Pile-Soil Interaction in Cohesionless Soils." U.S. Army Eng. Water. Exp. Stat., Contract Rep. S-75-5.
- Holloway, P. M., Clough, G. W., Vesic, A. S. 1978. "The Effect of Residual Driving Stresses on Pile Performance Under Axial Loads." Offshore Tech. Conf., OTC 3306.
- Housel, M. S. 1966. "Pile Load Capacity - Estimates and Test Results." J. Soil Mech. & Found. Des., ASCE, vol. 97, no. SM4: 1 - 29.
- Hunter, A. H., Davisson, M. T. 1969. "Measurements of Pile Load Transfer." ASTM Spec. Tech. Publ. 444 : 106 - 117.
- Issacs, D. V. 1931. "Reinforced Concrete Pile Formulae." Trans. Inst. Engrs., vol. 12, Austria : 313 - 323.
- Johannessen, I. J., Bjerrum, L. 1965. "Measurements of the Compression of a Steel Pile to Rock due to Settlement of the Surrounding Clay." Proc. 6th Int. Conf. Soil Mech. & Found. Eng., vol. II, Montreal : 261 - 264.
- Kahl, H., Muhs, H., Meyer, W. 1968. "Ermittlung der Grosse und des Verlaufs des Spitzendrucks bei Drucksondierungen in Ungleichforomigen Sand-Kies-germischen und in Kies", Mitteilungen DEGEBO, Berlin, West Germany, no. 21: 1 - 36.
- Kerisel, J. 1961. "Fondationes Profondes en Milieu Sableux." Proc. 5th Int. Conf. Soil Mech. & Found. Eng., vol. 2 : 78 - 83.
- Kézdi, A. 1975. "Pile Foundations". Chapter 19, in Fang and Winterkorn, Foundation Engineering Handbook, Van Nostrand.
- Kraft, L. M., Focht, J. A., Amerasinghe, S. F. 1981. "Friction Capacity of Piles Driven into Clay." J. Geot. Eng. Divn., ASCE, vol. 107, no. GT11, Nov. : 1521 - 1541.
- Ladanyi, B. 1963. "Expansion of a Cavity in a Saturated Clay Medium." J. Soil Mech. & Found. Des., ASCE, vol. 89, no. SM4 : 127 - 161.
- Lee, I. K. 1973. "Application of Finite Element Method in Geotechnical Engineering, Part I - Linear Analysis." Ch. 17 in Finite Element Techniques - A Short Course of Fundamentals and Applications, Univ. N.S.W., Australia.
- Leonards, G. A., Lovell, D. 1978. "Interpretation of Load Tests on High Capacity Driven Piles." ASTM Symposium on

Behavior of Deep Foundations. ASTM STP 670 : 388 - 415.

Litkouhi, S., Poskitt, T. J. 1980. "Damping Constants for Pile Driveability Calculations." *Geotechnique*, vol. 30., no.1 : 77 - 86.

Lowery, L. L., Hirsch, T. J., Edward, T. C., Coyle, H. M., Samson, C. H. 1969. "Use of the Wave Equation to Predict Soil Resistance on a Pile During Driving." 7th Int. Conf. Soil Mech. & Found. Eng., Mexico City.

Mansur, C. I., Hunter, A. H. 1970. "Pile Tests - Arkansas River Project." *J. Soil Mech. & Found. Des.*, ASCE, vol. 96, no. SM5, Sept: 1545 - 1587.

Marmon, R. H., Holton, D. H. 1982. "I-95/82 Interchange Test Program." *Found. Facts*, vol. 13 : 34 - 45.

Meigh, A. C., Nixon, I. K. 1961. "Comparison of in-situ Tests of Granular Soils." *Proc. 5th Int. Conf. Soil Mech. & Found. Eng.*, vol. 1, Paris : 499 - 502.

Melzer, K. J. 1968. "Sondenuntersuchungen in Sand." *Mitteilungen der Vereinigung der Grosskesselbetreiber*, Aachen, Germany, no. 43: 1 - 345.

Meyerhof, G. G. 1956. "Penetration Tests and Bearing Capacity of Cohesionless Soils." *Proc. ASCE*, vol. 82, no. SM1 : 1 - 19.

Meyerhof, G. G. 1959. "Compaction of Sands and Bearing Capacity of Piles." *J. Soil Mech. & Found. Des.*, ASCE, vol. 85, no. SM6 : 1 - 29.

Meyerhof, G. G. 1976. "Bearing Capacity and Settlement of Pile Foundations." *J. Geot. Eng. Divn.*, ASCE, vol. 102, no. GT3, March : 196 - 228.

Meyerhof, G. G. 1983. "Capacity of Sands with Reduction Factors." *J. Geot. Eng. Divn.*, ASCE, vol. 109, no. 6, June : 797 - 806.

Mohan, D., Jain, G. S. 1961. "Bearing Capacity of Piles in Expansive Clays." *Proc. 5th Int. Conf. Soil Mech. & Found. Eng.*, vol 2 : 117.

Morris, P. S., Barksdale, R. D. 1980. "A New Pile-Driving Formula." *Military Eng.*, vol. 74, no. 480 : 211 - 213.

Muhs, H., Weiss, K. 1971. "Untersuchung von Grenztragfähigkeit und Setzungsverhalten Flachgegründeter Einzelfundamente in Ungleichförmigen Nichtbindigen Boden", *Mitteilungen DEGEBO*, Berlin, West Germany, no. 26: 1 - 39.

- Norlund, R. L. 1963. "Bearing Capacity of Piles in Cohesionless Soils." J. Soil Mech. & Found. Des., ASCE, vol. 89, no. SM3 : 1 - 35.
- O'Neill, M. W., Hawkins, R. A., Mahar, L. J. 1982. "Load Transfer Mechanisms in Pile and Pile Groups." J. Geot. Eng. Divn., ASCE, vol. 108, no. GT12, Dec. : 1605 - 1623.
- Olsen, R. E., Flaate, K. S. 1967. "Pile-Driving Formulas for Friction Piles in Sands." J. Soil Mech. & Found. Des., ASCE, vol. 93, no. SM6 : 279 - 296.
- Parola, J. I. 1970. "Mechanics of Impact Pile Driving." Ph. D. dissertation, Univ. of Illinois, Urbana, Ill.
- Perloff, W. H. 1974. "Pressure Distribution and Settlement." Handbook of Foundation Engr., ed. H. Winterkorn & H. Y. Fang, Van Nostrand, New York : 148 - 196.
- Poulos, H. G., Davis, E. H. 1980. Pile Foundation Analysis and Design. New York : Wiley.
- Ramey, G. E., Hudgins, A. P. 1977. "Sensitivity and Accuracy of the Pile Wave Equation". Ground Eng., vol. 10, Oct.: 45 - 47.
- Rausche, F., Goble, G. G. 1978. "A Critical Examination of the Wave Equation." Trans. Res. Record, no. 665, Sept. : 209 - 213.
- Robinsky, E. I., Morrison, C. E. 1964. "Sand Displacement and Compaction Around Model Friction Piles." Can. Geot. J., vol. 1, no. 2 : 81.
- Samson, C. H., Hirsch, T. J., Lowery, L. L. 1963. "Complete Study of Dynamic Behavior of Piling." J. Struct. Divn., ASCE, vol. 89, no. ST4 : 413 - 449.
- Schultze, E., Melzer, K. J. 1965. "The Determination of the Density and the Modulus of Compressibility of Non-Cohesive Soils by Soundings." Proc. 6th Int. Conf. Soil Mech. & Found. Eng., vol. 1, Montreal : 354 - 358.
- Seed, A. B., Reese, L. C. 1957. "The Action of Soft Clay along Friction Piles." Trans. ASCE, vol. 122 : 731 - 754.
- Skempton, A. W. 1951. "The Bearing Capacity of Clays." Build. Res. Congress, London Inst. Civ. Engrs., div. I : 180 pp.
- Skempton, A. W. 1959. "Cast-In-Situ Bored Piles in London Clay." Geotechnique, vol. 9 : 158.

Smith, E. A. L. 1960. "Pile Driving Analysis by the Wave Equation." J. Soil Mech. & Found. Des., ASCE, vol. 86, no. SM4 : 35 - 61.

Soderberg, L. 1962. "Consolidation Theory Applied to Foundation Pile Time Effect." Geotechnique, vol. 12 : 217.

Sorensen, T., Hansen, B. 1957. "Pile Driving Formulas, An Investigation Based on Dimensional Considerations and a Statistical Analysis." Proc. 4th Int. Conf. Soil Mech. & Found. Eng., vol. 2 : 61 - 65.

Sowers, G. F., Martin, C. B., Wilson, L. L., Fausold, M. 1961. "The Bearing Capacity of Friction Pile Groups in Homogeneous Clay from Model Studies." Proc. 5th Int. Conf. Soil Mech. & Found. Eng., vol. 2 : 155 - 159.

Sutherland, H. B. 1963. "The Use of In-Situ Tests to Estimate the Allowable Bearing Pressure of Cohesionless Soils." Struct. Engr., vol. 41. no. 3, March : 85 - 92.

Tavenas, F. A. 1971. "Load Test Results on Friction Piles in Sand." Can. Geot. J., vol. 8 : 7 - 22.

Taylor, D. W. 1948. Fundamentals of Soil Mechanics, New York : Wiley.

Terzaghi, K., Peck, R. B. 1967. Soil Mechanics in Engineering Practice. New York : Wiley.

Thompson, C. D., Devata, M. 1980. "Evaluation of Ultimate Bearing Capacity of Different Piles, Proc. Int. Sem. Appl. Stress-Wave Theory on Piles, Stockholm : 163 - 195.

Thompson, C. D., Thompson, D. E., 1983. "Real and Apparent Relaxation of Driven Piles." Dynamic Measurements of Piles and Piers, ASCE.

Timoshenko, P. S., Goodier, J. 1951. Theory of Elasticity. New York : McGraw-Hill.

Tomlinson, M. J. 1957. "The Adhesion of Piles Driven in Clay Soils." Proc. 4th Int. Conf. Soil Mech. & Found. Eng., vol. 2 : 66 - 71.

Tomlinson, M. J. 1971. "Some Effects of Pile Driving on Skin Friction." Conf. on Behav. of Piles, Inst. Civ. Engrs., London : 59 - 66.

Vardé, O. A. et al. 1969. "Behavior of Large Tank Pile Foundations on Clay", Progress Report, Specialty Session, Fricción Negativa y Asentamientos de Cimentaciones Sobre Pilotes, 7th Int. Conf. Soil Mech. and Found. Eng., Mexico City.

Vesic, A. S. 1964. "Model Testing of Deep Foundations in Sand and Scaling Laws." Panel Discussion, Session II, Proc. N. Amer. Conf. Deep Found., vol. II, Mexico City : 525 - 533.

Vesic, A. S. 1967. "A Study of Bearing Capacity of Deep Foundations." Final Rep., Proj. B-189, School of Civil Eng., Georgia Inst. Tech., Atlanta, Ga.

Vesic, A. S. 1969. "Load Transfer, Lateral Loads and Group Action of Deep Foundation." Performance of Deep Foundations, ASTM Spec. Tech. Publ. No. 444 : 5 - 14.

Vesic, A. S. 1970. "Tests on Instrumented Piles, Ogeechee River Sites." J. Soil Mech. & Found. Des., ASCE, vol. 96, no. SM2, March : 561 - 584.

Vesic, A. S. 1972. "Expansion of Cavities in Infinite Soil Mass." J. Soil Mech. & Found. Eng., ASCE, vol 98, no. SM3, March : 265 - 290.

Vesic, A. S. 1973. "On Prediction Resistance and Bearing Capacity of Piles in Sand." 8th Int. Conf. Soil Mech. & Found. Eng., vol. 4-2, Moscow : 78 - 81.

Vesic, A. S. 1977. "Design of Pile Foundation." Trans. Res. Board, no. 42 : 1 - 68.

Vesic, A. S., Clough, G. W. 1968. "Behavior of Granular Materials Under High Stresses." J. Soil Mech. & Found. Des., ASCE, vol. 94, no. 3, May : 661 - 688.

Vijayvergiya, V. N., Focht, J. A. 1972. "A New Way to Predict the Capacity of Piles in Clay." 4th Annual Offshore Tech. Conf., vol.2, Houston : 865 - 874.

Walker, L. K., Darvall, P. L. P. 1973. "Dragdown on Coated and Uncoated Piles." Proc. 8th Int. Conf. Soil Mech. & Found. Eng., vol. 2.1, Moscow : 257 - 262.

Whitaker, T. 1970. The Design of Piled Foundations. Oxford : Pergamon.

Wright, S. J., Reese, L. C. 1977. "Limit State Design of Drilled Shafts." Res. Report, Cntr. Highway Res., Univ. Texas at Austin : 132 pp.

Wright, S. J., Reese, L. C. 1979. "Design of Large Diameter Bored Piles." Ground Eng., vol. 12, no. 8, Nov. : 17 - 51.

APPENDICES

Appendix A - PILE Program

A complete listing of PILE is subsequently provided. Tapes of the PILE program are available from Purdue University. The program was written in Fortran language. The following specifications are necessary for normal operations on the CDC 6500:

core used = 056000b octal words

time used = 026.080 cpu seconds

nl = 127400 words

cx = 33.082 sec

```
program main (tape5,tape6,tape8,plot)
```

```
c    pile capacity
```

```
integer shear,soil(100),consl(100),instl
real len,layer,nq(100),ks(100),nc(100),ko(100)
integer wav
dimension phi(100),q1(100),xl(100),qp(10,100),depth(10,100),
1fl(100),fs(10,100),n(100),qpcap(10,100),fscap(10,100),qult(10,100)
2,cu(100),gam(100),strs(10,100),zc(100),xc(100),del(100),
3bet(10,100),alpha(100),ocr(100),bp(10,100),atap(10,100),ncorr(100)
```

```
c    pile and soil characteristics
```

```
ir=5
read(ir,3) nstrat,b,apnt,ashft,len,qprmt,shear,prcnt,instl,neqn,
lwav
3 format(i4,3f8.6,2f8.2,a4,f4.2,a6,2x,2i4)
write(6,100)
100 format(///lx,72hpurdue university                ultimat
le static pile capacity)
write(6,106)
106 format(///lx,17hpile - soil model)
write(6,111)
111 format(///lx,9hsoil data)
if(shear.eq.'dc')write(6,112)
if(shear.eq.'uu')write(6,113)
112 format(/3x,46hshear strength : drained - long term condition)
113 format(/3x,49hshear strength : undrained - short term condition)
write(6,114)nstrat
114 format(/3x,19hnumber of strata = ,i2)

dol0i=1,nstrat
read(ir,1)soil(i),xl(i),n(i),ncorr(i),phi(i),nq(i),cu(i),gam(i),
1zc(i),ks(i),del(i),alpha(i),nc(i),consl(i),ocr(i)
1 format(a4,f8.2,2i4,f4.2,f4.1,f4.2,f4.1,5f4.2,a4,f4.2)

phi(i)=phi(i)*3.1415927/180.
del(i)=del(i)*3.1415927/180.
gam(i)=gam(i)/2000.
if(cu(i).eq.0.0)go to 7
if(consl(i).eq.'oc')go to 4
ko(i)=(1.0-sin(phi(i)))
go to 7
4 if(ocr(i).eq.0.0)go to 6
ko(i)=(1.0-sin(phi(i)))*sqrt(ocr(i))
go to 7
6 ko(i)=1.5*(1.0-sin(phi(i)))
7 xc(i)=zc(i)*b
if(i.eq.1)go to 2
if(xc(i).ne.xc(i-1))go to 2
xc(i)=xc(i)-xl(i-1)
2 if(soil(i).eq.'arn')qcfac=4.0
```

```

      if(soil(i).eq.'lim')qcfac=3.0
      if(soil(i).eq.'arg')qcfac=2.0
      if(ncorr(i).eq.0)ncorr(i)=n(i)
      q1(i)=qcfac*ncorr(i)
      if(n(i).eq.0)q1(i)=cu(i)*nc(i)
      if(cu(i).gt.0.0) go to 9
      fl(i)=qcfac*0.005*n(i)
      if(prcnt.ge.1.0)fl(i)=1.5*fl(i)
      go to 10
9    if(shear.eq.'uu')go to 5
      fl(i)=cu(i)
      if(prcnt.ge.1.0)fl(i)=1.5*fl(i)
      go to 10
5    fl(i)=alpha(i)*cu(i)
10   continue

      layer=0.0
      sstrs=0.0
      qqp=0.0
      ffs=0.0
      ffscap=0.0
      bbp=b
      do 12 i=1,nstrat

      write(6,115)
115  format(/129h-----)
      1-----
      2-----)
      write(6,116)i,x1(i),phi(i)*180./3.1415927,cu(i)
116  format(/12,1x,20hstratum thickness = ,f6.2,1x,3hft.,10x,30hangle o
      1f internal friction = ,f6.2,1x,4hdeg.,3x,30hundraind shear stren
      2gth = ,f5.2,1x,4htsf.)
      if(gam(i).eq.0.0)go to 118
      write(6,117)n(i),ncorr(i),nq(i),nc(i)
117  format(3x,20haverage n value = ,i6,1x,5hb/ft.,1x,1h(,i3,1h),2x,
      130hbearing capacity factor, nq = ,f6.2,8x,30hbearing capacity fact
      2or, nc = ,f5.2)
      write(6,119)gam(i)*2000.,ks(i),ko(i)
119  format(3x,20hunit weight = ,f6.2,1x,10hlbs/cu.ft.,3x,30hcoef
      1. of earth pressure, ks = ,f6.2,8x,30hcoef. of earth pressure, ko
      2= ,f5.2)
      write(6,120)zc(i),del(i)*180./3.1415927,alpha(i)
120  format(3x,20hcritical depth r = ,f6.2,1x, 3x,10x,30hangle of sh
      1aft friction = ,f6.2,1x,4hdeg.,3x,30hadhesion coefficient
      2 = ,f5.2)
      go to 122
118  write(6,121)n(i),ncorr(i)
121  format(3x,20haverage n value = ,i6,1x,5hb/ft.,1x,1h(,i3,1h))
122  if(i.eq.nstrat)write(6,123)
123  format(/129h-----)
      1-----
      2-----)

```



```

nl=x1(i)
do 13 j=1,nl
x=j
depth(i,j)=layer+x
if(gam(i).eq.0.0) go to 8

```

c qp and fs computations in terms of effective stresses

```

36 if(i.eq.nstrat)go to 37
   if(q1(i).gt.q1(i+1))go to 34
37 strs(i,j)=sstrs+gam(i)*x
   if((cu(i).ne.0.0).and.(shear.eq.'uu'))go to 61

69 qp(i,j)= strs(i,j)*nq(i)
   go to 65
61 qp(i,j)=cu(i)*nc(i)
65 if(xc(i).le.0.0)go to 38
   if(x.gt.xc(i))qp(i,j)=qp(i,j-1)
   go to 35
38 qp(i,j)=qqp
35 if(qp(i,j).ge.q1(i))qp(i,j)=q1(i)
   if(j.eq.1)go to 32
   if(qp(i,j).eq.qp(i,j-1))q1(i)=qp(i,j)
32 go to 40
34 strs(i,j)=sstrs+gam(i)*x
   if((cu(i).ne.0.0).and.(shear.eq.'uu'))go to 62

70 qp(i,j)=strs(i,j)*nq(i)
   go to 66
62 qp(i,j)=cu(i)*nc(i)
66 if(xc(i).le.0.0)go to 39
   if(x.gt.xc(i))qp(i,j)=qp(i,j-1)
   go to 33
39 qp(i,j)=qqp
33 if(qp(i,j).ge.q1(i))qp(i,j)=q1(i)
   if(j.eq.1)go to 31
   if(qp(i,j).eq.qp(i,j-1))q1(i)=qp(i,j)
31 h=10.0*b
   hh=x1(i)-h
   if(x.lt.hh) go to 40
   if(x1(i).le.h)go to 40
   qp(i,j)=qp(i,j)-abs(q1(i)-q1(i+1))*(x-hh)/h

40 if(cu(i).gt.0.0)go to 50
   if(i.eq.nstrat)go to 47
   if(f1(i).gt.f1(i+1))go to 44
47 fs(i,j)= strs(i,j)*ks(i)*tan(del(i))
   if(xc(i).le.0.0)go to 48
   if(x.gt.xc(i))fs(i,j)=fs(i,j-1)
   go to 45
48 fs(i,j)=ffs
45 if(fs(i,j).ge.f1(i))fs(i,j)=f1(i)
   if(j.eq.1)go to 42

```

```

      if(fs(i,j).eq.fs(i,j-1))f1(i)=fs(i,j)
42  go to 15
44  fs(i,j)=      strs(i,j)*ks(i)*tan(del(i))
      if(xc(i).le.0.0)go to 49
      if(x.gt.xc(i))fs(i,j)=fs(i,j-1)
      go to 43
49  fs(i,j)=ffs
43  if(fs(i,j).ge.f1(i))fs(i,j)=f1(i)
      if(j.eq.1)go to 41
      if(fs(i,j).eq.fs(i,j-1))f1(i)=fs(i,j)
41  h=10.0*b
      hh=x1(i)-h
      if(x.lt.hh)go to 15
      if(x1(i).le.h)go to 15
      fs(i,j)=fs(i,j)-abs(f1(i)-f1(i+1))*(x-hh)/h
      go to 15

```

c fs computations for cohesive layer

```

50  bet(i,j)=ko(i)*tan(phi(i))
      if(depth(i,j).lt.50.0)go to 56
      bet(i,j)=0.15+(200.0-depth(i,j))*(bet(i,j)-0.15)/150.0
      if(depth(i,j).gt.200.0)bet(i,j)=0.15
56  if(i.eq.nstrat)go to 57
      if(f1(i).gt.f1(i+1)) go to 54
57  if(shear.eq.'uu')go to 63
      fs(i,j)=strs(i,j)*bet(i,j)
      go to 67
63  fs(i,j)=alpha(i)*cu(i)
67  if(xc(i).le.0.0)go to 58
      if(x.gt.xc(i))fs(i,j)=fs(i,j-1)
      go to 55
58  fs(i,j)=ffs
55  if(fs(i,j).ge.f1(i))fs(i,j)=f1(i)
      if(j.eq.1)go to 52
      if(fs(i,j).eq.fs(i,j-1))f1(i)=fs(i,j)
52  go to 15
54  if(shear.eq.'uu')go to 64
      fs(i,j)=strs(i,j)*bet(i,j)
      go to 68
64  fs(i,j)=alpha(i)*cu(i)
68  if(xc(i).le.0.0)go to 59
      if(x.gt.xc(i))fs(i,j)=fs(i,j-1)
      go to 53
59  fs(i,j)=ffs
53  if(fs(i,j).ge.f1(i))fs(i,j)=f1(i)
      if(j.eq.1)go to 51
      if(fs(i,j).eq.fs(i,j-1))f1(i)=fs(i,j)
51  h=10.0*b
      hh=x1(i)-h
      if(x.lt.hh)go to 15
      if(x1(i).le.h)go to 15
      fs(i,j)=fs(i,j)-abs(f1(i)-f1(i+1))*(x-hh)/h

```

go to 15

c qp computations

```

8  if(i.eq.nstrat)goto17
   if(q1(i).gt.q1(i+1))goto14
17  qp(1,j)=qqp+q1(i)*x/(10.0*b)
   if(i.eq.1)go to 19
   qp(1,j)=qqp+abs(q1(i)-q1(i-1))*x/(10.0*b)
19  if(qp(1,j).ge.q1(i))qp(1,j)=q1(i)
   go to 20
14  qp(1,j)=qqp+q1(i)*x/(10.0*b)
   if(i.eq.1)go to 11
   qp(1,j)=qqp+abs(q1(i)-q1(i-1))*x/(10.0*b)
11  if(qp(1,j).ge.q1(i))qp(1,j)=q1(i)
   h=10.0*b
   hh=x1(i)-h
   if(x.lt.hh)goto20
   if(x1(i).le.h)go to 20
   qp(1,j)=qp(1,j)-abs(q1(i)-q1(i+1))*(x-hh)/h

```

c fs computations

```

20  if(i.eq.nstrat)goto27
   if(f1(i).gt.f1(i+1))goto24
27  fs(1,j)=ffs+f1(i)*x/(10.0*b)
   if(i.eq.1)go to 28
   fs(1,j)=ffs+abs(f1(i)-f1(i-1))*x/(10.0*b)
28  if(fs(1,j).ge.f1(i))fs(1,j)=f1(i)
   go to 15
24  fs(1,j)=ffs+f1(i)*x/(10.0*b)
   if(i.eq.1)go to 25
   fs(1,j)=ffs+abs(f1(i)-f1(i-1))*x/(10.0*b)
25  if(fs(1,j).ge.f1(i))fs(1,j)=f1(i)
   h=10.0*b
   hh=x1(i)-h
   if(x.lt.hh)goto15
   if(x1(i).le.h)go to 15
   fs(1,j)=fs(1,j)-abs(f1(i)-f1(i+1))*(x-hh)/h

```

c point and shaft capacities

```

15  qpcap(1,j)=qp(1,j)*apnt
   if(prcnt.eq.0.0)go to 93
   bp(1,j)=bbp+x*prcnt/100.
   atap(1,1)=(bbp+bp(1,j))*3.1415927/2.0
   if(j.eq.1)go to 92
   atap(1,j)=(bp(1,j)+bp(1,j-1))*3.1415927/2.0
92  ashft=atap(1,j)

93  fscap(1,1)=ffscap+ashft*(fs(1,j))/2.0
   if(j.eq.1)go to 21
   fscap(1,j)=ffscap+ashft*(fs(1,j)+fs(1,j-1))/2.0

```

```

21  ffscap=fscap(1,j)
    if(inst1.eq.'bored')qpcap(1,j)=qpcap(1,j)/3.0
    if(inst1.eq.'bulb')qpcap(1,j)=qpcap(1,j)*2.0
    if(inst1.eq.'bored')fscap(1,j)=fscap(1,j)/2.0
    if(inst1.eq.'h pile')fscap(1,j)=fscap(1,j)/2.0
    qult(1,j)=qpcap(1,j)+fscap(1,j)
    if(qprmt.eq.0.0) go to 18
    if(qult(1,j).ge.qprmt) go to 101
18  if(len.eq.0.0)gotol3
    if(depth(1,j).ge.len)gotol01
13  continue
    sstrs=strs(1,n1)
    qqp=qpcap(1,n1)
    ffs=fscap(1,n1)
    layer=depth(1,n1)
    bbp=bp(1,n1)

12  continue
101 write(6,107)
107 format(/1x,9hpile data)
    write(6,108)inst1
108 format(/3x,25hmethod of installation: ,a6)
    write(6,109)b,apnt,ashft
109 format(/3x,20hpoint width          = ,f10.6,1x,3hft.,6x,24hpoint area
1      = ,f10.6,1x,6hsq.ft.,3x,23hshaft area          = ,f10
2.6,1x,6hsq.ft.)
    if(prcnt.ne.0.0)write(6,110)prcnt
110 format(/3x,16hpercent taper = ,f5.2,1x,1h_)
    write(6,103)qpcap(1,j)
    write(6,104)fscap(1,j)
    write(6,105)qult(1,j),depth(1,j)
103 format(////1x,31hultimate point capacity      = ,f10.2,1x,4htons)
104 format(/1x,31hultimate shaft capacity          = ,f10.2,1x,4htons)
105 format(/1x,31hultimate total pile capacity = ,f10.2,1x,4htons,25x,
129htotal depth of penetration = ,f10.2,1x,4hfeet)

    if(neqn.eq.0)go to 201
    call dynmc(qult(1,j),neqn,depth(1,j),apnt,1r)

201 if(wav.eq.0)go to 200
    call segs(fscap,qult(1,j),qpcap(1,j),depth(1,j),ashft,1r)

200 stop
    end

    subroutine dynmc(ru,neqn,ll,aa,1r)

    real n(6),l(6),ll
    integer eqn(6)
    dimension w(6),h(6),c(6),ef(6),wp(6),a(6),ep(6)
1,cl(6),c2(6),c3(6),set(6),blow(6),fs(6)

    do 100 k=1,neqn

```

```

      1(k)=11*12.
      a(k)=aa*144.
100 continue

      write(6,80)
80 format(1hl,///75h                                     dynamic
      1 formulas for pile driving)
      do 10 k=1,neqn
      read (ir,11) eqn(k),w(k),h(k),c(k),ef(k),n(k),wp(k),c1(k),c3(k),
      lep(k),fs(k)
11 format(a6,2x,f8.0,4f4.2,f8.0,f4.3,f4.2,f8.0,f4.2)
10 continue
      ru=ru*2000.
      do 12 k=1,neqn
      if(eqn(k).eq.'enr')go to 20
      if(eqn(k).eq.'hiley')go to 21
      if(eqn(k).eq.'janbu')go to 22
      if(eqn(k).eq.'danish')go to 23
      if(eqn(k).eq.'gates')go to 24

20 set(k)=(w(k)*h(k)/ru*fs(k))-c(k)
      write(6,31)
31 format(///1x,27hformula : engineering news)
      write(6,32)w(k),c(k)
32 format(//31x,22hweight of hammer, w = ,f9.0,4x,4hlbs.,12x,30hfact o
      lr for energy losses, c = ,f11.2)
      write(6,33)h(k)
33 format(33x,20hdrop of hammer, h = ,f11.2,2x,3hin.)
      go to 13

21 c2(k)=ru*fs(k)*1(k)/(a(k)*ep(k))
      eiv=(w(k)+wp(k)*(n(k)**2))/(w(k)+wp(k))
      set(k)=(ef(k)*w(k)*h(k))*eiv/ru*fs(k)-0.5*(c1(k)+c2(k)+c3(k))
      write(6,41)
41 format(///1x,16hformula : hiley)
      write(6,42)w(k),wp(k)
42 format(//31x,22hweight of hammer, w = ,f9.0,4x,4hlbs.,21x,21hweigh
      lt of pile, wp = ,f9.0,4x,4hlbs.)
      write(6,43)h(k),c1(k)
43 format(33x,20hdrop of hammer, h = ,f11.2,2x,3hin.,9x,34helastic co
      mpimpression of head, c1 = ,f11.2,2x,3hin.)
      write(6,44)ef(k),c2(k)
44 format(18x,35hefficiency factor for hammer, ef = ,f11.2,14x,34hela
      stic compression of pile, c2 = ,f11.2,2x,3hin.)
      write(6,45)n(k),c3(k)
45 format(19x,34hefficiency factor for impact, n = ,f11.2,12x,36helas
      tic compression of ground, c3 = ,f11.2,2x,3hin.)
      go to 13

22 cd=0.75+0.15*wp(k)/w(k)
      set(k)=(w(k)*h(k))/(2.0*ru*fs(k)*cd)-(1(k)*ru)/(2.0*a(k)*ep(k))
      write(6,51)
51 format(///1x,16hformula : janbu)

```



```

    write(6,52)w(k),wp(k)
52 format(/31x,22hweight of hammer, w = ,f9.0,4x,4hlbs.,21x,21hweight
    of pile, wp = ,f9.0,4x,4hlbs.)
    write(6,53)h(k),a(k)
53 format(33x,20hdrop of hammer, h = ,f11.2,2x,3hin.,16x,27hcross-sec
    tion of pile, a = ,f11.2,2x,6hsq.in.)
    write(6,54)ep(k),l(k)
54 format(17x,36hmodulus of elasticity of pile, ep = ,f9.0,4x,4hpsi.,
    125x,17hpile length, l = ,f11.2,2x,3hin.)
    go to 13

23 set(k)=ef(k)*w(k)*h(k)/ru*fs(k)-sqrt(ef(k)*w(k)*h(k)*l(k)/
    1(2.0*a(k)*ep(k)))
    write(6,61)
61 format(/31x,17hformula : danish)
    write(6,62)w(k),a(k)
62 format(/31x,22hweight of hammer, w = ,f9.0,4x,4hlbs.,15x,27hcross
    l-section of pile, a = ,f11.2,2x,6hsq.in.)
    write(6,63)h(k),l(k)
63 format(33x,20hdrop of hammer, h = ,f11.2,2x,3hin.,26x,17hpile leng
    th, l = ,f11.2,2x,3hin.)
    write(6,64)ef(k),ep(k)
64 format(18x,35hefficiency factor for hammer, ef = ,f11.2,12x,36hmod
    ulus of elasticity of pile, ep = ,f9.0,4x,4hpsi.)
    go to 13

24 set(k)=10.0**((1.0-(ru*fs(k)/2000.)/(3.0/7.0*sqrt(ef(k)*w(k)*
    lh(k)/12.0)))
    write(6,71)
71 format(/31x,16hformula : gates)
    write(6,72)w(k),ef(k)
72 format(/31x,22hweight of hammer, w = ,f9.0,4x,4hlbs.,7x,35heffici
    lency factor for hammer, ef = ,f11.2)
    write(6,73)h(k)
73 format(33x,20hdrop of hammer, h = ,f11.2,2x,3hin.)

13 blow(k)=1.0/set(k)*12.0
    write(6,74)ru/2000.*fs(k)
74 format(/31x,34hultimate dynamic resistance, ru = ,f6.0,4x,4htons)
    write(6,75)set(k)
75 format(1x,34hpermanent set, s = ,f8.2,2x,3hin.)
    write(6,76)blow(k)
76 format(1x,34hdriving resistance, 1.0/s = ,f6.0,4x,9hblows/ft
    l.)
12 continue
    ru=ru/2000.
    return
end

```

```

subroutine segs(fscap,qu,qp,d,ashft,ir)

```

```

dimension fscap(10,100),fsseg(10,100),nstrat(100),su(100),

```



```

    la(100),stp(100),pm(100),alph(100),qs(100),sj(100),l(100),
    2ssu(100)
    read(ir,10)nseg,nlayer,atp,e,w
10  format(2i4,f8.3,f8.0,f8.2)
    do 11 j=1,nseg+1
        if(j.eq.nseg+1)go to 13
        read(ir,12)nelm,nstrat(j),alph(j),a(j),qs(j),sj(j)
12  format(2i4,f8.2,f8.3,2f4.2)
        if(nstrat(j).eq.0)nstrat(j)=nstrat(j-1)
        if(alph(j).eq.0.0)alph(j)=alph(j-1)
        if(a(j).eq.0.0)a(j)=a(j-1)
        if(qs(j).eq.0.0)qs(j)=qs(j-1)
        if(sj(j).eq.0.0)sj(j)=sj(j-1)
        stp(j)=(a(j)*e)/(alph(j)*12.)
        pm(j)=a(j)*alph(j)*w/144000.
        l(j)=alph(j)
        go to 11
13  read(ir,14)nelm,qs(j),sj(j)
14  format(i4,20x,2f4.2)
        if(qs(j).eq.0.0)qs(j)=qs(j-1)
        if(sj(j).eq.0.0)sj(j)=sj(j-1)
11  continue

```

```

    ii=0
    ffseg=0.0
    do 20 i=1,nlayer
        jj=0
        do 21 j=1,nseg
            if(i.ne.1)go to 27
            jj=jj+1(j)
            if(nstrat(j).ne.i)go to 20
            go to 28
27  jj=jj+1(ii+1)
            if(nstrat(ii+1).ne.i)go to 20
28  fsseg(i,j)=fscap(i,jj)-ffseg
            ffseg=fscap(i,jj)
            ii=1+ii
            su(ii)=fsseg(i,j)/qu
21  continue
20  continue
        su(ii+1)=qp/qu
        call weap(stp,pm,qu,alph,qs,sj,su,nseg,d,atp,e,w)
        return
    end

```

```

    subroutine weap(pstp,ppm,prult,palph,pqs,psj,psu,nseg,pxpt,pap,
    lpep,prp)

```

```

c  ***** w e a p wave equation analysis for piles *****
c  this program was developed for the federal highway administration
c  by goble & associates, inc.
c
c  1980 version
c

```

```

c   for information regarding the program coding contact
c                                     goble & associates, inc.
c                                     4423 emery industrial parkway
c                                     warrensville heights, ohio 44128
c                                     (216)-831-6131

```

```

c   note that this program utilizes a data file which contains
c   proprietary information. this data, therefore, may not be
c   duplicated in any form without the expressed written permission
c   of either the fhwa or goble consulting engineers

```

```

c   common vinh,dinh,aoh,voh,doh,anh,vnh,dnh,sth,hm,am,sta,ma,esoil,
1  aop,anp,dop,dnp,vop,vnp,stp,pm,cdp,res,reso,sj,sok,qs,su,
2  splice,depib,aram,ecus,ecap,ept,eant,vfall,vcham,xpt,sjjj,
3  emp,rho,dth,dtp,n,m,jrat,iult,rult,soilm,dsacp,
4  fpto,iw,pcom,tdel,iout,out,expp,temi,temo,inp,stroko,strokn,vo,po
5  ,dfin,jdout,jpmax,ex,icol,fintmx,temax,result,ismith,iter
6  ,ityph,ipreig,vfallm,strmax,dcyl,pbar,area,jex
   common/dot/ibuf(2000),ibufo
   dimension vinh(10),dinh(10),aoh(10),voh(10),doh(10),anh(10),
1  vnh(10),dnh(10),sth(10),hm(10),vnp(99),
2  aop(99),anp(99),dop(99),dnp(99),vop(99),stp(99),pm(99),cdp(99),
3  res(99),reso(99),sj(99),sok(99),qs(99),su(99),splice(99),
4  dsacp(3),out(3200),inp(13),ex(600),result(100),jex(600)
5  ,am(10),sta(10),iseg(15),jselec(13),alph(99),title(10)
6  ,pbar(20),stinh(4,10),stinp(4,99),area(99),strar(10)
7  ,iex(40),pstp(99),ppm(99),palph(99),pqs(99),psj(99),psu(99)
   ibufo = -1
9877 continue
   strmax = 0.0
   call iptn(jselec,iseg,ymax,alph,title,iostr,stm,ioute,
1  pstp,ppm,prult,palph,pqs,psj,psu,nseg,pxpt,pap,pep,prp)
   if (iw.gt.-100) iwr = iw
   if (iw.lt.-100) go to 9876

c   iptn reads all input information and sets up hammer-pile-soil-
c   model
c   note iptn automatically calls for the right hammer and sets ityph
c   for air steam hammers stroke and vfall are set
c
c   initialize for new problem

   nl = n+1
   pcomr = pcom
   iout = iout
   if (iout.gt.9) iout = iout-10
   if (iout.gt.19) iout = iout-20
   ifuel = 0
   iwt = 0
   dstrok = 0.0
   inow = 0

```

```

      epsstr=0.05
c      maximum quake at bottom element
      q = qs(n)
      if (qs(n1).gt.q) q = qs(n1)
      coli=(xpt/sqrt(emp/rho))*1000.
      nult=iult
      if (nult.eq.0) nult = 1
      iult = 1
      dult=rult
      if(ityph.eq.3) go to 502
      if (pbar(1).lt.10.e-5) pbar(1) = aram
      if (ityph.ne.2) go to 502
      epsstr = 0.03
c      for closed end hammers, determines vfall for uplift condition
      call down(strmax,vfallm,pbar,hm,depib,m,iw,iout,downt)
      if (stroko.gt.0.1) go to 503
      vfall = vfallm
      stroko = strmax
      go to 502
c      if stroke is assigned find corresponding vfall
503 call down(stroko,vfall,pbar,hm,depib,m,iw,iout,downt)
502 continue
501 istr=0
c      initialize for new rult
      ifa = 0
      rtons = rult/2.0
      rtoe = su(n1)*0.5
      if(ioute.ne.1)write (6,9)rtons,rtoe
9      format (1h1,30x,5hrult=,f7.1,9h, at toe=,f7.1,5h tons/31x,33(1h-)/
1)
      nml=n-1
      dtr = 1.0
      dtrm = 1.0
      do 200 i=1,nml
c      check damping
      sdj=sj(i)
      if (ismith.eq.1) sdj=su(i)*sj(i)
      if (sdj.gt.10.e-8) dtr = pm(i)/sdj
      if (dtr.lt.dtrm) dtrm = dtr
200 continue
      sdj = sj(n) + sj(n+1)
      if (ismith.eq.1) sdj=sj(n)*su(n)+sj(n1)*su(n1)
      if (sdj.gt.10.e-8) dtr = pm(n)/sdj
c
      if (dtr.lt.dtrm) dtrm = dtr
      dtrm = dtrm*0.8
      if (dtrm.ge.dtp) go to 711
      if (dtrm.lt.0.00001) write (6,21) dtrm
      if (dtrm.lt.0.00001) go to 401
      dtp = dtrm
      dth = dtp/float(jrat)
      dtrm = dtrm*1000
      write (6,18) dtrm

```

```

711 continue
21  format (51h damping gtr critical, t.i. required lss .01 msec.(,e10
    1.5,1h))
18  format (43h large damping requires new time increment=,f7.4,
    1 6h msec.)
500  continue
    jmax = 6
    if (icol.gt.0) jmax = 13
    if (jmax.gt.n) jmax = n
    if (ityph.lt.3) go to 650
c   perform air/stm hammer analysis and go to output (out2)
    call airstm
    iblow=-1
    dfin = ((dfin/12.0)+qs(n)-q)*12.0
    if (abs(dfin).gt.0.0) iblow=12.0/dfin+0.5
    go to 900
650  continue
c   begin analysis for all diesel hammers
    vfr = vfall
    call startc
    tstart = fintmx
    m2=m+2
c   store initial values from start analysis (may be reused)
    stinh(1,10)=vo
    stinh(2,10)=fpto
    stinh(3,10)=float(ipreig)+0.1
    stinh(4,10)=po
    do 660 i=1,m2
        stinh(1,i)=aoh(i)
        stinh(2,i)=vinh(i)
660    stinh(3,i)=dinh(i)
        do 661 i=1,n1
            stinp(1,i)=aop(i)
            stinp(2,i)=vop(i)
            stinp(3,i)=dop(i)
661    stinp(4,i)=reso(i)
        if(ipreig.gt.-100)go to 662
        iblow=0
        do 668 i=2,10
            iad=i*10
668    result(iult+iad)=0.0
        vfall=vfr
        i=iult
        go to 400
662  continue
c   for non-starting hammers (ipreig=-100) proceed with next rult at
c   400; all final results set to zero.
    call diesel
    tdies = fintmx
c   determine stroke by analyzing upward motion
    call up(pbar,dnh,vnh,dnp,vnp,hm,pm,m,n,dcyl,dfin,vfall,vo,po,
    1 vfallm,strokn,depib,aram,expp,iwt,iw,iout,psi,tup)
c   no penetration and no increase of stroke means quit (300)

```



```

    if (dfin.lt.0.0.and.strokn.lt.stroko*1.025) go to 300
    if (iostr.lt.0) go to 750
c    reduce fuel for uplift (901)
    if (iwt.eq.1.and.ityph.eq.2.and.iostr.ne.1) go to 901
c    no uplift but previous uplift means fuel sufficiently reduced(300)
    if (ifuel.gt.0.and.iostr.ne.1) go to 300
    if (ityph.eq.1.and.strokn.gt.strm) write (6,20)
20  format (35h *** caution ram might blow out ***)
    if (ityph.eq.1.and.strokn.gt.strm) strokn=strm
c    determine stroke difference,store new stroke
    dostr = dstrok
    dstrok = abs(strokn-stroko)/stroko
    istr = istr + 1
    strar(istr) = stroko
    if (iostr.eq.1.or.dstrok.lt.epsstr) go to 300
    stroko = strokn
    if (istr.gt.4.and.dstrok.gt.(float(20-istr))*0.05*dostr) go to 301
    if (istr.gt.6) go to 301
    if (ityph.gt.1.or.istr.lt.2.or.pbar(20).gt.0.0) go to 675
    if (strar(istr).gt.strar(istr-1).and.strokn.gt.strar(istr)) go to
* 675
    if (strar(istr).lt.strar(istr-1).and.strokn.lt.strar(istr)) go to
* 675
c    for a new stroke which is between previous values
    strokn = 0.5*(strar(istr)+strokn)
    vfall = sqrt((strokn-depib)*64.34)
    stroko = strokn
c    at 650 a new complete analysis starts
    go to 650
c    change fuel setting
750  istr = istr + 1
    if (iwt.eq.1) strokn = strmax
    dstrf = (strokn-stroko)/stroko
    if (iwt.eq.1) dstrf = dstrf + (vfall-vfallm)/vfallm
    if (abs(dstrf).gt..2) dstrf=.2*dstrf/abs(dstrf)
    strar(istr) = strokn
    strokn = stroko
    if (abs(dstrf).lt.epsstr) go to 300
    if (istr.gt.4) go to 663
76  format (21h fuel setting changed,f6.3,7h perct.)
    if (iwt.eq.1.and.dstrf.lt.0.1.and.ityph.eq.2) dstrf=dstrf + 0.1
    pcom = pcom*(1.0-dstrf)
    if (iout.lt.0) write (6,76) dstrf
    vfall = vfr
    go to 761
675  continue
    if (dstrok.gt.2.0*epsstr) go to 650
c    initial values are reused (at 662)
    do 666 i=1,m2
    vinh(i)=stinh(2,i)
    if (i.le.m) vinh(i)=(vfall/vfr)*vinh(i)
    aoh(i)=stinh(1,i)
666  dinh(i)=stinh(3,i)

```

```

      go to 667
901  ifuel = ifuel + 1
      istr = istr+1
      strokn = strmax
      strar(istr) = stroko
      stroko = strokn
      vfall = vfallm
      pcom = pcomr*(1.0-float(ifuel)/10.)
11   format (36h fuel setting reduced by ten percent)
      if (ifuel.gt.4) go to 663
      if (iout.lt.0) write (6,11)
      if (abs(strar(istr)-strmax) .gt. 10.0e-8) go to 650
c    reinitialize
761  continue
      do 664 i=1,m2
          aoh(i)=stinh(1,i)
          vinh(i)=stinh(2,i)
664  dinh(i)=stinh(3,i)
667  continue
      do 665 i=1,n1
          aop(i)=stinp(1,i)
          vop(i)=stinp(2,i)
          dop(i)=stinp(3,i)
665  reso(i)=stinp(4,i)
          vo=stinh(1,10)
          fpto=stinh(2,10)
          ipreig=stinh(3,10)
          po=stinh(4,10)
          go to 662
663  write (6,12)
          go to 300
12   format (20h no fuel conversion )
301  write (6,8)
8    format (25h stroke does not converge )
300  continue
c
c    one rult has been analyzed
c
      iblow=-1
      dfin = ((dfin/12.0)+qs(n)-q)*12.0
      if (abs(dfin).gt.0.0) iblow=12./dfin + 0.5
      if (ifuel.eq.5) ifuel = 4
900  continue
c    output for all hammers
      if(ityph.lt.3) pcome=(pcom-pcomr)/(pcomr*0.01)
      if(ioute.eq.1)go to 754
      call out2(out,inp,sth,hm,stp,pm,emp,rho,icol,m,n,ntp,jpmax,jdout,
1    temax,fintmx,ex,jmax,iw,iout,result,area,jex,ityph)
      if(ifuel.gt.0.and.ityph.eq.2)write (6,51)ifuel
51   format(11h0there were,i2,16h fuel reductions)
c
1    format (1h0,13x,37hstrokes analyzed and last return (ft),7f6.1)
      if (iostr.lt.0) go to 753

```



```

    if (ityph .lt. 3) write (6,1) (strar(i),i=1,istr),
* strokn
77  format (1h0,13x,42hreturned strokes and stroke analyzed, (ft),
1  7f6.1)
    go to 754
753  write (6,77) (strar(i),i=1,istr),stroko
c
70  format (14x,22h total pressure change,f6.1,2h ])
    write (6,70) pcomc
754  continue
10  il = -iult
c    jdout no. of time increments between output values + 1
c    jpmx maximum j in output
c    maxjp maximum mo. of output values
    maxjp=(jpmx/jdout)-1
    stpl=stp(1)/ept
c
c    plot
c
    if (iult.lt.nult) call output(out,result,inp,iout,ntp,jpmx,jdout,
1  il,n,maxjp,alph,iseq,jselec,ymax,col1,title,stpl,pm(1))
c
c    determine overall extrema for summary
c
    sexmin = 10.**6
    sexmax = 0.0
c
    do 600 i=1,n
    i3n=i+3*n
    i2n=i+2*n
    if(sexmax.ge.ex(i3n))go to 601
    sexmax=ex(i3n)
    iex(4*iult-1)=i
    iex(4*iult)=jex(i3n)
601  if(sexmin.le.ex(i2n))go to 600
    sexmin=ex(i2n)
    iex(4*iult-3)=i
    iex(4*iult-2)=jex(i2n)
600  continue
    result(iult)=rult*.5
    result(iult+10) = iblow
    if (result(iult+10).lt.0.0) result(iult+10) = 9999999.
    if (ityph .lt. 3) result(iult+20) = stroko
    result(iult+30) = sexmin
    result(iult + 40) = sexmax
c    anp(99) stores enthr
    result(iult+60)=anp(99)
    result(iult+80)=strokn
    if (ityph.eq.3) go to 700
c
c    determine blows/minute for diesels
c
    if (ityph.eq.1.and.pbar(20).gt.0.1) go to 680

```

```

      if (ityph.eq.2) go to 681
      tdown = sqrt((stroko-depib)*0.06217)*2.0
      go to 682
c     for closed end hammers find cycle time
681   call down(stroko,vfall,pbar,hm,depib,m,iw,inow,tdown)
      tdown=tdown*2.0
      go to 682
680   dcy = 0.0
c     for vacuum hammers only - determine time of cycle
      call vacham(pbar,aram,depib,dcy,stroko,fstr,vft,inow,iw)
      dd = depib-0.001
      call vacham(pbar,aram,depib,dcy,dd,fsbo,vfb,iout,iw)
      ac = 32.17*(1.0+0.5*(fstr+fsbo)/pbar(20))
      tdown = sqrt((stroko-depib)*2.0/ac)*2.0
682   continue
c     total cycle time
      tt = tdown + tstart+tdies+tup
      result(iult+50) = 60./tt
c     determine bounce chamber pressure
      if (pbar(1).gt.0.0) result(iult+70)=(pbar(8)/pbar(1))/0.144
      if (ifuel.eq.0) result(iult+70) = psi
700   if (iult.ge.nult.or.iblow.gt.1200.or.iblow.lt.0) go to 402
c
c
c     set up for next rult analysis
c
      i = iult
      if (ityph.eq.3.or.iostr.ne.0) go to 904
c     new stroke (assumed) for diesels
      strokn = 1.2*stroko
      if (iult.gt.1) strokn = result(iult+20)*1.5-result(iult+19)*0.5
      if (strokn.gt.strm) strokn=strm
c     velocity at ports
      vfall=0.0
      if (strokn.gt.depib) vfall=sqrt(64.34*(strokn-depib))
      dcyl = 0.0
      if (pbar(20).gt.0.0) call vacham(pbar,aram,depib,dcyl,strokn,fsuck
1    ,vfall,iout,iw)
      if (ityph.eq.1) go to 904
      vfall = vfallm
      if (strokn.lt.strmax)
1    call down(strokn,vfall,pbar,hm,depib,m,iw,iout,tdown)
      if (strokn.gt.strmax) strokn = strmax
      if (ifuel.gt.0) ifuel = ifuel - 1
      pcom = pcomr*(1.0-float(ifuel)*0.1)
904   continue
      if (iostr.eq.0) stroko = strokn
      if (iostr.ne.0) vfall = vfr
c
c     new static soil resistance
c
400   continue
      if (result(i+1).gt.0.001) rult=result(i+1)*2.

```

```

        if (result(i+1).gt.0.001) go to 399
        rult = rult + dult
399  iult=iult+1
        if (cdp(99).gt.0.5) go to 395
c      for proportionally increasing skin friction
        do 398 i=1,nl
        su(i) = su(i)*rult/(result(iult-1)*2.0)
398  sok(i) = su(i)/qs(i)
        go to 501
395  continue
c      for constant skin friction
        su(nl)=su(nl)+rult-result(iult-1)*2.0
        sok(nl) = su(nl)/qs(nl)
        go to 501
401  iult=iult-1
c
c      now one problem is done
c      output summary and
c      final plot
c
402  if (dfin.le.0.0) write (6,23)
4    format (1h1,12x,10a4/1h0,28x,7hsummary)
        write (6,4) title
        23 format (53h analysis was terminated as no permanent set resulted )
c      summary is output in sumout
        call sumout(result,ityph,iult,iw,iex)
c
        if (nult.eq.1) iult = 0
        call output (out,result,inp,iout,ntp,jpmax,jdout,iult,n,maxjp,
1    alph,iseq,jselec,ymax,col1,title,stpl,pm(1))
c
c
c      go to read new problem
c
        go to 9877
9876 continue
        return
        end
        subroutine iptn(jselec,iseq,ymax,alph,title,iostr,strm,ioute,
1    lpstp,ppm,prult,palph,pqs,psj,psu,nseg,pxpt,pap,pep,prp)
c
        common vinh,dinh,aoh,voh,doh,anh,vnh,dnh,sth,hm,am,sta,ma,esoil,
1    aop,anp,dop,dnp,vop,vnp,stp,pm,cdp,res,reso,sj,sok,qs,su,
2    splice,depib,aram,ecus,ecap,ept,eant,vfall,vcham,xpt,sjjj,
3    emp,rho,dth,ntp,n,m,jrat,iult,rult,soilm,dsacp,
4    fpto,iw,pcom,tdel,iout,out,expp,temi,temo,inp,stroko,strokn,vo,pc
5    ,dfin,jdout,jpmax,ex,icol,fintmx,temax,result,ismith,iter
6    ,ityph,ipreig,vfallm,strmax,dcyl,pbar,area,jex
        common/slak/open(99)
        dimension vinh(10),dinh(10),aoh(10),voh(10),doh(10),anh(10),
1    vnh(10),dnh(10),sth(10),hm(10), vnp(99),

```

```

2 aop(99),anp(99),dop(99),dnp(99),vop(99),stp(99),pm(99),cdp(99),
3 res(99),reso(99),sj(99),sok(99),qs(99),su(99),splice(99),
4dsacp(3),out(3200),inp(13),ex(600),result(100),jex(600)
1 ,ap(20),xp(20),title(10),ee(4),smith(2),viscou(2),dtyp(2)
2 ,am(10),sta(10),rp(20),ep(20),alph(99),hname(2),jselec(13),
3 iseg(15),pbar(20),area(99),pstp(99),ppm(99),palph(99),pqs(99),
4psaj(99),psu(99)

c
c initialize
c
  ir=5
  if=8
  data smith,viscou/4h s,4hmith,4h vis,4hcous/
  icol=0
  iw=6
  nhamr = 200
  pele=5.0
  nlim=98
  mlim=10
  do 30 i=1,3
30 dsacp(i)=.01
  cdp(1)=0.0
  splice(1)=0.
  do 40 i=2,99
  splice(i)=-5000.0
  if(i.le.10) result(i) = 0.0
  alph(i) = 0.0
40 cdp(i)=0.0
  exp=1.3
  tem1=25.0
  soilm=0.0
  sjjj=0.0
  dult=0.0
  do 146 i=1,13
146 jselec(i)=0

c
c start to read data
c read and print title
c if no more data - quit
c
1 format (20i4)
2 format (10f8.2)
4 format (1h1,19x,44hw e a p - wave equation analysis for piles//
1 35x,12h1980 version//
> ,5x,10a4)

c
  read (ir,5,end=9876,err=1020) title
  go to 9877
9876 iw =-1000
  return
9877 continue

```

```

      alph(1) = 1.0
5      format (10a4)
      write (6,4) title
c
c      read options and specs.
c
      read (ir,1,end=9876,err=1020) iout,ijj,ihamr,iostr,ifuel,ipel,n,
1 ispl,ncross,ibedam,ipercs,ismith,itys,maxjrt,iphi,iter,idaha
c
c      check and/or correct options
c
      ii=iabs(iout)
      ioute=0
      if(iout.eq.-100) ioute=1
      if(iout.eq.-100) iout=0
      pm(99) = 0.0
      if (idaha.eq.0) pm(99) = 2.0
      if (idaha.gt.0) pm(99) = idaha
      cdp(99) = 0.0
      if (ipercs.lt.0.0) cdp(99) = 1.0
      ipercs = iabs(ipercs)
      if (ibedam.eq.0) ibedam = 1
      if (ibedam.lt.0) ibedam = 0
      if (ii.ge.20) ii=ii-20
      if (ii.ge.10) ii=ii-10
      if (ii.gt.6) iout=0
      if (ifuel.lt.1.or.ifuel.gt.5) ifuel=1
      if (iter.eq.0) iter=2
      if (iter.lt.0) iter=0
      if (maxjrt.le.0) maxjrt=1
      if (ihamr.gt.nhamr) write (6,3)
      if (ihamr.lt.0) ihamr=0
      if (ii.eq.6) ij = 0
3      format (27h unknown hammer type, sorry)
      if (ihamr.gt.nhamr) stop
10     format (48h ***n was not set,extended input not possible***)
      if (n.le.1.and.ipel.gt.0) write (6,10)
      if (n.le.1.and.ipel.gt.0) stop
c
c      extended pile input
c
      do 111 i=1,nseg
      stp(i)=pstp(i)
      pm(i)=ppm(i)
      alph(i)=palph(i)
111 continue

      if (ipel.lt.2) go to 142
9      format (50h masses and stiffnesses for first element are not ,
*9hspecified)
      if (pm(1).lt.10.e-8.or.stp(1).lt.10.e-8) write (6,9)
      if (pm(1).lt.10.e-8.or.stp(1).lt.10.e-8) stop
      do 143 i=1,n

```

```

      stp(i)=stp(i)*12.0
      pm(i)=pm(i)/32.17
      if (stp(i).lt.0.0001) stp(i)=stp(i-1)
143  if (pm(i).lt.10.e-8) pm(i)=pm(i-1)
142  continue
      do 600 i=2,99
600  if (alph(i).lt.0.0001) alph(i)=alph(i-1)
      m1=8
      m2=9

c
c      driving system input
c

      read (ir,2) hm(m2),sth(m1),sth(m2)
      if(hm(m2).le.0.0)write (6,11)
11  format(51h need cap weight gtr 0; split ram, anvil or piletop)
      if(hm(m2).le.0.0) stop
      hm(m2)=hm(m2)/32.17
      sth(m2) = sth(m2)*12.
      sth(m1) = sth(m1)*12.
      read(ir,2) (ee(i),i=1,4),temax,b,a,c
      if (sth(m2).lt.1.0) ee(4) = 1.0
      do 141 i=1,4
      if (ee(i).lt.0.00001.or.ee(i).gt.1.0) ee(i) = 0.85
141  ee(i)=ee(i)**2
      eanv=ee(1)
      ecap=ee(2)
      ept=ee(3)
      ecus=ee(4)

c
c      ncross eql 0 means uniform
c
c      start pile input
c

      nn=2
      xp(1) = 0.0
      soilm=0.0
      tdorga=0.0
      al=0.0
      bl=0.0
      cl=0.0
      xpt=pxpt
      ap(1)=pap
      ep(1)=pep
      rp(1)=prp
      atp = ap(1)/144.0
      if (ep(1).lt.0.1) ep(1) = 29000.
      if (rp(1).lt.0.1) rp(1) = 492.
      emp=ep(1)*144.
      rho=rp(1)/32170.
      if (ncross.lt.1) go to 221
      i=2

```



```

c      for nonuniform piles
220  read (ir,2) xp(1),ap(1),ep(1),rp(1)
      nn=1
      if (ep(1).lt.0.1) ep(1) = ep(i-1)
      if (rp(1).lt.0.1) rp(1)=rp(i-1)
      i = i+1
12  format (48h pile properties at bottom of pile not specified)
      if (xp(i-1).lt.xp(i-2)) write (6,12)
      if (xp(i-1).lt.xp(i-2)) stop
      if (xp(i-1).lt.xpt) go to 220
      go to 222
221  continue
      ap(2)=ap(1)
      xp(2)=xpt
      ep(2)=ep(1)
      rp(2)=rp(1)
222  continue
c      determine no. of stations specified
      do 2100 i=1,20
      if (xp(i).ge.xpt-10e-8) go to 2101
2100  continue
2101  nn=i
      if (ipel.ne.0) go to 2102
      nl=xpt/pele+1
      if (n.le.1) n=nl
      if (n.gt.nlim) n=nlim
c
c      determine pile segments
c
2102  call piel(nn,n,xp,ap,ep,rp,stp,pm,alph,iout,iw,icol,ipel,
1  area)
c
c      obtain hammer data
c
      call hael(ihamr,dtcrrh,dtcra,hname,ir)
      if (abs(tdorga).gt.10.e-8) sjjj = tdorga
      strm = pbar(10)
c
c      check on self-generated pile data
c
      do 150 i=1,n
      if (stp(i).gt.100.0.and.pm(i).gt.10e-8) go to 150
7  format (54h pile masses or stiffnesses too small,check input info)
      write (6,7)
      stop
150  continue
c      fuel pressure and pressure multiplier for diesels
      pspec=pbar(10+ifuel)
      if (pspec.lt.0.01.and.ifuel.gt.1) pspec=1000.0
      if (vcham.gt.10.e-8) pcom=pspec/(14.7*(1.+depib*aram/vcham)**1.35)
      nml=n-1
      nl=n+1

```

```

      if (a.gt.0.1) pcom = a
      soilm=soilm/32.17
      if (a1.gt.10.e-8) dsacp(1)=a1
      if (b1.gt.10.e-8) dsacp(2)=b1
      if (c1.gt.10.e-8) dsacp(3)=c1
      if (c.gt.0.1) exp = c
      if (abs(b).gt.10.e-8) tdel=b

c
c      read soil and additional hammer data
c

      read (ir,2,end=9876,err=1020) qssss,qsqqqqq,sjjjj,sjsssss,rrrr,
1  esoil,stroke,efficy,psteam,rwt
      rult=prult
      if(psteam.gt.1.0) pcom=pcom*psteam/pspec
      if(psteam.gt.1.0.and.ityph.ne.3) write (6,163) psteam
163 format(27h comb. pressure input(psi)=,f6.1)
      sjs=sj(1)
      sjt=sj(n+1)
      if (efficy.gt.0.001) pbar(16)=efficy
      if (rwt.gt.0.0.and.ityph.ne.3) pbar(8)=rwt
      if (ityph.ne.3) go to 160

c
c      for a/s hammers override file data
c

      if (stroke.gt.0.001) pbar(10)=stroke
      if (stroke.lt.0.01) stroke=strm
      if (rwt.gt.0.0) pbar(18)=rwt
      if (psteam.gt.pbar(11)) psteam=pbar(11)
      if (psteam.lt.0.001) psteam=pbar(11)
c      double acting air/steam stroke (equivalent)
      if (pbar(11).gt.10.0) stroke=pbar(10)*(1.0+(psteam*pbar(18)))/(
1  pbar(11)*pbar(17)))
      vfall=sqrt(stroke*64.34*pbar(16))
      write (6,161) stroke,pbar(16),vfall
161 format (16h stroke (equiv.) ,f4.1,4h ft, ,11h efficiency ,
1  f5.2,18h , impact velocity ,f5.1,5h ft/s )

      if (pbar(11).gt.10.0) write (6,162) psteam,pbar(11),pbar(18),
1  pbar(17)
162 format (22h actual/ max. pressure ,f6.1,1h/,f6.1,23h psi, react./r
      lam weight,f6.1,1h/,f6.1,5h kips )
      do 164 i=21,30
164 result(i) = stroke
160 continue

c
c      determine ultimate capacity to be analyzed
c

      rult=2.*rult
      rmax=rult
      if (esoil.lt.10.e-4.or.esoil.gt.1.0) esoil=1.0
      esoil=esoil**2.
      do 114 i=2,n

```

```

      sj(i)=sj(1)
114  qs(i)=qs(1)
c
c      read extended soil input
c
      do 112 i=1,nseg+1
      qs(i)=pqs(i)
      sj(i)=psj(i)
      su(i)=psu(i)
112  continue

      if (itys.ge.0) go to 230
      do 200 i=2,n1
      if (qs(i).lt.0.0001) qs(i)=qs(i-1)
      if (sj(i).lt.0.0001) sj(i)=sj(i-1)
      if (itys.lt.-1.and.su(i).lt.0.0001) su(i)=su(i-1)
200  continue
230  continue

c
c      distribute resistance
c
      ityr = itys
      if (itys.ge.-1) call soiln (itys,ipercs,n,xpt,su,ir,alph)
      itys = ityr
      sus=0.
      do 184 i=1,n1
184  sus=sus+su(i)
      suj = sus - su(n1)

c
c      distribute case damping
c
      do 185 i=1,n1
      if (ismith.gt.0.or.i.eq.n1) go to 187
      if(itys.lt.0)go to 1187
      if (suj.lt.10.e-10) sj(i)=sj(i)/float(n)
      if (suj.ge.10e-10) sj(i)=sj(i)*su(i)/suj
1187  continue
      sj(i)=sj(i)*sqrt(stp(i)*pm(i))
187  continue
      qs(i)=qs(i)/12.
      su(i)=su(i)/sus

c
c      check quakes
c
185  if (qs(i).lt.0.0001) qs(i)=.08333
      if (ismith.lt.1) sj(n1)=sj(n1)*sqrt(stp(n)*pm(n))
6    format(i4,2f8.2)
      do 301 i=1,99
301  open(i)=0.0
      if (ispl.lt.1) go to 144

c
c      read splice and plotter info.
c

```

```

do 300 i=1,isp1
  read (ir,6,end=9876,err=1020) j,splice(j),open(j)
300  continue
  do 302 i=2,n
302  open(i)=open(i)+open(i-1)
    144  continue
    if (iabs(iout).gt.9) read (ir,2,end=1000,err=1020) ymax
    if (iabs(iout).gt.9) read (ir,1,end=1000,err=1020) (iseg(i),i=1,15
    1)
    go to 1003
1000  continue
1001  format (5lh insufficient plotter information,therefore values ,
  *llhare assumed)
  ymax = 0.0
  do 1004 i=1,15
1004  iseg(i) = 0
  write (6,1001)
1003  continue
c    determine pile damping
    dfac=float(ibedam)/50.
    do 145 i=1,n
145  cdp(i)=dfac*sqrt(pm(i)*stp(i))
c
c    print input info
c
  call outln(su,sj,qs,sth,stp,hm,pm,ihamr,ee,n,m,iw,xpt,ismith,
  lhname,esoil,ap(1),emp,rho,splice,cdp,alph,ityph,ioute)
  do 876 i=1,2
    dtyp(i) = viscou(i)
    if (ismith.eq.1) dtyp(i)=smith(i)
876  continue
    if (iphi.lt.100.and.ityph.lt.3) iphi=140
    if (iphi.lt.100.and.ityph.eq.3) iphi = 160
    phi = float(iphi)/100.
    pfac = 1./phi
8    format (/15x,26hoptions and specifications/
1 4h phi,f7.2,4x,12hs-damping ,2a4,4x,10hrwt (kips),f10.2/
2 5h iout,16,4x,10hp-damping ,110,4x,14hsoil dist. no.,16/
3 6h ifuel,15,4x,6hj skin,f14.2,4x,11htdel (sec.),f9.4/
4 6h iostr,15,4x,5hj toe,f15.2,4x,10htemax (ms),f10.2)
    do 188 i=2,n
      if (splice(i).gt.0.0) splice(i) = -splice(i)
      if (abs(splice(i)).lt.10.e-8) splice(i) = 10.e-8
      if (splice(i).lt.0.0.and.splice(i).ge.-.5)
        * write (6,189) i,splice(i)
189  format (10h at segmt.,i2,11h slack (ft),f10.5)
188  continue
47  format (7h soilm=,f5.2,5h kips,8h ds anv=,f5.3,
1 8h ds cap=,f5.3,9h ds ptop=,f5.3)
    is = itys
    if (is.eq.11) is = 0
    loe=iout
    if(ioute.eq.1.and.iout.eq.0)loe=-100

```

```

        write (6,8)phi,dtyp,rwt,ioe,ibedam,is,ifuel,sjs,b,iostr,
1  sjt,temax
506  format (15x,10hefficiency,f10.3)
        if (ityph.lt.3.and.pbar(16).gt.0.001) write (6,506) pbar(16)
        if (soilm.gt.0.0.or.al.gt.0.0.or.bl.gt.0.0.or.cl.
1  gt.0.0) write (6,47) soilm,al,bl,cl
        iult = 0
        if (rult.gt.10.e-4) go to 502
        if (rult.lt.0.) go to 191
c
c  generate rult values
c
        rult = sqrt(emp*rho)*atp*3.0
        i = rult*0.05 + 0.5
        if (i.lt.1) i = 1
        rult = i * 20.0
        if (emp.lt.3000000.) rult=rult/2.0
        rmax=rult*5.0
        iult=10
        go to 502
191  continue
c
c  read rult values
c
        read (ir,2) (result(i),i=1,10)
        do 500 i=1,10
        if (result(i).lt.0.001) go to 501
500  continue
501  continue
        rmax=0.001
        if(i.gt.1) rmax=result(i-1)*2.0
        iult = i-1
c
c  convert first rult to kips
c
        rult=result(1)*2.0
502  continue
        if (ityph.ne.1) go to 505
        dffall=stroke-depib
        if (dffall.gt.0.001) go to 106
c  stroke for vaccuum hammer
        stroke = 5.
        if (pbar(20).gt.10.e-8) stroke=3.0
        dcyl =0.0
        dffall = stroke-depib
106  continue
c
c  determine ram velocity at ports
c
        if (dffall.gt.0.0) vfall=sqrt(64.34*dffall)
        if (pbar(20).gt.10.e-8) call vacham(pbar,aram,depib,dcyl,stroke,
1  fsuck,vfall,iout,iw)
505  continue

```



```

c      soil stiffness
      do 115 i=1,n1
      su(i)=su(i)*rult
      sok(i) = su(i) / qs(i)
115    continue
c      check time increment including soil stiffness
      rrat = rmax/rult
      if (cdp(99).gt.0.5) rrat = 1
c
      dtcrp=pm(n)/((stp(n)+(sok(n1)+sok(n))*rrat)
      if (cdp(99).gt.0.5) dtcrp=pm(n)/((stp(n)+sok(n)+(rmax-rult)/qs(n1))
      nml=n-1
      do 1010 i=1,nml
      st = stp(i+1)
      if (stp(i).gt.st) st=stp(i)
      st=st+sok(i)*rrat
      dtm=pm(i)/st
      if (dtm.lt.dtcrp) dtcrp=dtm
1010    continue
      dtcrp=sqrt(dtcrp)
c      cushion damper
      pm(99) = pm(99)*sqrt(sth(m)*hm(m))/50.0
c      determine time increments for hammer and pile gtr critical
      dtp=pfac*dtcrp
      dth=pfac*dtcrh
c      check time increments
      if (dth.lt.10.e-6.or.dtp.lt.10.e-6) write (6,119)
      if (dth.lt.10.e-6.or.dtp.lt.10.e-6) stop
119    format (47h hammer or pile t.i. equal to zero, check input )
      if (dth.lt.dtcra) dth=pfac*dtcra
      if (dth.gt.dtp) go to 33
      do 118 jrat=1,20
      if (float(jrat+1)*dth.gt.dtp) go to 117
118    continue
c
c      determine time increments if hammer increments smaller than pile
c      t.i.
c
117    if (maxjrt.lt.1) maxjrt=1
      if (jrat.gt.maxjrt) jrat=maxjrt
      dtp=dth*float(jrat)
      go to 22
33    dth=dtp
      jrat=1
22    continue
21    format (15x,15htime incr. (ms),f5.3)
      dtpm=dtp*1000.
c
c      set up loading stiffnesses
c
      if (ityph.eq.3) sth(m1)=sth(m1)*ecus
      if (ityph.lt.3) sth(m1)=sth(m1)*ecap
      if (ma.gt.0) sta(ma)=sta(ma)*ecap

```



```

      sth(m2) = sth(m2)*ecus
      stp(1) = stp(1)*ept
      sth(m) = sth(m)*eanv
      write (6,21) dtpm
      stroko = stroke
c     set up output variables
      if (ijj.eq.1) read (ir,1,end=1005,err=1020) (inp(i),i=1,13)
      go to 1006
1005  ijj=0
1006  continue
      if (ijj.ne.1) call jjnp (icol,n,inp,jmax)
c   ... jmax=no. of pile variables that can be printed (jmax.le.n)
c     not used here, recomputed in analysis
      return
1020  continue
1021  format (11h data error)
      write (6,1021)
      stop
      end
      subroutine soiln (itys,ipercs,n,xpt,su,ir,alpha)

c     this program finds the percentage of total pile resistance that ac
c     at individual elements
c     note that this program assumes that pile elements are of equal len
c     except for top and bottom. these have 1/2 the length of the others
c     dis is an array that contains distributions of static soil resista
c     first index = type, second index 1=location 2=percentage
c     third index (max=6) for values
c     between given points of change of distribution the resistance is
c     assumed to vary linearly
c     xpt is the total pile length
c     perc* stands for percentage of total skin resistance at *
      dimension su(20),dis(11,2,20),alpha(99)
      do 201 i=1,11
      do 201 j=1,2
      do 201 k=1,20
      dis(i,j,k)=0.0
201  continue
      if (itys.lt.1) go to 300
c   the following are canned distributions: itys=1,...,5 triangular
c   itys=6,...,10 rectangular; grade assumed at top for itys=1 and 5,
c   20] below top for itys=2 and 6, ...
      do 202 i=1,10
      dis(i,1,4)=1.0
      d=float(i-1)*.2
      if(i.gt.5)d=d-1.0
      dis(i,1,2)=d
      dis(i,1,3)=d
      dis(i,2,2)=0.0
      dis(i,2,3)=0.0
      if(i.gt.5)dis(i,2,3)=1.0
202  dis(i,2,4)=1.0
c

```

```

c      go to 400
c
c      for input of soil distribuion
c
300  itys=11
2    format (2f8.3)
      do 306 i=1,20
        read (1r,2) dis(11,1,i),dis(11,2,i)
        il=i
        if (dis(11,1,i).gt.0.0) il=il+1
        disx=dis(11,1,i)
        dis(11,1,il)=dis(11,1,i)/xpt
        dis(11,2,il)=dis(11,2,i)/100.
        if (disx.ge.xpt-0.00001) go to 400
306  continue
400  xu=0.0
      if (dis(11,1,1).gt.0.0) dis(11,1,2)=0.0
      if (dis(11,1,1).gt.0.0) dis(11,1,1)=0.0
      percs=ipercs
      percs = percs*0.01
      if (percs.lt.10.0e-8) percs = 10.0e-8
      sumt=0.0
      salph=0.0
      do 110 i=1,n
110   salph=salph+alph(i)
      do 100 i=1,n
        xo=xu
        dl=alph(i)/salph
        xu=xo+dl
        sum=0.0
        do 101 j=2,20
          xxo = dis(itys,1,j)
          if (xxo.gt.xo) go to 102
101   continue
102   continue
          jl=j-1
          xxo=xo
          perco=dis(itys,2,jl)+ ((dis(itys,2,j)-dis(itys,2,jl))/(dis(itys,
1 1,j) - dis(itys,1,jl)))*(xo-dis(itys,1,jl))
          do 103 ij = j,20
            xxu=dis(itys,1,ij)
            if (xxu.gt.xu-10e-6) go to 104
            percu=dis(itys,2,ij)
            sum = sum + (perco+percu)*0.5*(xxu-xxo)
            xxo=xxu
            perco=percu
103   continue
104   continue
            xxu=xu
            j=ij
            jl=ij-1

```

```

    percu=dis(itys,2,j1)+ ((dis(itys,2,j)-dis(itys,2,j1))/(dis(itys,
1 1,j) - dis(itys,1,j1)))*(xu-dis(itys,1,j1))
    sum = sum + (perco+percu)*0.5*(xxu-xxo)
    su(i) = sum
    sumt = sumt + sum
100 continue
    do 105 i=1,n
105 su(i)=su(i)*percs/sumt
    su(n+1)=1.-percs
    return
end
subroutine stiff (ddis,dvel,stl,str,fo,fn,dsac,ddd)
    dlst = str - stl
    sti=stl+2.*dvel*dlst
    if (sti.lt.stl) sti = stl
    if (sti.gt.str) sti = str
    st=sti*dsac
    stl=st*ddd*.5
    stquad=stl*stl-fo*st
    if(stquad.lt.0.0)stquad=0.0
    stis=abs(stl)+sqrt(stquad)
    if (stis.gt.sti) stis = sti
    if (stis.lt.0.0) stis = 0.0
    fn = fo + ddd*stis
c
    return
end
subroutine sresn(dnp,dop,su,sok,i,reso,resn,soi)
    dimension dnp(1),dop(1),su(1),sok(1),reso(1),resn(1),soi(1)
    ddd=dnp(1)-dop(1)
    if(ddd.ge.0.0)dres=ddd*sok(i)
    if(ddd.lt.0.0)dres=ddd*soi(i)
    resn(i)=reso(i)+dres
    if (resn(i).lt.-su(i)) resn(i) = -su(i)
    if (resn(i).gt.su(i)) resn(i)=su(i)
    return
end
subroutine startc
    common vinh,dinh,aoh,voh,doh,anh,vnh,dnh,sth,hm,am,sta,ma,esoil,
1 aop,anp,dop,dnp,vop,vnp,stp,pm,cdp,res,reso,sj,sok,qs,su,
2 splice,depib,aram,ecus,ecap,ept,eav,vfall,vcham,xpt,sjjj,
3 emp,rho,dth,dtp,n,m,jrat,iult,rult,soilm,dsacp,
4 fpto,iw,pcom,tdel,iout,out,expp,temi,temo,inp,stroko,strokn,vo,pc
5 ,dfin,jdout,jpmax,ex,icol,fintmx,temax,result,ismith,iter
6 ,ityph,ipreig,vfallm,strmax,dcyl,pbar,area,jex
    common/slak/open(99)
    dimension vinh(10),dinh(10),aoh(10),voh(10),doh(10),anh(10),
1 vnh(10),dnh(10),sth(10),hm(10), vnp(99),
2 aop(99),anp(99),dop(99),dnp(99),vop(99),stp(99),pm(99),cdp(99),
3 res(99),reso(99),sj(99),sok(99),qs(99),su(99),splice(99),
4 dsacp(3),out(3200),inp(13),ex(600),result(100),jex(600)
5 ,am(10),sta(10),pbar(20),area(99)
    ipreig=0

```

```

    tcc=0.0
    temik=272.0+temi
    exp=1.35
    sttime = 0.0
    expt=1.0-1.0/exp
    patm=14.7*.144
    tempc=0.0
    if (tdel.gt.1.0) tempc=tdel
    if (vfall.lt.0.0) write (6,5)
5 format (27h negative velocity in start )
    if (pbar(16).gt.0.5) vfall=vfall*sqrt(pbar(16))
    if (vfall.lt.0.0) stop
    m1=m+1
    m2=m+2
    n1=n+1
c
c initialize hammer and pile variables
c
    dcy1 = 0.0
    delp = 0.0
    if (pbar(1).gt.aram) delp=(pbar(1)-aram)*0.144*14.7
    fsuck=0.0
    g=32.17
    ram=0.0
    do 100 i=1,m2
    if (i.lt.m1) ram = ram + hm(i)
    voh(i)=0.0
    doh(i)=0.0
    anh(i) = 0.0
100 aoh(i)=g
    do 101 i=1,n1
    res(i) = 0.0
    reso(i)=0.0
    anp(i) = 0.0
    vnp(i) = 0.0
    dnp(i) = 0.0
    vop(i)=0.0
    dop(i)=open(i)
101 aop(i)=1.e-6
c
c determine time increment
c
    stc1 = sth(m1)*0.5
    stp1=sth(m2)*stp(1)/(sth(m2)+stp(1))
    stp1 = stp1 * 0.5
    dtl=hm(m1)/stc1
    dt=hm(m2)/stp1
    if (dt.gt.dtl) dt=dtl
    dtl=hm(m2)/stc1
    if (dt.gt.dtl) dt=dtl
    dtl=pm(1)/stp1
    if (dt.gt.dtl) dt=dtl
    do 102 i=2,n

```

```

    dtl=pm(1)/(stp(1)+sok(1))
    if (i.eq.n) dtl=pm(n)/(stp(n)+sok(n)+sok(n+1))
102 if (dt.gt.dtl) dt=dtl
    nml = n-1
    dam = sj(n) + sj(n+1)
    if (ismith.eq.1) dam = sj(n)*su(n) + sj(nl)*su(nl)
    dam = dam/sqrt(pm(n)*stp(n))
    do 141 i = 1,n
    da = sj(i)/sqrt(pm(i)*stp(i))
    if (ismith.eq.1) da = da*su(i)
    if (da.gt.dam) dam = da
141 continue
    dam = dam*1.2
    if (dam.lt.2.) dam = 2.
    dt = sqrt(dt)/dam
    if (dt.lt.0.00001) dt = 0.00001
    tlim = 2.5*depib/vfall
    dlim = 0.25*depib
    if (dlim.lt.0.2) dlim = 0.2
c place ram at exhaust ports
    dtop=-depib
    doh(m)=-depib
    voh(m)=vfall
    aoh(m)=g
    d0 = vcham/aram
    dcu = pm(99)
    dct = 0.05*sqrt(stp1*hm(m2))
    din=depib+vcham/aram
    n5=n/2
    in=0
    j=0
    t2s = 0.0
    fslow = 0.0
    d3lc = (3.0*xpt/sqrt(emp/rho))*vfall
    dacrt = 0.5*depib
    if (d3lc.gt.dacrt) dacrt = d3lc
c dacrt is distance at which pile analysis starts
    if (iout.lt.0) write (6,1) dt,vfall,depib
1 format (14h dt, v, depib,f8.5,2f8.2)
2 format(13x,102hfp          fc          ft          v ram          v anv          v tc
lp      v mid      v toe      d ch      d top      toe res )
    if (iout.lt.0) write (6,2)
109 continue
    sttime = sttime+dt
    in=in+1
    j=j+1
    fslow = fslow + 0.025
    if (fslow.gt.1.) fslow = 1.0
    if (doh(m).gt.-dacrt) t2s = t2s + 0.025
    if (t2s.gt.1.0) t2s = 1.0
    it = 1
c
c predictor integration

```

```

      do 103 i=m,m2
        vnh(i)=aoh(i)*dt+voh(i)
103    dnh(i)=voh(i)*dt+doh(i)
        if (t2s.lt.10.e-8) go to 114
        do 104 i=1,n
          vnp(i)=aop(i)*dt+vop(i)
104    dnp(i)=vop(i)*dt+dop(i)
114    continue
        if (it.eq.3) go to 112
c      start wave analysis
        dbc = -dnh(m)
        if (pbar(20).gt.0.0) call vacham(pbar,aram,depib,dcyl,dbc,fsuck,
1    vfall,iout,iw)
        fp=aram*patm*((din/(d0+dnh(m1)-dnh(m)))*exp)
        fp = fp-fsuck+delp
        fcu = (dnh(m2)-dnh(m1))*stc1+(vnh(m2)-vnh(m1))*dcu
        fpt = (dnp(1)-dnh(m2))*stp1+(vnp(1)-vnh(m2))*dct
        dtop=dnh(m)-dnh(m1)
        anh(m) = g+(-fp+ftr(dtop,pbar,iwt))*fslow/ram
        anh(m1) = g + (fp*fslow+fcu)/hm(m1)
        anh(m2)=g+(-fcu+fpt)/hm(m2)
        do 305 i=m,m2
          vnh(i) = voh(i) + anh(i)*dt
305    dnh(i) = doh(i) + (voh(i) + vnh(i)) * 0.5 * dt
          if (t2s.lt.10.e-8) go to 112
          fo = -fpt*t2s
          do 105 i=1,n
            fu = (dnp(i+1)-dnp(i))*stp(i+1)+(vnp(i+1)-vnp(i))*cdp(i+1)
            res(i)=reso(i)+(dnp(i)-dop(i))*sok(i)
            if (res(i).gt.su(i)) res(i)=su(i)
            dam=sj(i)*vnp(i)
            if (ismith.eq.1) dam=dam*res(i)
            if (i.eq.n) go to 106
            anp(i)=g+(fo+fu-res(i)-dam)/pm(i)
            vnp(i) = vop(i) + anp(i) * dt
            dnp(i) = dop(i) + (vnp(i) + vop(i))*dt*0.5
105    fo=-fu
106    res(n1)=reso(n1)+(dnp(n)-dop(n))*sok(n1)
            if (res(n1).gt.su(n1)) res(n1)=su(n1)
            dat = sj(n1)*vnp(n)
            if (ismith.eq.1) dat=dat*res(n1)
            anp(n) = g+(fo-res(n1)-res(n)-dat-dam)/(pm(n)+soilm)
            vnp(i) = vop(i)
            vnp(n) = vop(n) + anp(n)*dt
            dnp(n) = dop(n) + (vnp(n) + vop(n))*0.5*dt
112    continue
            vnp(n1) = vnp(n)
            dnp(n1) = dnp(n)
            do 107 i=m,m2
              aoh(i)=anh(i)
              voh(i)=vnh(i)
107    doh(i)=dnh(i)
            if (t2s.lt.10.e-8) go to 113

```



```

do 108 i=1,n1
  reso(i)=res(i)
  aop(i)=anp(i)
  vop(i)=vnp(i)
108 dop(i)=dnp(i)
113 continue
  dl=dop(1)*12.
  dcham=(doh(ml)-doh(m))*12.
  if (iout.lt.0.and.in.eq.10) write (6,6) j,fp,fcu,fpt,voh(m),
  lvoh(ml),vop(1),vop(n5),vop(n),dcham,dl
  2 ,reso(n1)
  if (in.eq.10) in=0
  6 format (i5,11f10.3)
c ram close enough at impact block for ignition check
  if(voh(m).lt.0.0)go to 214
c ram turned around w/o impacting
  if (-doh(m).gt.dlim) go to 109
  po = fp/aram
  temo = temik*(po/patm)**expt
  timp=1.0
  if (voh(m).gt.voh(ml)) timp=(doh(ml)-doh(m))/(voh(m)-voh(ml))
c
c check whether ignition or impact is about to occur
c
  if (-tdel.gt.timp-0.001) go to 203
c pbar(9) is dist. between atom. inj. and anvil
  if (pbar(9).gt.0.0.and.tdel.lt.10.e-8) go to 212
  if (tempc.gt.1.0.and.temo.gt.tempc) go to 206
  if (-doh(m).lt.pbar(9).and.pbar(9).gt.0.0) tcc = tcc + dt
  if (tcc.gt.tdel-0.001.and.pbar(9).gt.0.0) go to 203
  go to 212
203 ipreig=0.001/dtp+0.5
c ignition will occur after 1 millisec.
  go to 202
206 ipreig = 0.00025/dtp + 0.5
c temperature ignition after 1/4 msec.
  go to 202
212 continue
  if (float(j)*dt.lt.tlim) go to 213
  write (6,9)
  ipreig = -100
  go to 202
214 continue
10 format(33h *** no steel to steel impact ***)
  write (6,10)
  go to 202
213 continue
  9 format (28h *** hammer will not run ***)
  if (dnh(ml)-dnh(m).gt.0.01*depib) go to 109
c ram not close enough and no ignition
c for 0 tdel and atomized fuel injection
c (assume at impact)
  if (tcc.gt.10.e-8) ipreig = (tdel-tcc)/dtp

```

```

        if (tcc.gt.10.e-8.and.ipreig .lt.1) ipreig = 1
c      now data preparation for diesel
202    mml=m-1
        fpto = -fpt
        vo = -fcu
        do 111 i=m,m2
            dinh(i)=doh(i)
111    vinh(i)=voh(i)
        do 110 il=1,mml
            i=m-il
            vinh(i)=voh(m)
110    dinh(i) = dinh(i+1) + fp/sth(i)
        fintmx = sttime
        return
    end
    subroutine vacham(pbar,aram,depib,dcyl,dbc,fsuck,vfall,iout,iw)
    dimension pbar(1)
    art = pbar(1)
    expb = pbar(4)
    dstart = pbar(2)
    exp=expb
    vin = pbar(5)
c      dstart is the distance that the ram moves before it closes the vac
c      chamber off.
        dela = art-aram
        din = vin/dela
        patm = 0.144*14.7
        if (dbc.lt.depib) go to 100
c      dbc is the distance of the ram from the anvil zero position or it
c      is the stroke,i.e  dbc = doh(m) for -doh(m) lt depib  dbc=stroke
c      dbc less than depib means dbs is not stroke
        fac = 1.0/(expb-1.0)
        ramw = pbar(20)
c      pbar(20) is the ram weight which is not needed for other than vac
c      hammers
c
        db = dbc + din - dstart
        de = depib + din - dstart
        vfall = dela*patm*(din**exp)*fac/ramw
        vfall = vfall*(db**(1.-exp)-de**(1.-exp))
        vfall = vfall + (1.0+patm*dela/ramw)*(dbc-depib)*2.0*32.17
        if (vfall.lt.0.0) write (6,2) vfall
        if (vfall.lt.0.0) stop
2      format (70h improper condition met when determining ram velocity a
1t ports, vfall=, el0.4)
        vfall = sqrt(vfall)
        return
100    dvn = dcyl+dbc-dstart+din
        if(dvn.lt.din) dvn=din
        fsuck = (din/dvn)**expb
        if (fsuck.gt.1.0) fsuck = 1.0
        fsuck = dela*patm*(1.0-fsuck)
1      format (31h negative volume in vac chamber,4f8.2)

```

```

return
end
subroutine up(pbar,dnh,vnh,dnp,vnp,hm,pm,m,n,dcyl,dfin,vfall,vo,
1 po,vfallm,stroke,depib,aram,exp,iwt,iw,iout,psi,t)
dimension pbar(1),dnh(1),vnh(1),dnp(1),vnp(1),hm(1),pm(1)

```

```

c
c this program is used to determine the velocity of the ram of
c ced's at the exhaust ports. the analysis uses a rigid body
c pile and ram assumption for closed end hammers whose upward
c velocity will not cause uplift the total stroke is determined
c

```

```

patm = 14.7*0.144
dfl = dfin/12.
t = 0.0
vr = 0.0
ml = m+1
m2=m+2
vp=vnh(ml)*hm(ml)+vnh(m2)*hm(m2)
vcham = vo
if (dnh(ml).gt.dnh(m)) vcham = vo-(dnh(ml)-dnh(m))*aram
g = 32.17
cylm=pbar(8)/g
fatm = patm*aram
delp = 0.0
if (pbar(1).gt.0.0) delp=(pbar(1)-aram)*2.117
if (pbar(1).gt.aram) fatm = patm*pbar(1)
if (pbar(20).gt.0.0) fatm=0.0
acyl=g
if (pbar(8).gt.0.0) acyl=dnh(10)
vcyl = vnh(10)
pam = 0.0
iblow = 0
ram = 0.0
iwt = 0
tt=0.0
in=1

```

```

1 format (7x,83htt      fp      fb      r vel      r pos      v cyl
1      d cyl      v top      d top      )
if (iout.lt.0) write (6,1)

```

```

c
c determine rigid ram velocity
c

```

```

do 100 i=1,m
ram = ram + hm(i)
100 vr = vr + vnh(i)*hm(i)

```

```

c
c determine rigid pile velocity
c

```

```

do 101 i=1,n
pam = pam + pm(i)
101 vp = vp + vnp(i)*pm(i)
vdr = vr/ram
vdp=vp/(pam+hm(ml)+hm(m2))

```

```

3      if (vdr.gt.-0.05) write (6,3)
      format (4lh ram still moving downward at end of blow)
      if (vdr.gt.-0.05) stop
      ddr=dnh(m)
      ddp=dnh(m1)
      dcdt=dcyl-dnh(m1)
      vcvt=vcyl-vnh(m1)
      vcyl=vdp+vcvt
      dcyl=ddp+dcdt
      if (dcyl.gt.ddp) vcyl=vdp
      if (dcyl.gt.ddp) dcyl=ddp
      iup = 0
      if (vdp.lt.0.0) iup =1
      dpos = ddr
      drais = 0.01*depib

```

```

c
c      start loop to determine incrementally the ram speed during
c      ram upwards motion
c
1000  delt = abs(drais/vdr)
      t=t+delt*float(1-iblow)
      tt=tt+delt
      in=in+1
      if (iout.lt.0.and.in.eq.5) write (6,2) tt,fp,fb,vdr,dpos,vcyl,
1      dcyl,vdp,ddp
      if (vcyl.lt.0.0.and.vcyl.lt.vdp) vcyl=vdp
2      format (9f10.4 )
      if (in.eq.5) in=0
      if (iup.eq.1.and.ddp.lt.dfi) vdp = 0.0
      if (iup.ne.1.and.ddp.gt.dfi) vdp = 0.0
      ddp = ddp + vdp*delt
      vcyl = vcyl + acyl * delt
      dcyl = dcyl + vcyl * delt
      if (dcyl.ge.ddp) vcyl=vdp
      if (dcyl.gt.ddp) dcyl = ddp
      dpos = dpos - drais
      v = vcham + (ddp-dpos)*aram
      fp = 2.117*aram
      if (iblow.eq.0) fp =(po*(vo/v)**expp)*aram
      dcyra = dpos-dcyl
      dcb = -dpos
      fb = ftr(dcyra,pbar,iwt)
      if (pbar(20).gt.0.0) call vacham(pbar,aram,depib,dcyl,dcb,fb,vfall
1      ,iout,iw)
      if (pbar(8).gt.0.0) acyl=g+(fatm-fb)/cylm
      if (pbar(20).gt.0.0) fb=fb+patm*aram
      if (pbar(20).lt.10.e-8) fb=fb-delp
      ac = g-(fp-fb)/ram
      ve = vdr + ac*delt
      vdr = ve
      vo = v
      po = fp/aram
      if (dpos.gt.-depib+dcyl) go to 1000

```

```

      if (iblow.eq.1) go to 1001
c
c      iblow = 1 means ram has passed the exhaust ports
c
      iblow = 1
      vfall = -vdr
c      for open end hammers
      stroke = vdr*vdr/64.34 + depib
      draiss = 0.05
      if (pbar(20).gt.0.0) go to 1001
      if (pbar(6).lt.0.001) return
      if (vfall.gt.vfallm) iwt = 1
      if (vfall.gt.vfallm) psi=pbar(8)/(pbar(1)*0.144)
c
c      vfall gt vfallm means uplift
c
      if (vfall.gt.vfallm) return
c
c      for hammers whose bounce chamber ports have not yet been closed
c
      depa = pbar(7)-pbar(6)-depib
      if (deps.lt.0.0) go to 1001
      if (vdr*vdr.lt.64.34*deps) go to 1001
      vdr = sqrt(vdr*vdr-64.34*deps)
      dpos = dpos-deps
      vdr = -vdr
1001 if (vdr.lt.0.0.and.iwt.ne.1) go to 1000
c
c      ram reached upmost point
c
      stroke = -dpos-dcyl
c
c      bounce chamber pressure in psi (for vfall lt vfallm)
c
      dpos = -stroke
      fb = ftr(dpos,pbar,iwt)
      psi=fb/(pbar(1)*0.144)-14.7
      return
      end
      subroutine pilean (itp,aop,vop,dop,anp,vnp,dnp,stp,pm,splice,sj,
1  cdp,res,reso,su,sok,vnpit,vcap,soilm,sjjj,ntp,n,it,ismith,epsv,
2  iai,esoil,ftp,jrat,iter,soi,dt2,dt6,pmi,raum,dsum)
      dimension aop(99),vop(99),dop(99),anp(99),dnp(99),pm(99),stp(99),
1  splice(99),sj(99),cdp(99),res(99),reso(99),su(99),sok(99),vnp(99)
2  ,soi(99),pmi(99)
c      this subprogram performs a wave analysis for a pile subject
c      to either of the following three pile top variables
c      iai equal to 0 , hammer force used in pilean
c      iai equal to 1, measured force used in pilean
c      iai equal to 2, measured acceleration used in pilean
c      start to compute the pile variables
c
c      splice is usually at the structural limit of the pile material (te

```



```

c      hus negativ) except where splices give lower value
c      if splice is between 0 and -0.5 ft it is a slack
c      it is the iteration counter (assigned outside)
c      fu and fo are forces at an element downwards positively
c
      nml=n-1
      itp=0
      if (it.gt.0) go to 550
      il=1
c      prediction
      if (iai.lt.2) call integr(dtp,aop,aop,vop,dop,vnp,dnp,il,dt2,dt6)
      if (iai.eq.2) call integr(dtp,aop,anp,vop,dop,vnp,dnp,il,dt2,dt6)
      g=32.170
      do 300 i=2,n
      call integr(dtp,aop,aop,vop,dop,vnp,dnp,i,dt2,dt6)
300    continue
550    continue
c
c      wave analysis
c      first pile top
c
400    fu = (dnp(2)-dnp(1))*stp(2)+(vnp(2)-vnp(1))*cdp(2)
      fod = (vcap-vnp(1))*cdp(1)
      if (fod+ftp.lt.splice(1)) fod = 0.0
      if (iai.eq.1) fod = 0.0
      if (fu.gt.-splice(2)) fu = -splice(2)
      if (splice(2).lt.-0.5.or.fu.lt.0.0) go to 175
      ext = dnp(2)-dnp(1)
      if (ext.lt.0.0) go to 175
      fu = 0.0
      if (ext.lt.-splice(2)) go to 175
      fu = (ext+splice(2))*stp(2)+(vnp(2)-vnp(1))*cdp(2)
175    continue
      i = 1
      call sresn(dnp,dop,su,sok,i,reso,res,soi)
      da = vnp(1)*sj(1)
      if (ismith.eq.1) da = da * res(1)
      if (iai.ne.2) go to 200
      ftp = (anp(1)-g)*pm(1)-fu-fod+res(1) + da
      go to 201
200    continue
      anp(1)=g+(ftp+fod+fu-res(1)-da)*pmi(1)
      call integr(dtp,aop,anp,vop,dop,vnp,dnp,i,dt2,dt6)
      rsum=res(1)
      dsum=da
c
c      elements between top and toe
c
201    do 301 i=2,nml
      il=i+1
c
      fo = -fu
      fu = (dnp(il)-dnp(i))*stp(il) + (vnp(il)-vnp(i))*cdp(il)

```



```

    if (fu.gt.-splice(il)) fu = -splice(il)
    if (splice(il).lt.-0.5.or.fu.lt.0.0) go to 176
    ext = dnp(il)-dnp(i)
    if (ext.lt.0.0) go to 176
    fu = 0.0
    if (ext.lt.-splice(il)) go to 176
    fu=(ext+splice(il))*stp(il)+(vnp(il)-vnp(i))*cdp(il)
176  continue
    call sresn(dnp,dop,su,sok,i,reso,res,soi)
    da = vnp(i) * sj(i)
    if (ismith.eq.1) da = da * abs(res(i))
    rsum=rsum+res(i)
    dsum=dsum+da
    anp(i)=g+(fo+fu-res(i)-da)*pmi(i)
    call integr(dtp,aop,anp,vop,dop,vnp,dnp,i,dt2,dt6)
301  continue
c
c    start pile toe analysis
c
    nl=n+1
    dnp(nl)=dnp(n)
    call sresn(dnp,dop,su,sok,nl,reso,res,soi)
    if (res(nl).lt.0.0) res(nl)=0.0
    dab=vnp(n)*sj(nl)
    if (ismith.eq.1) dab=dab*abs(res(nl))
    if (dab.lt.0.0) dab=0.0
c
c
    call sresn(dnp,dop,su,sok,n,reso,res,soi)
    da = vnp(n) * sj(n)
    if (ismith.eq.1) da = da*abs(res(n))
    fo = -fu
    anp(n)=g+(fo-res(n) -da-res(nl)-dab)*pmi(n)
    rsum=rsum+res(n)+res(nl)
    dsum=dsum+da+dab
    call integr(dtp,aop,anp,vop,dop,vnp,dnp,n,dt2,dt6)
    itp=itp+1
    if (jrat.eq.1) go to 750
    if (itp.gt.iter) go to 750
c    check convergence (if dth not eq dtp)
    err = abs(vnpit-vnp(n))
    vnpit = vnp(n)
c    for no convergence repeat analysis
    if (err.gt.epsv) go to 400
750  continue
    return
end
subroutine jjnp(icol,n,jnp,jmax)
dimension jnp(13)
jmax=n
c    for n le 13 or 6, for icol gt or eq 0, respectively
do 105 i=1,13
    jnp(i)=0

```

```

105  if(1.le.n) jnp(1)=1
      if(icol.eq.0)go to 115
c    for 132 col. form
      nspace=11
      if(n.lt.14)go to 400
      jmax=13
      go to 120
115  nspace=4
c    for 70 col. form
      if(n.lt.7)go to 400
      jmax=6
120  ns=nspace+1
      dincr=float(n-2)/float(nspace)
      dincr=2.0
      do 130 i=3,ns
        dincr=dincr+dincr
130  jnp(i)=dincr+.5
      jnp(ns+1)=n
400  return
      end

c
c
      subroutine outln (su,sj,qs,sth,stp,hm,pm,ihamr,ee,n,m,iw,xpt,
1  ismith,hname,esoil,apo,emp,rho,splice,cdp,alph,ityph,ioute)
      dimension hname(2),su(99),sj(99),qs(99),sth(10),hm(10),ee(4),
1  stp(99),pm(99),cree(99),splice(99),cdp(99),dim(4)
2  ,alph(99)
      data dim/4h (s,4h/ft),4h (ks,4h/ft)/
4  format (15x,14h hammer model ,2a4)
5  format(6x,41h element      weight  stiffness      coeff. )
6  format(6x,43h number      (kips)   (k/in)   restitution )
7  format(6x,9h anvil      ,f10.3,f10.1,f10.3)
8  format(6x,9h cap      ,f10.3,f10.1,f10.3)
9  format (6x,9h cushion ,10x,f10.1,f10.3)
10 format (6x,9hpile top ,20x,f10.3)
12 format(53h      weight stiffn.  pdamp.  splice  soil-s  soil-d,
1  16h      quake  l.b.t.)
13 format(37h      (kips) (k/in) (ks/ft) (kips),8x,2a4,
1  16h      (in) (ft))

c
c
1401 format (9x,i2,4x,f10.3,f10.1,3f10.3,f10.1)
14 format (i5,f8.3,f8.0,f8.2,f8.0,3f8.3,f8.1)
15 format (2x,3htoe,32x,3f8.3)
16 format (/2x,35hcoefficient of restitution of soil      ,f8.3)
11 format (26x,17h pile properties //)
17 format (2x,12hpile length=,f5.0,5h ft.,,14h area(at top)=,
  *f6.1,5h s-in)
18 format (2x,17he. modul(at top)=,f6.0,6h ksi.,,
  *19h spec. wt.(at top)=,f5.0,10h lbs/cu ft//)
4711 format(1h )
      m1=m+1
      m2=m+2

```

```

do 4712 i=1,3
c
c      output hammer model
c
4712 write (6,4711)
do 100 i=1,4
100  cree(i) = sqrt(ee(i))
    write (6,4) hname(1),hname(2)
    write (6,5)
    write (6,6)
    wth=hm(1)*32.17
    stfh=0.0
    i=1
    write (6,1401) i,wth
    if (m.eq.1) go to 104
    do 101 i=2,m
    stfh=sth(i-1)/12.0
    wth=hm(i)*32.17
101  write (6,1401) i,wth,stfh
104  continue
    anvwt=hm(m1)*32.17
    anvstf=sth(m)/12.0
    capwt=hm(m2)*32.17
    capstf=sth(m1)/12.0
    if (ityph.eq.3) capwt=hm(m1)*32.17
    if (ityph.eq.3) capstf=sth(m)/12.0
    if (ityph.ne.3) write (6,7) anvwt,anvstf,cree(1)
    erep=cree(2)
    if (ityph.eq.3) erep=cree(1)
    write (6,8) capwt,capstf,erep
    cshstf=sth(m2)/12.
    if (ityph.eq.3) cshstf=sth(m1)/12.0
    if (cshstf.gt.10.**9.) cshstf=0.
    write (6,9) cshstf,cree(4)
    if (cshstf.lt.0.00001) cshstf=10.**12
    write (6,10) cree(3)
c
c      print pile properties and model
c
4713 format (1h1)
    if (ioute.ne.1)write (6,4713)
    write (6,11)
    write (6,17) xpt,apo
    a1 = emp/144.
    a2 = rho*32170.
    write (6,18) a1,a2
    i1=1
    if(ismith.ne.1)i1=3
    i2=i1+1
    write (6,12)
    write (6,13)(dim(i),i=i1,i2)
c
    salph=0.0

```

```

do 200 il=1,n
200 salph=salph+alph(il)
   dd = 0.0
   do 102 i=1,n
      pwt= pm(i)*32.17
      pstf= stp(i)/12.0
      qsf=qs(i)*12.0
      dd=dd+xpt*alph(i)/salph
      sum=pwt+pstf+cdp(i)+abs(splce(i))+su(i)+sj(i)+qsf
      dif=abs(sum-suo)
      suo=sum
      if(i.lt.3.or.i.ge.n) dif=1.0
      if(ioute.ne.1) dif=1.0
      if(dif.gt.1.e-4)write (6,14)i,pwt,pstf,cdp(i),splce(i),su(i),
1      sj(i),qsf,dd
102 continue
   al = qs(n+1)*12.0
   write (6,15) su(n+1),sj(n+1),al
22  format (45h skin friction constant for all rult values )
   if (cdp(99).gt.0.5) write (6,22)
   al = sqrt(esoil)
   write (6,16) al
   return
end
subroutine integr(dt,xo,xn,xio,xioo,xin,xiin,i,dt2,dt6)
dimension xo(1),xn(1),xio(1),xioo(1),xin(1),xiin(1)
xin(i)=xio(i)+dt2*(xo(i)+xn(i))
xiin(i)=xioo(i)+xio(i)*dt+(xo(i)+xo(i)+xn(i))*dt6
return
end
subroutine out2(out,jnp,sth,hm,stp,pm,emp,rho,icol,m,n,ntp,jpmax,
1lj dout,jemax,fintmx,ex,jmax,iw,iout,result,area,jex,ityph)
real in
dimension sth(10),hm(10),stp(99),pm(99),conv(10),stscnv(15),
1out(3200),jnp(13),ex(600),result(99),area(99),jex(600)
data p,psi,sea,in/4h p,4hpsi),4h sea,4h in)/
h1 = p
h2 = psi
if (ityph.eq.3) h1 = sea
if (ityph.eq.3) h2 = in
1  format (25x,30h hammer and pile forces(kips) )
2  format (22x,36h hammer and pile velocities(ft/sec) )
3  format (24x,32h hammer and pile stresses (ksi) )
4  format (22x,36h hammer and pile accelerations(g's) )
5  format (20x,40h hammer and pile displacements (inches) )
6  format (51x,30h hammer and pile forces(kips) )
7  format (48x,36h hammer and pile velocities(ft/sec) )
8  format (50x,32h hammer and pile stresses (ksi) )
9  format (48x,36h hammer and pile accelerations(g's) )
10 format (46x,40h hammer and pile displacements (inches) )
11 format (2x,25h jp time hammer , 9x,15h pile elements )
12 format(6x,29h (ms) ram m anvil top ,16,4i8)
13 format (2x,25h jp time hammer ,30x,15h pile elements )

```

```

14  format (6x,29h (ms)  ram m      anvil      top ,16,11i8)
15  format (15,f6.1,15f8.1)
16  format (15,f6.1,15f8.3)
21  format(15,f6.1,15f8.2)
31  format (15h      j      time      ,a4,5x,28hd ram      anv      ftop      vtop,
      *62h      dtop      fmid      vmid      dmid      sum st sum dp rt toe      ftoe,
      *15h      vtoe      dtoe)
32  format (15h      j      time      ,a4,6x,27hftop      vtop      dtop      ftoe,
      *15h      vtoe      dtoe)
33  format (8x,9h(ms)      (,a4,32h      (in)      (in)      (kips) (ft/s) ,
      *61h      (in)      (kips) (ft/s)      (in)      (kips) (kips) (kips)      (kips),
      *14h      (ft/s)      (in))
34  format (8x,9h(ms)      (,a4,32h      (kips) (ft/s)      (in)      (kips),
      *14h      (ft/s)      (in))
35  format (15,f7.1,f9.1,2f8.3,2(f9.1,f6.1,f8.3),f9.1,2f7.1,f9.1,f6.1,
      *f8.3)
36  format (15,f7.1,f9.1,2(f9.1,f6.1,f8.3))
      ml = m+1
      klim = 200
      imax = (jmax+2)*klim
      k = 1
      kk = 1

c
      ehmr = 31000.*144.
      rhohmr=0.492/32.17

c
      output option for printing
      iiout = iabs(iout)
      if (iiout.ge.20) iiout=iiout-20
      if (iiout.ge.10) iiout=iiout-10
      if (iiout.eq.0) go to 135

c
      for iiout eq 0 do not print variables vs time
      conv(1) = 1.
      conv(2) = 1.
      conv(3)=1.0
      conv(4) = 1./32.17
      conv(5) = 12.
      if (icol.gt.0) go to 101

c
      title for 132 columns
      if (iiout.eq.1) write      (6,1)
      if (iiout.eq.2) write      (6,2)
      if (iiout.eq.3) write      (6,3)
      if (iiout.eq.4) write      (6,4)
      if (iiout.eq.5) write      (6,5)
      if (iiout.eq.6) write      (6,32) h1
      if (iiout.eq.6) write      (6,34) h2
      if (iiout.eq.6) go to 130
      write      (6,11)
      write      (6,12) (jnp(i),i=2,jmax)
      go to 102
101  continue

c
      title for 80 columns
      if (iiout.eq.1) write      (6,6)
      if (iiout.eq.2) write      (6,7)

```



```

      if (iiout.eq.3) write (6,8)
      if (iiout.eq.4) write (6,9)
      if (iiout.eq.5) write (6,10)
      if (iiout.eq.6) write (6,31)h1
      if (iiout.eq.6) write (6,33) h2
      if (iiout.eq.6) go to 130
      write (6,13)
      write (6,14) (jnp(i),i=2,jmax)
102  continue
      if (iiout.ne.3) go to 107
c
c      stress conversion from forces
c
      const1 = 1./144.
      if (m.gt.1) const1 = sqrt(sth(m-1)*hm(m)/(ehmr*rhohmr))
      const2=sqrt(sth(m)*hm(m1)/(ehmr*rhohmr))
      const1=1.0/(const1*144.0)
      const2=1.0/(const2*144.0)
      do 103 i=1,jmax
      k = jnp(i)
103  stscnv(i)=1.0/area(k)
      k = 1
      kk = 1
      do 106 i=1,imax
      if (i.gt.klim) go to 104
      out(i) = out(i)*const1
      go to 106
104  if (i.gt.2*klim) go to 105
      out(i) = out(i)*const2
      go to 106
105  out(i) = out(i) * stscnv(kk)
      k = k+1
      if (k.le.klim) go to 106
      k = 1
      kk = kk + 1
106  continue
c
c      for all except 6 option
c
107  do 108 i=1,imax
108  out(i) = out(i) * conv(iiout)
      jj = 0
      do 110 jp=jdout,jpmax,jdout
      jj = jj+1
      jjmax = (jmax+1)*klim+jj
      tt = float(jp)*dtp*1000.
      if (iiout.eq.1.or.iiout.eq.2.or.iiout.eq.4) write (6,15) jp,tt,
1(out(i),i=jj,jjmax,klim)
      if (iiout.eq.3) write (6,21) jp,tt,(out(i),i=jj,jjmax,klim)
      if (iiout.eq.5) write (6,16)jp,tt,(out(i),i=jj,jjmax,klim)
110  continue
c
c      for 6 option only, first 132 columns

```



```

c
130  if (iout.ne.6) go to 135
      if (iout.eq.6.and.icol.eq.0) go to 131
      kp=0
      do 132 jp = jdout,jpmax,jdout
      kp=kp+1
      kpmax=15*klim+kp
132  write (6,35) jp,(out(1),i=kp,kpmax,klim)
      go to 135
c
c      6 option - 80 columns
c
131  kp=0
      do 134 jp = jdout,jpmax,jdout
      kp=kp+1
      kpmax1=200+kp
      kp2 = 4*klim+kp
      kpmax2=9*klim+kp
134  write (6,36) jp,(out(1),i=kp,kpmax1,klim),(out(1),i=kp2,kpmax2,
1    klim)
135  continue
c
c      for all options extreme value tables
c
17   format(1h0,24x,27htable of extrema, time step)
18   format(49h0no.      fmin,jmn      fmax,jmx      strmin,jsn strmax,jsx,
1     21h      vmax,jvx      dmax,jdx)
19   format(49h          (k)          (k)          (ks1)          (ks1)          ,
1     21h      (f/s)          (in)          )
20   format(i4,2(f7.1,1h,,13),1x,2(f7.2,1h,13),f7.1,1h,,13,f6.3,1h,,13)
      write (6,20)
      write (6,17)
      write (6,18)
      write (6,19)
      do 120 i=1,n
      nn=5*n+i
      in2=i+2*n
120  write (6,20) i,ex(i+n),jex(i+n),ex(i),jex(i),(ex(j),jex(j),
1    j=in2,nn,n)
      return
      end
      subroutine piel (mp,n,xp,ap,ep,rp,stp,pm,alpha,iout,iw,icol,
1    ipel,area)
      dimension xp(20),ap(20),ep(20),rp(20),stp(99),pm(99),alpha(99)
1    ,area(99)
      salph=0.0
      do 101 il=1,n
101  salph=salph+alph(il)
c
c      element counter i2, cross section index i3
c
      if (ipel.eq.2) go to 108
      i2=1

```

```

      i3=2
      ij=i3-1
      xt=0.0
      xb=(xp(mp)*alph(i2))/salph
      dx = xb
      at=ap(1)
      rpt=rp(1)
      ept=ep(1)
      go to 110
c
c      repeat 102 loop for every segment i2
c
102  xt=xb
      dx = (xp(mp)*alph(i2))/salph
      xb = xt + dx
110  stp(i2)=10.e12
      pm(i2)=0.0
      area(i2)=0.0
c
c      repeat 103 loop until next cross section depth is below segment
c
103  if (xb.lt.xp(i3)) go to 107
      x1=xp(i3)-xt
      xt=xp(i3)
      aao=(at+ap(i3))/2.
      at=ap(i3+1)
      rpo=(rpt+rp(i3))/2.
      rpt=rp(i3+1)
      epo=(ept+ep(i3))/2.
      ept=ep(i3+1)
      if (x1.lt.10.e-8) go to 105
      area(i2) = area(i2)+ aao*x1
      summ=rpo*x1*aao/(144000.0*32.17)
      sumk=aao*epo/x1
      pm(i2)=summ+pm(i2)
      stp(i2)=(sumk*stp(i2))/(sumk+stp(i2))
105  i3=i3+1
      if (i3.gt.mp) go to 118
      ij=i3-1
      go to 103
107  x1=xb-xt
c
c      for bottom portion of segment
c
      xx = xp(i3) - xp(ij)
      xf = 1.0
      if (xx.gt.0.0) xf = (xb-xp(ij))/xx
      aaup=ap(ij)+(ap(i3)-ap(ij))*xf
      aao=(at+aaup)/2.
      at=aaup
      rpu = rp(ij)+(rp(i3)-rp(ij))*xf
      rpo=(rpt+rpu)/2.
      rpt=rpu

```

```

    epu = ep(ij)+(ep(i3)-ep(ij))*xf
    epo=(epu+ept)/2.
    ept=epu
    if (x1.lt.10.e-8) go to 118
    area(i2) = area(i2)+aao*x1
    summ=rpo*x1*aao/(144000.0*32.17)
    sumk=aao*epo/x1
    stp(i2)=(sumk*stp(i2))/(sumk+stp(i2))
    pm(i2)=summ+pm(i2)
118  area(i2) = area(i2)/dx
    if (i2.eq.n) go to 108
    i2=i2+1
c
c    next segment
c
    go to 102
108  continue
c
c    print given information
c
111  format (20x,16hpile description/)
112  format (16h x bel. top (ft),13f8.1)
113  format (16h a (sq. in.)      ,13f8.1)
114  format (16h e (ksi)          ,13f8.0)
115  format (17h gamma (lb/cu ft) ,f7.1,12f8.1)
    write (6,111)
    max=6
    if (icol.eq.132) max=13
    i4=1
    i5=max
    if (i5.gt.mp) i5=mp
116  write (6,112) (xp(i),i=14,i5)
    write (6,113) (ap(i),i=14,i5)
    write (6,114) (ep(i),i=14,i5)
    write (6,115) (rp(i),i=14,i5)
    if (i5.ge.mp) go to 117
    i4=i5+1
    i5=i4+max
    if (i5.gt.mp) i5=mp
    go to 116
117  continue
    if (ipel.ne.2) return
c
c    determine average areas for the case where masses and
c    stiffnesses were given
c
    d1 = 0.0
    do 145 i=1,n
    do 146 j=2,mp
    if (xp(j).gt.d1) go to 147
146  continue
147  area(i)=ap(j-1)
145  d1 = d1 + alph(i)*xp(mp)/salph

```

```

return
end
subroutine diesel
common vinh,dinh,aoh,voh,doh,anh,vnh,dnh,sth,hm,am,sta,ma,esoil,
1 aop,anp,dop,dnp,vop,vnp,stp,pm,cdp,res,reso,sj,sok,qs,su,
2 splice,depib,aram,ecus,ecap,ept,eenv,vfall,vcham,xpt,sjjj,
3 emp,rho,dth,ntp,n,m,jrat,iult,rult,soilm,dsacp,
4 fpto,iw,pcom,tdel,iout,out,expp,temi,temo,inp,stroko,strokn,vo,po
5 ,dfin,jdout,jpmax,ex,icol,fintmx,temax,result,ismith,iter
6 ,ityph,ipreig,vfallm,strmax,dcyl,pbar,area,jex
dimension vinh(10),dinh(10),aoh(10),voh(10),doh(10),anh(10),
1 vnh(10),dnh(10),sth(10),hm(10),vnp(99),
2 aop(99),anp(99),dop(99),dnp(99),vop(99),stp(99),pm(99),cdp(99),
3 res(99),reso(99),sj(99),sok(99),qs(99),su(99),splice(99),
4 dsacp(3),out(3200),inp(13),ex(600),result(100),jex(600)
5 ,aoh(99),voah(99),doah(99),area(99),delah(99),sta(10),am(10)
6 ,pbar(20),pmi(99),soi(99),hmi(10)
klim = 200
c klim ... max no. of values of one variable in output,
c klim*15=dim(out)
expt = 1.-1./expp
cdh = pm(99)
c
c set up output variables
c
iout = iabs(iout)
if (iout.ge.20) iout=iout-20
if (iout.ge.10) iout=iout-10
scj = 0.0
iai=0
patm=2.117
dt2=ntp*.5
dt6=ntp*ntp/6.0
dh2=dth*.5
dh6=dth*dth/6.0
jout=0
jdout = 6.0*xpt/(sqrt(emp/rho)*ntp)
jdout = jdout / 100
tjd = float(jdout)*ntp*1000.
if (tjd.gt.0.5) jdout = (0.0005/ntp) + 0.5
if (jdout.lt.1) jdout=1
if (iout.eq.6) jdout = jdout + 1
c jmax ... number of output time to analysis pile time increment
jmax = 6
if (icol.gt.0) jmax=13
if (jmax.gt.n) jmax=n
c jmax ... number of pile variables printed (dependent on form
c size)
c
c initialize
c
fintmx=0.5
ts = 0.0005

```

```

if (pbar(9).gt.0.0) ts = sjjj
if (pbar(9).gt.0.0.and.ts.lt.10.e-8) ts = 0.01
red=0.90
if (emp.lt.3000000.) red = 0.0
kk=0
nl=n+1
nml=n-1
ftpo = fpto
ftit = fpto
ftpoa =ftpo
tlim = 0.0
fanvoa = 0.0
fcuoa = vo
vnpit = vop(n)
vnhit = vinh(m)
vo = vcham + (dinh(m+1)-dinh(m))*aram
timp = 0.0
tt=0.0
ti=0.0
tblow = -1.0
rimp = 1.
ridel = 1.
pcomr = pcom
if (pbar(19).gt.0.0) go to 801
pcomf = 0.5*(stroko-depib)/depib
c reduction due to poor scavenging
if (pcomf.gt.1.0) pcomf = 1.0
if (pcomf.lt.0.5) pcomf = 0.5
pcom=1.0+(pcom-1.0)*pcomf
801 continue
pcoml=pcom
ifirst=0
irem = 0
vlpmax = 0.0
dnpmax = 0.0
mml=m-1
ml=m+1
m2=m+2
g=32.17
enthru = 0.0
emax = 0.0
t21c = 2000.*xpt*sqrt(rho/emp)
c limit of main loop
nt = 0.005*t21c/dtp
ooa=1./144
oojrt=1./float(jrat)
dsil=1./dsacp(1)
dsi2=1./dsacp(2)
dsi3=1./dsacp(3)
dtpm=dtp*1000.
jck = nt
if (float(nt)*dtp.lt.0.050) nt = 0.050/dtp
if(temax.gt.1.0) nt=temax/dtpm

```

```

c      loading and rebound stiffnesses for pile top, anvil, and capblock
c
c      stpt1 = sth(m2)*stp(1)/(sth(m2)+stp(1))
c      stpt2 = (sth(m2)/ecus)*(stp(1)/ept)/((sth(m2)/ecus)+stp(1)/ept)
c      stal = sth(m)
c      star = stal/eanv
c      stcl = sth(m1)
c      stcr = stcl/ecap
c      hammer variables to initial values
c      do 100 i=1,m2
c      hmi(i)=1./hm(i)
c      voh(i)=vinh(i)
c      doh(i)=dinh(i)
c      voah(i)=vinh(i)
c      doah(i)=dinh(i)
c      delah(i) = 0.0
c      aoah(i)=g
100    continue
c      convergence criteria
c      epsv = 0.02
c      epsf = 1.0
c      do 97 i=1,n
c      pmi(i)=1./pm(i)
c      soi(i)=sok(i)/esoil
97     aop(i) = g
c      pmi(n)=1./(pm(n)+soilm)
c      soi(n1)=sok(n1)/esoil
c      do 450 i=1,3200
450    out(i)=0.0
c      extremum array
c      do 460 j=1,6
c      do 460 i=1,n
c      ij=n*(j-1)+i
c      ex(ij) = 0.0
c      jex(ij) = 0
460    continue
c      acyl = g
c      pbr = 0.144*14.7*pbar(1)
c      if (pbar(1).lt.10.e-8) pbr = 2.117*aram
c      delp = 0.0
c      if (pbar(1).gt.aram) delp = 2.117*(pbar(1)-aram)
c      if (pbar(20).gt.0.0) pbr = 0.0
c      cylinder variables
c      if(pbar(8).gt.0.0) cylmi=g/pbar(8)
c      dcyl = doh(m1)
c      vcyl = voh(m1)
c      ignition variables
c      tslow = 1.0
c      tsa=1.0/pcom
c      tsb=(1.0-tsa)/ts
c      contact area variables
c      afa=(3.-fintmx)/2.

```



```

afb=(1.-fintmx)/(2.*dsacp(1))
exp = 1.35
if (iout.lt.0) write (6,653)
c
c start main loop
c
do 700 j=1,nt
if (j.gt.jck) jck=-1
if (vlpmax.lt.vnp(1)) vlpmax = vnp(1)
if (red.lt.0.01) go to 103
if (vnp(1).gt.0.9*vlpmax.or.vlpmax.lt.5.0) go to 103
stpt1 = stpt2 * red
red = 0.0
103 continue
dpos = doh(m)-dcyl
c force on top of ram for regular closed end or vac hammer
ptr = ftr(dpos,pbar,iwt)
dbc = -doh(m)
if (pbar(20).gt.0.0) call vacham(pbar,aram,depib,dcyl,dbc,ptr,
1 vfall,iout,iw)
if(pbar(8).gt.0.0)acyl=g+(pbr-ptr)*cylmi
if(pbar(20).gt.0.0)ptr=ptr+pbr
if (pbar(20).lt.10.e-8) ptr = ptr-delp
dcyl = dcyl + vcyl * dtp
vcyl = vcyl + acyl * dtp
if (dcyl.gt.doh(m1)) vcyl = voh(m1)
if (dcyl.gt.doh(m1)) dcyl = doh(m1)
tt=tt+dtpr
c absolute time counter (millisec.)
c time after impact (timp)
c rimp=0.0 means first impact has occurred
if (rimp.gt.0.5) timp=(doh(m)-doh(m1))/voh(m)
if (rimp.lt.0.5) timp=timp+ntp
ti=ti+ntp*(1.0-pcom/pcom1)
c ignition start for atomized fuel
if (pbar(9).gt.0.0.and.rimp.lt.0.5) ridel = 0.0
if (timp.ge.tdel.and.ipreig.eq.0) ridel = 0.0
if (j.gt.0.and.j.eq.ipreig) ridel = 0.0
if (ridel.gt.0.5.or.pcom.lt.0.01) go to 470
temo=temo*pcom
po=po*pcom
exp = exp
pcom = 0.0
470 continue
if (tt.lt.50.) go to 471
if (ddc.lt.12.*depib) go to 471
c for ram reaching ports
if(tblow.lt.0.0)pblow=(pn-patm)*100.*ntp
tblow = 1.0
pn = pn - pblow
if(pn.lt.patm)pn=patm
po = pn
471 continue

```

```

        it=0
c
c      overall loop starts
c
800    continue
        fcuo = fcua
        ftpo = ftpoa
        fanvo = fanvoa
        do 99 i=1,m2
            voh(i)=voah(i)
            aoh(i)=aoah(i)
99      doh(i)=doah(i)
c
c      hammer analysis
c
        do 600 jh=1,jrat
            ith=0
c      prediction
            do 585 i=1,m2
                if (it.eq.0) anh(i)=aoh(i)
                if (it.gt.0) anh(i)=aoh(i)+float(jh)*delah(i)
585      continue
            do 101 i=1,m2
101      call integr(dth,aoh,anh,voh,doh,vnh,dnh,i,dh2,dh6)
c
c      hammer cycle starts
c
500    continue
        i=1
        anh(1)=g+((dnh(2)-dnh(1))*sth(1)+ptr)*hmi(1)
        call integr(dth,aoh,anh,voh,doh,vnh,dnh,i,dh2,dh6)
        do 102 i=2,mml
            il=i+1
            iml=i-1
            anh(i)=g+((dnh(il)-dnh(i))*sth(i)+(dnh(iml)-dnh(i))
1            *sth(iml))*hmi(i)
            call integr(dth,aoh,anh,voh,doh,vnh,dnh,i,dh2,dh6)
102    continue
c      now check displacement of ram and impact block
        v = vcham + aram*(dnh(ml)-dnh(m))
        if (v.lt.vcham) v = vcham
        pn = po*(vo/v)**exp
        temn = temo*(pn/po)**expt
c      last ram element
        ddis = dnh(ml)-dnh(m)
        if (pcom.lt.0.1) tslow=tsa+tsb*ti
        if (tslow.gt.1.0) tslow = 1.0
        pe = pn*tslow
        afac = 1.0
        if (ddis.lt.-dsacp(1)) afac=afa+afb*ddis
        if (ddis.lt.-3.*dsacp(1)) afac=fintmx
        fp = pe*aram*afac
        fanv = 0.0

```

```

      if (ddis.gt.0.0) go to 811
      dvel = vnh(m1) - vnh(m)
c
      ddiso = doh(m1)-doh(m)
      ddd = ddis - ddiso
c
      anvil spring force always tension= +
      call stiff(ddis,dvel,stal,star,fanvo,fanv,dsil,ddd)
      if (fanv.gt.0.0) fanv = 0.0
811  fintac = -fp+fanv
      ddc=ddis*12.0
      if (ddis.gt.0.0) irem = 0
      if (ddis.lt.0.0) rimp = 0.0
      if (ddis.lt.0.0.and.irem.ne.1) ifirst = 1
c
      ifirst = 1 means impact has occurred the first time
      if (ifirst.ne.1) go to 675
      timp=0.0
      jck=0.0015*t21c/dtp
c
      irem = 1
      tlim = tt + 1.15*t21c
      if (tlim.lt.10.) tlim = 10.
675  continue
      anh(m)=g+((dnh(mml)-dnh(m))*sth(mml)+fintac)*hmi(m)
      call integr(dth,aoh,anh,voh,doh,vnh,dnh,m,dh2,dh6)
c
      anvil
      fintac = -fintac
      dha = 0.0
      ddis = dnh(m2)-dnh(m1)
      fcu = 0.0
      if (ddis.gt.0.0) go to 812
      dvel = vnh(m2)-vnh(m1)
      dha = dvel*cdh
c
      ddiso = doh(m2)-doh(m1)
      ddd = ddis - ddiso
      fcuo = -fcuo
      call stiff(ddis,dvel,stcl,ster,fcuo,fcu,dsi2,ddd)
      if (fcu.gt.0.0) fcu=0.0
      fcuo = -fcuo
812  continue
      anh(m1)=g+(fintac+fcu+dha)*hmi(m1)
      call integr(dth,aoh,anh,voh,doh,vnh,dnh,m1,dh2,dh6)
c
      cap
576  continue
      dtl = float(jh)*dth
      i = 1
      if(it.gt.0.and.jrat.eq.1)go to 575
      call integr(dth,aop,aop,vop,dop,vnp,dnp,1,dh2,dh6)
      if(jrat.eq.1)go to 575
      dtl=float(jh)*dth
      dnpl=dop(1)+(dtl/dtp)*(dnp(1)-dop(1))
      vnpl=vop(1)+(dtl/dtp)*(vnp(1)-vop(1))
      if(it.gt.0)go to 576

```

```

    dl2=dt1*.5
    dl6=dt1*dt1/6.0
    call integr(dt1,aop,aop,vop,dop,vnp,dnp,l,dl2,dl6)
575  continue
    dnp1=dnp(1)
    vnpl=vnp(1)
    ddis = dnp1-dnh(m2)
    vcap = 0.0
c
    dvel = 0.0
    ftp = 0.0
    if (ddis.gt.0.0) go to 813
    dvel = vnpl-vnh(m2)
    vcap = vnh(m2)
c
    ddiso = dop(1)-doh(m2)
    ddd = ddis-ddiso
    ftpo = -ftpo
    call stiff(ddis,dvel,stpt1,stpt2,ftpo,ftp,ds13,ddd)
    if (ftp.gt.-splice(1)) ftp = -splice(1)
    ftpo = -ftpo
813  fcu = -fcu
    anh(m2)=g+(fcu+ftp+dvel*cdp(1)-dha)*hmi(m2)
    call integr(dth,aoh,anh,voh,doh,vnh,dnh,m2,dh2,dh6)
    ith = ith +1
    ftp = -ftp
c
c    if dth eq dtp (jrat=1) the following is skipped (one time
c    increment)
c
    if (jrat.eq.1) go to 650
    if (ith.gt.iter) go to 650
    err = abs(vnhit-vnh(m))
    vnhit = vnh(m)
    if (err.gt.epsv) go to 500
650  continue
    do 501 i=1,m2
    delah(i)=aoh(i)+(anh(i)-aoh(i))*oojrt
    aoh(i)=anh(i)
    voh(i)=vnh(i)
501  doh(i)=dnh(i)
    temo = temn
    po = pn
    vo = v
    fanvo = fanv
    ftpo = ftp
    fcuo = fcu
600  continue
c
c    pile analysis
c
    call pilean (ipt,aop,vop,dop,anp,vnp,dnp,stp,pm,splice,sj,cdp,res
1  reso,su,sok,vnpit,vcap,soilm,scj,dtp,n,it,ismith,epsv,iai,esoil,

```

```

2 ftp,jrat,iter,soi,dt2,dt6,pmi,rsum,dsum)
c
c   check for convergence
c
    it = it + 1
    if (it.gt.iter) go to 850
    errv = abs(vnpit-vnp(n))
    if (jrat.eq.1) vnpit = vnp(n)
    if (jrat.eq.1.and.errv.gt.epsv) go to 800
    err = abs(ftit-ftp)
    ftit = ftp
    if (err.gt.epsf) go to 800
850 continue
c   prepare output for current
c   time increment and
c   set up starting values
c   for next step
c
    vmaxp = 0.0
    do 401 i=1,n1
    reso(i) = res(i)
    aop(i) = anp(i)
    vop(i)=vnp(i)
    if (abs(vop(i)).gt.vmaxp) vmaxp = abs(vop(i))
401 dop(i)=dnp(i)
    if (dnp(n).gt.dnpmax) dnpmax = dnp(n)
    fanvoa = fanvo
    fcua = fcua
    ftpoa = ftpo
    do 402 i=1,m2
    aoah(i)=anh(i)
    voah(i)=vnh(i)
402 doah(i)=dnh(i)
    dopinc = dop(1)*12.
    den = vop(1)*dtp*ftp
c   transferred energy
    enthru = enthru + den
    if (enthru.gt.emax) emax = enthru
    jout = jout + 1
    ftpo = ftp
    if (jout.ne.jdout) go to 699
    jout = 0
    pt=ptr/pbar(1)
    no2 = n/2
c
    pq=pe*ooa
652 format (1h,i4,4i2,f6.1,f8.3,2f6.1,2f8.1,2f6.1,2(f6.1,f8.1),3f8.1,
1 f7.3,f6.0)
    do 686 i=no2,nml
    if(i.ne.no2.and.i.ne.nml)go to 686
    il = i+1
    com = dnp(i)-dnp(il)
    f = com*stp(il)

```

```

        if (f.lt.splice(11)) f = splice(11)
        if (f.gt.0.0.or.splice(11).lt.-.5) go to 685
        f = 0.0
        if (com.lt.splice(11)) f = (com-splice(11))*stp(11)
685    fopn = f
        if (i.eq.no2) fop5 = f
686    continue
c    debug output
653    format
1      (132h      jp f t h p      time      dcham      vru      van      fint      fpt
a      vl      en      v5      f5      vn      fn      cp      rs      da
b      d      t      )
        if (iout.lt.0) write (6,652)
a      j,ifirst,it,ith,ipt,tt,ddc ,vnh(m),vnh(m1),fintac,ftp,vop(1),
1      enthru,vop(no2),fop5,vop(n),fopn,pq,rsum,dsum,dopinc,pt
        if (iout.lt.6) call fill(kk,klm,out,jmax,iout,dnh,sth,
1      fintac,ftp,inp,dnp,stp,splice,vnh,vnp,anh,anp,jpmax,maxjp,ex,area,
2      j,m,n,jex,jck)
c      jpmax is the latest j
c      fills in the output array out(klm*15)
c      maxjp is the maximum number of values stored of one variable
c      maxjp equals approx. jpmax/jdout
        if (iout.eq.6) call fill2(n,m,out,pq,dnh,ftp,vnp,dnp,fop5,sj,
1      rsum,dsum,res,fopn,kk,icol,tt,ismith,j,maxjp,jpmax,stp,area,
2      splice,ex,jex,j,jck)
c    check whether analysis was carried out sufficiently long
        if (temax.gt.1.0) go to 699
        if (vlpmax*0.20.lt.vmaxp) go to 699
        if (tt.lt.tlim) go to 699
        if (tt.lt.t2lc.or.tt.lt.20.0) go to 699
        if ((dnh(m1)-dnh(m)).gt..1*depib.and.dnp(n).lt.dnpmax*0.9) go to
1      701
        if ((dnh(m1)-dnh(m)).gt..2*depib.and.dnp(n).lt.0.98*dnpmax) go to
1      701
699    ifirst = 0
700    continue
701    continue
        fintmx = float(j)*dtp
        dfin = (dnpmax-qs(n))*12.
        vnh(10)=vcyl
        dnh(10) = acyl
        if (iout.lt.0) write (6,654) enthru,emax
        anp(99) = emax
654    format (30h0transferred energy, fin- max=,f6.1,2h -,f6.1,5h k-ft)
        pcom = pcomr
        temo = temn
        return
    end
    subroutine down (strmax,vfallm,pbar,hm,depib,m,iw,iout,t)
    dimension pbar(1),hm(1)
    t=0.0
    art=pbar(1)
    str = strmax

```



```

patm=14.7*0.144
pmax=patm+pbar(8)/pbar(1)
prat=(patm/pmax)**(1.0/pbar(4))
vbin = pbar(6)*pbar(1) + pbar(5)
dup=(vbin/pbar(1))*(1.0-prat)
dmax=pbar(6)-dup
dsf = pbar(2)
c dup = distance from bounce chamber vents
c dmax = distance from top
c dpos = distance from anvil (negative)
if (dmax.gt.dsfc) go to 202
c max stroke penetrates into safety chamber
prat = (vbin/(pbar(5)+dsf*art))**pbar(4)
prat = (prat*patm/pmax)**(1./pbar(4))
dmax = dsf*prat
202 continue
dpos = dmax - pbar(7)
ram = 0.0
do 200 i = 1,m
c --- delete original source code
200 ram = ram + hm(i)
c
strmax = -dpos
if (str.gt.strmax) str = strmax
c start with ram at position where uplift is imminent
totd = -dpos-depib
if (str.lt.0.1) go to 300
totd = str-depib
dpos = -str
strmax = str
300 continue
nst = totd/0.05
dfall = totd/float(nst)
g = 32.17
ao = g
vo = 0.0
do 100 i=1,nst
a = g+(ftr(dpos,pbar,iwt)-patm*art)/ram
va = vo/a
delt = sqrt(va*va+2.*dfall/a)-va
t = t+delt
v = vo + 0.5*(a+ao)*delt
dpos = dpos + dfall
vo = v
100 ao = a
vfallm = v
if (str.gt.0.1.and.iout.lt.0 ) write (6,2) str,v
2 format (20h stroke (less max.)=,f6.1,23h ft, velocity at ports=,
1f6.1,5h ft/s )
if (str.gt.0.1) return
if (iout.lt.0) write (6,1) strmax,vfallm
1 format(28h maximum stroke (no uplift)=,f6.2,30h ft maximum veloci
ly at ports=,f6.2,5h ft/s)

```

```

      return
    end
    subroutine fill(kk, klim, out, jmax, iout, dnh, sth, fintac, ftp, inp,
1 dnp, stp, splice, vnh, vnp, anh, anp, jpm, maxjp, ex, area, j, m, n, jex, jck
      dimension out(1), dnh(1), sth(1), inp(1), dnp(1), stp(1),
1 splice(1), vnh(1), vnp(1), anh(1), anp(1), ex(1), area(1), jex(1)
      kk = kk + 1
      m1 = m+1
      kkmax = (jmax+1)*klim + kk
c
c      check if out is full
c
      if (kk.gt.klim) go to 459
      k2 = kk + klim
      k3 = k2 + klim
      k4 = k3 + klim
      l = 1
      if (iout.ne.1.and.iout.ne.3) go to 452
c
c      force assignment
c
      out(kk) = 0.0
      if (m.gt.1) out(kk)=(dnh(m-1)-dnh(m))*sth(m-1)
      out(k2) = fintac
      out(k3) = ftp
      l = 2
      kkl = inp(2)
      do 451 i = k4, kkmax, klim
      kml = kkl-1
      com = dnp(kml)-dnp(kkl)
      out(i) = com*stp(kkl)
      if (out(i).lt.splice(kkl)) out(i) = splice(kkl)
      if (out(i).gt.0.0.or.splice(kkl).lt.-.5) go to 686
      out(i) = 0.0
      if (com.lt.splice(kkl)) out(i) = (com-splice(kkl))*stp(kkl)
      if (out(i).lt.0.0.and.out(i).gt.-.5) out(i) = 0.0
686 continue
      l = l+1
451 kkl = inp(1)
      go to 458
452 if (iout.ne.2) go to 454
c
c      velocity assignment
c
      out(kk) = vnh(m)
      out(k2) = vnh(m1)
      do 453 i=k3, kkmax, klim
      kkl = inp(1)
      out(i) = vnp(kkl)
453 l = l + 1
      go to 458
454 if (iout.ne.5) go to 456
c

```

```

c      displacement assignment
c
      out(kk) = dnh(m)
      out(k2) = dnh(m1)
      do 455 i=k3, kkmax, klim
      kkl = inp(1)
      out(i) = dnp(kkl)
455    l = l+1
      go to 458

c
c      acceleration assignment
c
456    out(kk) = anh(m)
      out(k2) = anh(m1)
      do 457 i=k3, kkmax, klim
      kkl = inp(1)
      out(i) = anp(kkl)
457    l = l+1
458    jpmax = j
      maxjp = kk
459    continue

c
c      calculate the extremum array ex(n*6)
c      first forces and stresses
c
      sexl = ftp/area(1)
      if (ftp.gt.ex(1)) jex(1)=j
      if (ftp.gt.ex(1)) ex(1) = ftp

c
c      jck less 0 means no minimum force or minimum stress check
c
      if (jck.lt.0) go to 105
      if (ftp.lt.ex(n+1)) jex(n+1)=j
      if (ftp.lt.ex(n+1)) ex(n+1) = ftp
      if (sexl.lt.ex(2*n+1)) jex(2*n+1)=j
      if (sexl.lt.ex(2*n+1)) ex(2*n+1)=sexl
105    continue
      if (sexl.gt.ex(3*n+1)) jex(3*n+1)=j
      if (sexl.gt.ex(3*n+1)) ex(n*3+1)=sexl
      do 461 i=2, n
      com = dnp(i-1)-dnp(i)
      felp = com*stp(i)
      if (felp.lt.splice(i)) felp = splice(i)
      if (felp.gt.0.0.or.splice(i).lt.-.5) go to 685
      felp = 0.0
      if (com.lt.splice(i)) felp = (com-splice(i))*stp(i)
      if (felp.lt.0.0.and.felp.gt.-.5) felp = 0.0
685    continue
      selp = felp/area(i)
      if (felp.gt.ex(i)) jex(i)=j
      if (felp.gt.ex(i)) ex(i) = felp
      if (jck.lt.0) go to 103
      if (felp.lt.ex(n+1)) jex(n+1)=j

```

```

      if (felp.lt.ex(n+1)) ex(n+1) = felp
      if (selp.lt.ex(2*n+1)) jex(2*n+1)=j
      if (selp.lt.ex(2*n+1)) ex(2*n+1) = selp
103  continue
      if (selp.gt.ex(3*n+1)) jex(3*n+1)=j
461  if (selp.gt.ex(3*n+1)) ex(3*n+1) = selp
c
c    velocity and displacement extrema
c
      do 462 i=1,n
      if (vnp(i).gt.ex(4*n+1)) jex(4*n+1)=j
      if (vnp(i).gt.ex(4*n+1)) ex(4*n+1) = vnp(i)
      disi = dnp(i) * 12.
      if (disi.gt.ex(5*n+1)) jex(5*n+1)=j
462  if (disi.gt.ex(5*n+1))ex(5*n+1) = disi
      return
      end
      subroutine fill12 (n,m,out,pq,dnh,ftp,vnp,dnp,fno5,sj,rsum,dsum,
1  res,ftn,kk,icol,tt,ismith,j,maxjp,jpmax,stp,area,splice,ex,jex,
2  jl,jck)
      dimension out(3200),dnh(10),vnp(99),dnp(99),res(99),sj(99),
1  splice(99),stp(99),area(99),ex(600),jex(600)
      nl=n+1
      ml=m+1
      no5=n/2
      kk=kk+1
      klim=200
c
c    check if out is full
c
      if (kk.gt.klim) go to 102
      k1=kk+klim
      k2=k1+klim
      k3=k2+klim
      k4=k3+klim
      k5=k4+klim
      k6=k5+klim
      k7=k6+klim
      k8=k7+klim
      k9=k8+klim
      k10=k9+klim
      k11=k10+klim
      k12=k11+klim
      k13=k12+klim
      k14=k13+klim
      k15=k14+klim
c
c    assign variables to out
c
      out(kk)=tt
      out(k1)=pq
      out(k2)=dnh(m)*12.
      out(k3)=dnh(ml)*12.

```

```

out(k4)=ftp
out(k5)=vnp(1)
out(k6)=dnp(1)*12.
out(k7)=fno5
out(k8)=vnp(no5)
out(k9)=dnp(no5)*12.
out(k10)=rsum
out(k11)=dsum
rtoe = sj(n1)*vnp(n)
if (rtoe.lt.0.0001) rtoe=0.
if (ismith.eq.1) rtoe=rtoe*abs(res(n1))
out(k12)=res(n1)+rtoe
out(k13)=ftn
out(k14)=vnp(n)
out(k15)=dnp(n)*12.
if (icol.ne.0) go to 101
out(k7)=out(k13)
out(k8)=out(k14)
out(k9)=out(k15)
101 jpmax=j
maxjp=kk
c calculate the extremum array ex(n*6)
102 continue
sexl=ftp/area(1)
if (ftp.gt.ex(1)) jex(1)=j1
if (ftp.gt.ex(1)) ex(1)=ftp
c
c jck less 0 means no minimum force and stress check
c
if (jck.lt.0) go to 105
if (ftp.lt.ex(n+1)) jex(n+1)=j1
if (ftp.lt.ex(n+1)) ex(n+1)=ftp
if (sexl.lt.ex(2*n+1)) jex(2*n+1)=j1
if (sexl.lt.ex(2*n+1)) ex(2*n+1)=sexl
105 continue
if (sexl.gt.ex(3*n+1)) jex(3*n+1) = j1
if (sexl.gt.ex(3*n+1)) ex(3*n+1) = sexl
do 103 i=2,n
com = dnp(i-1)-dnp(i)
felp = com*stp(i)
if (felp.lt.splice(i)) felp=splice(i)
if (felp.gt.0.0.or.splice(i).lt.-.5) go to 685
felp = 0.0
if (com.lt.splice(i)) felp = (com-splice(i))*stp(i)
if (felp.lt.0.0.and.felp.gt.-0.5) felp=0.0
685 continue
selp=felp/area(i)
if (felp.gt.ex(i)) jex(i)=j1
if (felp.gt.ex(i)) ex(i)=felp
if (jck.lt.0) go to 106
if (felp.lt.ex(n+i)) jex(n+i)=j1
if (felp.lt.ex(n+i)) ex(n+i)=felp
if (selp.lt.ex(2*n+i)) jex(2*n+i)=j1

```



```

    if (selp.lt.ex(2*n+1)) ex(2*n+1)=selp
106 continue
    if (selp.gt.ex(3*n+1)) jex(3*n+1)=jl
103 if (selp.gt.ex(3*n+1)) ex(3*n+1)=selp
    do 104 i=1,n
        if (vnp(i).gt.ex(4*n+1)) jex(4*n+1)=jl
        if (vnp(i).gt.ex(4*n+1)) ex(4*n+1)=vnp(i)
        disi=dnf(i)*12.
        if (disi.gt.ex(5*n+1)) jex(5*n+1)=jl
104 if (disi.gt.ex(5*n+1)) ex(5*n+1)=disi
    return
end
function ftr (dpos,pbar,iwt)
dimension pbar (20)
iwt=0
patm=14.7*0.144
art=pbar(1)
dsf=pbar(2)
expb=pbar(4)
vct=pbar(5)
depbb=pbar(6)
dbbt=pbar(7)
rwh=pbar(8)+patm*art
dbc=pbar(7)+dpos
vbin = depbb*art + vct
if (dbc.gt.depbb.or.dbbt.lt.0.1) ftr=patm*art
if (dbc.gt.depbb.or.dbbt.lt.0.1) return
vbfin=dbc*art+vct
vsf=dsf*art
if (dbc.lt.dsf) vbfin=vsf+vct
ftr=patm*art*(vbin/vbfin)**expb
if (dbc.lt.dsf) ftr=ftr*(vsf/(art*dbc))**expb
if (ftr.gt.rwh) iwt=1
if (ftr.gt.rwh) ftr=rwh
return
c art ... area at top of ram
c dsf ... distance of safety chamber from top
c expb ... exponent
c vct ... volume of compression tank
c depbb ... distance of exhaust ports from top
c rwh ... weight of cylinder etc.
c dbbt ... distance between top and bottom - ram length
c vbin ... initial volume of bounce chamber
c vbfin ... final volume of bounce chamber
c dbc ... current position ... distance from top
end
subroutine hael (ihamr,dtrh,dtrc,hname,ir)
c
common vinh,dinh,aoh,voh,doh,anh,vnh,dnh,sth,hm,am,sta,ma,esoil,
1 aop,anp,dop,dnp,vop,vnp,stp,pm,cdp,res,reso,sj,sok,qs,su,
2 splice,depib,aram,ecus,ecap,ept,eant,vfall,vcham,xpt,sjjj,
3 emp,rho,dth,ntp,n,m,jrat,iult,rult,soilm,dsacp,
4 fpto,iw,pcom,tdel,iout,out,expp,temi,temo,inp,stroko,strokn,vo,

```



```

5 ,dfin,jdout,jpmax,ex,icol,fintmx, temax,result,ismith,iter
6 ,ityph,ipreig,vfallm,strmax,dcyl,pbar,area,jex
   dimension vinh(10),dinh(10),aoh(10),voh(10),doh(10),anh(10),
1  vnh(10),dnh(10),sth(10),hm(10),vnp(99),
2  aop(99),anp(99),dop(99),dnp(99),vop(99),stp(99),pm(99),cdp(99),
3  res(99),reso(99),sj(99),sok(99),qs(99),su(99),splice(99),
4dsacp(3),out(3200),inp(13),ex(600),result(100),jex(600),
istore(1,26,2),sta(10),am(10),hname(2),pbar(20),
1 area(99)
   if=8
12 format (2a4,9f8.2)
   2 format (10f8.2)
   ihamrr = ihamr
   do 107 i=1,20
   am(1) = 0.0
107  pbar(1)=0.0
   convs = 12.0
   if (ihamr.eq.0) go to 50
c   for ihamr = 0 read from cards, otherwise read from file
   do 52 i=1,ihamr
52  read (if,54) ((store(1,j,k),k=1,2),j=1,26)
54  format (2a4/10(5e15.5/))
   rewind if
   go to 51
50  continue
   read (ir,12) ((store(1,i,j),j=1,2),i=1,5),
1  store (1,6,1)
   read (ir,2) store(1,6,2), ((store(1,i,j),j=1,2),
1  i=7,25),store(1,26,1)
c
51  continue
   ihamrr = ihamr
   ihamr = 1
   ityph=store(1,26,1)+.1
   if(ityph.lt.1)ityph=1
   ma = store(ihamr,25,2)
   m=store(ihamr,8,2)
   if (m.lt.1) write (6,10) ihamrr
10  format (33h insufficient hammer info. ihamr=,i3)
   if (m.lt.1) stop
   m1=m+1
   m2=m+2
   sth(m1)=sth(8)
   sth(m2)=sth(9)
   hm(m2)=hm(9)
   do 100 il=1,2
100  hname(il)=store(ihamr,1,il)
   hm(m1)=store(ihamr,9,1)/32.17
   sth(m)=store (ihamr,9,2)*convs
   if (sth(m1).lt.0.1) sth(m)=2.0*sth(m)
   if (sth(m1).lt.0.1) sth(m1)=sth(m)
   if (sth(m2).lt..1) sth(m2)=10.0**12
   stost=sth(m2)

```

```

    sth(m2)=stost*stp(1)/(stost+stp(1))
    if(ityph.ne.3)go to 104
    hm(m1)=hm(m2)
    sth(m)=sth(m1)
    sth(m1)=sth(m2)
    m2=m1
    if(sth(m).lt.1.e-2)sth(m)=hm(m1)*.5e8
104  continue
    do 102 i1=1,m2
    i2=i1-1
    if (i1.eq.m) hm(i1)=store(ihamr,i1+1,1)/32.17
    if (i1.ge.m) go to 101
    hm(i1)=store(ihamr,i1+1,1)/32.17
    sth(i1)=store (ihamr,i1+1,2)*convs
101  if (i1.eq.1) i2=1
    do 102 i3=i2,i1
    tcrh=hm(i1)/sth(i3)
    if (i3.eq.1) dtcrh=tcrh
    if (tcrh.lt.dtcrh) dtcrh=tcrh
102  continue
    dtcrh=sqrt(dtcrh)
    dtcra=1000.0
    sth(m2)=stost
    if (ma.eq.0) go to 105
    do 103 j1=1,ma
    j2=j1-1
    if (j1.eq.1) j2=1
    j3 = j1 + 21
    am(j1)=store(ihamr,j3,1)/32.17
    sta(j1) = store(ihamr,j3,2)*convs
    do 103 j4=j2,j1
    tcra=am(j1)/sta(j4)
    if (tcra.lt.dtcra) dtcra=tcra
103  continue
    dtcra=sqrt(dtcra)
105  tdel=store(ihamr,10,1)
c    store(ihamr,11,1) is vin in in3
    vcham=store(ihamr,11,2)/1728.0
    depib=store(ihamr,12,1)/12.
    aram=store(ihamr,12,2)/144.0
    sjjj = store(ihamr,10,2)
    do 106 i=17,21
    pbar(i-16)=store(ihamr,i,1)
106  pbar(i-11)=store(ihamr,i,2)
    pbar(11)=store(ihamr,13,1)
    pbar(12)=store(ihamr,13,2)
    pbar(13)=store(ihamr,14,1)
    pbar(14)=store(ihamr,14,2)
    pbar(15)=store(ihamr,15,1)
    pbar(19)=store(ihamr,19,1)
c    pbar(20) gtr 0 identifies vacuum chamber hammer
    pbar(20)=store(ihamr,21,2)
    strm=store(ihamr,16,1)/12.

```

```

pbar(10)=strm
strmax = 0.0
c  pcom is pressure for corresponding fuel setting
c  pbar(11) is the maximum down pressure on a double acting a/s
c  hammer identifier for such hammers
expp=store(ihamr,16,2)
if (expp.lt.0.1) expp=1.3
if (pbar(1).lt.10.0e-8) pbar(1) = aram
efficy=store(ihamr,15,2)
if (efficy.lt.0.01) efficy=0.95
rmwt=0.0
do 220 i=1,m
220 rmwt=rmwt+hm(i)
pbar(16)=efficy
pbar(17)=rmwt*32.17
pbar(18)=store(ihamr,19,2)
return
end
subroutine airstm
common vinh,dinh,aoh,voh,doh,anh,vnh,dnh,sth,hm,am,sta,ma,esoil,
1 aop,anp,dop,dnp,vop,vnp,stp,pm,cdp,res,reso,sj,sok,qs,su,
2 splice,depib,aram,ecus,eass,ept,eant,vfall,vcham,xpt,sjjj,
3 emp,rho,dth,ntp,n,m,jrat,iult,rult,soilm,dsacp,
4 ftpo,iw,pcom,tdel,iout,out,expp,tem1,temo,inp,stroko,strokn,vo,po
5 ,dfin,jdout,jpmax,ex,icol,fintmx, temax,result,ismith,iter
6 ,ityph,ipreig,vfallm,strmax,dcyl,pbar,area,jex
dimension vinh(10),dinh(10),aoh(10),voh(10),doh(10),anh(10),
1 vnh(10),dnh(10),sth(10),hm(10),vnp(99),am(10),sta(10),voa(10),
2 doa(10),aoa(10),vna(10),dna(10),ana(10),delaa(10),
3 aop(99),anp(99),dop(99),dnp(99),vop(99),stp(99),pm(99),cdp(99),
4 res(99),reso(99),sj(99),sok(99),qs(99),su(99),splice(99),
4dsacp(3),out(3200),inp(13),ex(600),result(100),jex(600)
6 ,aoah(99),voah(99),doah(99),area(99),delah(99)
7 ,voaa(10),aoaa(10),doaa(10),pbar(20),hmi(10),pmi(99),soi(99)
8 ,ami(10)
klim = 200
jout = 0
nl=n+1
nml = n-1
ml = m+1
mar = ma
if (mar.lt.1) mar = 1
iai=0
dt2=ntp*.5
dt6=ntp*ntp/6.0
dh2=dth*.5
dh6=dth*dth/6.0
dsi1=1./dsacp(1)
dsi2=1./dsacp(2)
dsi3=1./dsacp(3)
oojrt=1./float(jrat)
dtpm=ntp*1000.
tt=0.0

```

```

ml = ma
cdh = pm(99)
if (ml.gt.ml) ml = ml
tlim = 0.0
iiout = iabs(iout)
twoloc=2.0*xpt/sqrt(emp/rho)
t2lc = twoloc*1000.
if (iiout.gt.19) iiout = iiout - 20
if (iiout.gt. 9) iiout = iiout - 10
jdout = 3.0*twoloc/dtp
jdout = jdout / 100
tjd = float(jdout)*dtp*1000.
c   tjd is time increment of output in milliseconds
if (tjd.gt.0.5) jdout = (0.0005/dtp) + 0.5
if (jdout.lt.1) jdout=1
jmax = 6
c   6 colums of pile output if icol =0  13 otherwise if not gtr n
if (icol.gt.0) jmax=13
if (jmax.gt.n) jmax=n
kk=0
red = 0.9
if (emp.lt.3000000.) red = 0.0
scj = 0.0
wram = 0.0
g = 32.170
do 902 i=1,m
902  wram = wram + hm(i)*g
    wass=0.0
    do 905 i=1,mar
905  wass=wass+am(i)*g
    faoa=wass
    gas = g*1.5
    if (wass.gt.0.0) gas = (1.-sjjj)*g*(1.+wram/wass)
    gr = g
    if (ma.lt.1) faoa = 0.0
    fa = 0.0
    fao = 0.0
    ftpo=wass+hm(ml)*g
    ftit = ftpo
    ftpoa =ftpo
    fanvoa = 0.0
    vnplit = vop(n)
    vnhit = vinh(m)
    tbef=0.002
    vfallb=vfall-g*tbef
    vinh(ml) = 0.0
    do 903 i=1,mar
    ami(i)=1./am(i)
    delaa(i) =0.0
    voaa(i) =0.0
903  aoaa(i) =0.0
    dnpmax = 0.0
    mml=m-1

```

```

maml = ma-1
enthru = 0.0
emax = 0.0
nt = 5.0*twoloc/dtp
jck = 1.5*twoloc/dtp+0.003/dtp
if (float(nt)*dtp.lt.0.050) nt = 0.050/dtp
if (temax.gt.1.0) nt=temax/dtpm
stpt1 = sth(m1)*stp(1)/(sth(m1)+stp(1))
stpt2 = (sth(m1)/ecus)*(stp(1)/ept)/((sth(m1)/ecus)+stp(1)/ept)
stal = sth(m)
star = stal/eanv
stas1 = sta(mar)
stasr = stas1/eass
wp=0.0
do 51 i=1,nml
pmi(i)=1./pm(i)
soi(i)=sok(i)/esoil
51 wp=wp+pm(i)*g
soi(n)=sok(n)/esoil
soi(nl)=sok(nl)/esoil
pmi(n)=1./(pm(n)+soilm)
do 98 i=n,nl
dop(i)=0.0
vop(i)=0.0
aop(i)=g
98 reso(i)=0.0
do 52 i=2,n
il=nl-i
dop(il)=dop(il+1)+((ftpo+wp)/stp(il+1))*float(i)/float(n)
vop(il)=0.0
aop(il)=g
reso(il)=0.0
52 wp=wp-pm(il)*g
dinh(m1)=dop(1)+sqrt(ftpo*dsacp(2)/stpt1)
mal=ma+1
if (ma.lt.1) go to 1906
doaa(ma)=dinh(m1)+sqrt(wass*dsacp(3)/sta(ma))
w=wass-am(ma)*g
if (ma.lt.2) go to 1906
do 906 i=2,maml
il=mal-i
doaa(il)=doaa(il+1)+w/sta(il)
906 w=w-am(il)*g
1906 continue
do 901 i=1,m
vinh(i)=vfallb
901 dinh(i)=-0.5*(vfallb+vfall)*tbef+dinh(m1)
do 100 i=1,m1
voah(i)=vinh(i)
doah(i)=dinh(i)
delah(i) = 0.0
aoah(i)=g
hmi(i)=1./hm(i)

```

```

100  continue
    epsv = 0.1
    epsf = 1.0
    limout=klim*16
    do 450 i=1,limout
450  out(i)=0.0
        do 460 j=1,6
        do 460 i=1,n
            ij=n*(j-1)+i
            ex(ij) = 0.0
            jex(ij) = 0
460  continue
        fintmx=0.0
        if (iout.lt.0) write (6,653)
        vlpmax = 0.0

c
c
c  start main loop
c
    do 700 j=1,nt
        if (j.gt.jck) jck=-1
        tt=tt+dtpm
        it=0
        if (vlpmax.lt.vnp(1)) vlpmax = vnp(1)
        if (red.lt.0.01) go to 103
        if (vnp(1).gt.0.9*vlpmax.or.vlpmax.lt.5.0) go to 103
        stpt1 = stpt2 * red
        red = 0.0
103  continue
800  continue
        fao = faoa
        ftpo = ftpoa
        fanvo = fanvoa
        do 99 i=1,m1
            voa(i) = voaa(i)
            aoa(i) = aoaa(i)
            doa(i) = doaa(i)
            voh(i)=voah(i)
            aoh(i)=aoah(i)
99   doh(i)=doah(i)
c
c  hammer analysis
c
        do 600 jh=1,jrat
            ith=0
c  prediction
            do 585 i=1,m1
                if (it.eq.0) go to 909
                anh(i) = aoh(i)+float(jh)*delah(i)
                ana(i) = aoa(i)+float(jh)*delaa(i)
                go to 585
909  continue
        anh(i) = aoh(i)

```



```

ana(i) = aoa(i)
585 continue
do 586 i=1,mar
586 call integr(dth,aoa,ana,voa,doa,vna,dna,i,dh2,dh6)
do 101 i=1,m1
101 call integr(dth,aoh,anh,voh,doh,vnh,dnh,i,dh2,dh6)
500 continue
i=1
if (vlpmax.gt.1.5) g=g*(-sjjj)
if (m.eq.1) go to 904
anh(1)=g+(dnh(2)-dnh(1))*sth(1)*hmi(1)
call integr(dth,aoh,anh,voh,doh,vnh,dnh,i,dh2,dh6)
if (m.eq.2) go to 904
do 102 i=2,mml
i1=i+1
i1=i-1
anh(i)=g+((dnh(i1)-dnh(i))*sth(i)+(dnh(i1)-dnh(i))*sth(i1))
1 *hmi(i)
call integr(dth,aoh,anh,voh,doh,vnh,dnh,i,dh2,dh6)
102 continue
904 continue
c last ram element
fanv = 0.0
danv = 0.0
ddis = dnh(m1)-dnh(m)
if (ddis.gt.0.0) go to 811
dvel = vnh(m1) - vnh(m)
danv = cdh*dvel
c
ddiso = doh(m1)-doh(m)
ddd = ddis - ddiso
c anvil spring force always tension= +
call stiff(ddis,dvel,stal,star,fanvo,fanv,ds11,ddd)
if (fanv.gt.0.0) fanv = 0.0
811 continue
fintac = fanv
if (tlim.lt.0.0001.and.fintac.lt.10.0) tlim = tt+t2lc*1.15
ftr = 0.0
if (m.gt.1) ftr = (dnh(mml)-dnh(m))*sth(mml)
anh(m)=g+(ftr+fintac+danv)*hmi(m)
g = gr
call integr(dth,aoh,anh,voh,doh,vnh,dnh,m,dh2,dh6)
fintac = -fintac
c now check total assembly. there are at least two assembly elements
if (ma.lt.1) go to 911
if (vlpmax.gt.1.5) g = gas
i = 1
ana(1)=g+(dna(2)-dna(1))*sta(1)*ami(1)
call integr(dth,aoa,ana,voa,doa,vna,dna,i,dh2,dh6)
if (ma.lt.3) go to 912
do 907 i=2,maml
i1 = i+1
i1 = i-1

```

```

      ana(i)=g+((dna(i1)-dna(i))*sta(i)+(dna(i1)-dna(i))*sta(i1))
1    *ami(i)
      call integr(dth,aoa,ana,voa,doa,vna,dna,i,dh2,dh6)
907  continue
912  continue
      fa = 0.0
c    last assembly element
      ddis = dnh(m1)-dna(ma)
      if (ddis.gt.0.0) go to 908
      dvel=vnh(m1)-vna(ma)
c
      ddiso = doh(m1)-doa(ma)
      ddd = ddis-ddiso
      fao = -fao
      call stiff(ddis,dvel,stas1,stasr,fao,fa,dsi3,ddd)
      if (fa.gt.0.0) fa = 0.0
      fao = -fao
908  continue
      if(ma.eq.1)ana(1)=g+fa*ami(1)
      if (ma.gt.1)
1    ana(ma)=g+((dna(ma1)-dna(ma))*sta(ma1)+fa)*ami(ma)
      call integr(dth,aoa,ana,voa,doa,vna,dna,ma,dh2,dh6)
      fa = -fa
      g = gr
911  continue
c    now analyze cap
      dtl=float(jh)*dth
      i=1
      if(it.gt.0.and.jrat.eq.1)go to 575
      call integr(dth,aop,aop,vop,dop,vnp,dnp,1,dh2,dh6)
      if(jrat.eq.1)go to 575
      dtl=float(jh)*dth
      dnpl=dop(1)+(dtl/dtp)*(dnp(1)-dop(1))
      vnpl=vop(1)+(dtl/dtp)*(vnp(1)-vop(1))
      if(it.gt.0)go to 576
      dl2=dtl*.5
      dl6=dtl*dtl/6.
      call integr(dtl,aop,aop,vop,dop,vnp,dnp,1,dl2,dl6)
575  continue
      dnpl=dnp(1)
      vnpl=vnp(1)
576  continue
      ftp = 0.0
      ddis = dnpl-dnh(m1)
      if (ddis.gt.0.0) go to 813
      dvel = vnpl-vnh(m1)
c
      ddiso = dop(1)-doh(m1)
      ddd = ddis-ddiso
      ftpo = -ftpo
      call stiff(ddis,dvel,stpt1,stpt2,ftpo,ftp,dsi2,ddd)
      if (ftp.gt.-splice(1)) ftp = -splice(1)
      ftpo = -ftpo

```

```

813  continue
    anh(ml)=g+(fintac+fa+ftp-danv+dvel*cdp(1))*hmi(ml)
    call integr(dth,aoh,anh,voh,doh,vnh,dnh,m1,dh2,dh6)
    ith = ith +1
    ftp = -ftp
c    check to see if hammer has converged
    if (jrat.eq.1) go to 650
    if (ith.gt.iter) go to 650
    err = abs(vnhit-vnh(m))
    vnhit = vnh(m)
    if (err.gt.epsv) go to 500
650  continue
    do 501 i=1,m1
    delaa(i)=(ana(i)-aoa(i))*oojrt
    delah(i)=(anh(i)-aoh(i))*oojrt
    aoa(i) = ana(i)
    voa(i) = vna(i)
    doa(i) = dna(i)
    aoh(i)=anh(i)
    voh(i)=vnh(i)
501  doh(i)=dnh(i)
    vcap = vnh(m1)
    fanvo = fanv
    ftpo = ftp
    fao = fa
600  continue
c
c    pile analysis
c
    call pilean (ipt,aop,vop,dop,anp,vnp,dnp,stp,pm,splice,sj,cdp,res,
1  reso,su,sok,vnpit,vcap,soilm,scj,dtp,n,it,ismith,epsv,iai,esoil,
2  ftp,jrt,iter,soi,dt2,dt6,pmi,rsum,dsum)
c    check to see if overall cycle converged
    it = it + 1
    if (it.gt.iter) go to 850
    errv = abs(vnpit - vnp(n))
    if (jrat.eq.1) vnpit = vnp(n)
    if (jrat.eq.1.and.errv.gt.epsv) go to 800
    err = abs(ftit - ftp)
    ftit = ftp
    if (err.gt.epsf) go to 800
850  continue
c
c    go to 800 means repeat
c    go to 850 means sufficiently many iterations were performed
    vnpmax = 0.0
    do 401 i=1,n1
    reso(i)=res(i)
    aop(i)=anp(i)
    vop(i)=vnp(i)
    if (abs(vop(i)).gt.vnpmax) vnpmax = abs(vop(i))
401  dop(i)=dnp(i)
    if (dnp(n).gt.dnpmax) dnpmax = dnp(n)

```

```

fanvoa = fanvo
faoa = fao
ftpoa = ftpo
c set old equal new
do 402 i=1,m1
  aoaa(i) = ana(i)
  voaa(i) = vna(i)
  doaa(i) = dna(i)
  aoah(i)=anh(i)
  voah(i)=vnh(i)
402 doah(i)=dnh(i)
  dopinc = dop(1)*12.
  dra = (dnh(m1)-dnh(m))*12.0
  sea = (dnh(m1)-dna(mar))*12.0
  dbo = dnp(n)
  den = vop(1)*dtp*ftp
  enthru = enthru + den
  if (enthru.gt.emax) emax = enthru
  jout = jout + 1
  if (jout.ne.jdout) go to 699
  jout = 0
  ftpo=ftp
c output for neg. option only
no2 = n/2
c
652 format (1h ,i4,i2,3i1,3f6.1,f6.2,2f7.0,2f6.1,2(f6.1,f8.1),3f8.1,
1 2f6.1,f7.2)
  do 686 i=no2,nml
    if(i.ne.no2.and.i.ne.nml)go to 686
    il = i+1
    com = dnp(i)-dnp(il)
    f = com*stp(il)
    if (f.lt.splice(il)) f = splice(il)
    if (f.gt.0.0.or.splice(il).lt.-.5) go to 685
    f = 0.0
    if (com.lt.splice(il)) f = (com-splice(il))*stp(il)
685 fopn = f
    if (i.eq.no2) fop5 = f
686 continue
653 format
1 (132h jp fiph time vru van sep fint ftp
a en v5 f5 vn fn fa rs da d
bdb sea )
  if (iout.lt.0) write (6,652)
a j,ifirst,it,ith,ipt,tt,vnh(m),vnh(m1),dra,fintac,ftp,vo
1 ),enthru,vop(no2),fop5,vop(n),fopn,fa,rsum,dsum,dopinc,dbo,sea
c
c store output
  if (iout.lt.6) call fill(kk,klim,out,jmax,iout,dnh,sth,finta,
1 ftp,inp,dnp,stp,splice,vnh,vnp,anh,anp,jpmax,maxjp,ex,area,j,m,
2 jex,jck)
c
c fills in the output array out(klim*15)

```

```

c
    if (iout.eq.6) jl = j
    if (iout.eq.6) call fill2 (n,m,out,sea,dnh,ftp,vnp,
1 dnp,fop5,sj,rsum,dsum,res,fopn,kk,icol,tt,ismith,jl,maxjp,jpmax,
2 stp,area,splice,ex,jex,j,jck)
    if (iout.ge.0) go to 463
    if (fintac.gt.fintmx) fintmx=fintac
463 continue
    if (temax.gt.1.0) go to 699
    if (tt.lt.tlim) go to 699
    if (tt.lt.0.5*t21c) go to 700
    if (dnp(n).lt.dnpmax*0.8) go to 701
699 ifirst = 0
c
c     end of main loop
c
700 continue
701 continue
    dfin = (dnpmax-qs(n))*12.
    efin = enthru
    if (dfin.gt.0.0) iblct =12./dfin
654 format (1h0,10h  fin set= ,f8.4,2lh inches    fin enthru= ,f6.1,
1 19h k-ft    max enthru= ,f6.1,19h k-ft    blow count= ,i5 )
    if (iout.lt.0) write (6,654) dfin,efin,emax,iblct
    anp(99) = emax
    return
    end
    subroutine output (out,res,jnp,iout,dt,jpmax,jdout,iult,n,jdlim,
1 alph,iseg,jselec,ymax,coli,title,stpl,pml)
    common/dot/ibufo(2000),ibufo
    dimension jselec(13),big(5),ply(5),prj(5),punit(5)
2 ,out(3000),jnp(13),res(100),iseg(15),alph(99),
1 jdum(15),pys(5),iunit(5),salph(99),title(10)
    ibufo = ibufo+1
    if (ibufo.eq.0) call plots(ibuf,2000,4)
c
    i9=1
112 jpm=jdlim
c
c
    ymaxr=ymax
    if (iabs(iout).lt.10) return
    dtp=dt*float(jdout)*1000.0
    dtpm=dtp*float(jdlim)
    irult = res(1)
    iu = iabs(iult)
    if (iult .lt. 0) iult = 0
    if (iu .gt. 1) irult = res(iu)
    ymin=0.
    pleny=4.
    plenx=6.
    ymax=ymax*.999
    ymaxc=0.

```



```

        if (iabs(iout).eq.16.or.iabs(iout).eq.26) go to 171
        do 120 j=1,13
        if (jnp(j).eq.0) go to 121
120      jdum(j)=j+2
121      jjnp=13
        if (j.ne.13) jjnp=j-1
        if (j.eq.13.and.jnp(13).eq.0) jjnp=12
        jjnp2=jjnp+2
        if (iabs(iout).eq.10.or.iabs(iout).eq.20) go to 300
        do 100 isegno = 3,jjnp2
        do 100 j=1,jpm
        i = (isegno-1)*200+j
        if (out(i).lt.ymin) ymin = out(i)
100      if (out(i).gt.ymaxc) ymaxc = out(i)
        go to 170
171      il=1
        eaoc=1.
        if (iseg(i9).eq.0) kk=1
        if (iseg(i9).eq.1.or.iseg(i9).eq.3) kk=4
        if (iseg(i9).eq.2) kk=5
173      do 172 i=1,jpm
        ij=kk*100+i
        if (kk.eq.5.and.il.eq.2) eaoc=sqrt(stpl*pml)
        if (out(ij)*eaoc.lt.ymin) ymin=out(ij)*eaoc
172      if (out(ij)*eaoc.gt.ymaxc) ymaxc=out(ij)*eaoc
        il=il+1
        kk=kk+9
        if (iseg(i9).eq.3) kk=5
        if (kk.eq.10) go to 170
        if (il.lt.3) go to 173
170      if (ymaxc.gt.10.*ymax) ymax=ymaxc
        del=(ymax-ymaxr)/ymax
105      ym = ymax
        do 101 j=1,15
        if (ym.lt.1.) go to 102
101      ym = ym/10.
102      rj = 10.**(j-2)*2.
        i = ymax/rj+1.0
c
        ymax = rj*float(i)
        if (abs(del).lt.0.01) ymax=ymaxr
        ymaxr=ymax
        ymi = -ymax
        if (-ymin.lt.ymax) ymi=-ymax/3.0
        if (-ymin.lt.ymax/6.0) ymi = 0.0
        ymin = ymi
        ys = (ymax-ymin) / pleny
        xs = dtpm/plenx
        k = dtpm/12.
        xunit = k+1
        call fscale(-10.0*xs,ymin,xs,ys,ymax,lh ,0.,0.)
        call fmove(0.0,0.0)
        if (iabs(iout).eq.16.or.iabs(iout).eq.26) go to 145

```



```

do 103 i=1,15
  if (iseg(i).eq.0.and.i.gt.1) go to 103
  isegno = iseg(i)
  if (isegno.eq.0) isegno = 3
  do 106 j = 1,jpm
    ii = (isegno-1)*200 + j
106  call fplot(float(j)*dtp,out(ii))
    call fmove(0.,0.)
103  continue
145  if (iabs(iout).ne.16.and.iabs(iout).ne.26) go to 150
    if (iseg(19).eq.0) kk=1
    if (iseg(19).eq.1.or.iseg(19).eq.3) kk=4
    if (iseg(19).eq.2) kk=5
    do 140 j=1,jpm
      ii=kk*100+j
140  call fplot (float(j)*dtp,out(ii))
      if (kk.eq.1) go to 150
      kk=kk+9
      if (iseg(19).eq.3) kk=5
      eaoc=1.
      if (kk.eq.5) eaoc=sqrt(stp1*pml)
      call fmove (0.,0.)
      do 141 j=1,jpm
        ii=kk*100+j
141  call fplot (float(j)*dtp,out(ii)*eaoc)
        call fmove (4.*xs,3.5*ys)
        if (kk.eq.13.or.kk.eq.5) call fchar (0.,.12,9,9htop force)
        if (kk.eq.14) call fchar (0.,.12,8,8htop vel.)
        call fmove (5.*xs,3.56*ys)
        call fplot (5.4*xs,3.56*ys)
        call fmove (4.*xs,3.65*ys)
        if (kk.eq.13) call fchar (0.,.12,9,9htoe force)
        if (kk.eq.14) call fchar (0.,.12,8,8htoe vel.)
        if (kk.eq.5) call fchar (0.,.12,9,9hvel.*ea/c)
        call fmove (5.*xs,3.71*ys)
        call fplot (5.4*xs,3.71*ys)
150  call fmove (dtpm+.3*xs,-.12*ys)
        call fchar(0.,.12,4,4hmsec)
        call fmove (dtpm+.3*xs,.05*ys)
        call fchar (0.,.12,3,3hl/c)
        call fmove (dtpm,0.)
        l=dtpm/xunit
        do 130 i=1,1
          xuse=xunit*(1-i+1)
          call fplot (xuse,0.)
          call fplot (xuse,-.05*ys)
          call fmove (xuse-.25*xs,-.2*ys)
c
          call fnumb (0.,.12,0,xuse)
          call fmove (xuse,0.)
130  continue
        call fplot (0.,0.)
        call fmove (2.8*xs,-.4*ys)

```

```

call fchar (0.,.12,4,4htime)
call fmove (dtpm,0.)
l=dtpm/coli
do 146 i=1,1
li=1-i+1
xuse=coli*li
call fplot (xuse,0.)
call fplot (xuse,0.05*ys)
call fmove (xuse-.35*xs,.1*ys)
call fnumb (0.,.12,-1,float(li))
call fmove (xuse,0.)
146 continue
call fmove (0.,ymin)
call fplot (-.05*xs,ymin)
call fmove (-.9*xs,ymin-.06*ys)
call fnumb (0.,.12,2,ymin)
call fmove (0.0,ymin)
m=pleny+.001
do 131 i=1,m
yuse=ymin+i*ys
call fplot (0.,yuse)
call fplot (-.05*xs,yuse)
call fmove (-.9*xs,yuse-.06*ys)
call fnumb (0.,.12,2,yuse)
call fmove (0.,yuse)
131 continue
call fmove (-xs,ymax-3.0*ys)
iouts=iout
if (iabs(iout).gt.20) iout=iabs(iout)-10
if (iseg(i9).eq.1.and.iabs(iout).eq.16) iout=11
if (iseg(i9).eq.3.and.iabs(iout).eq.16) iout=11
if (iseg(i9).eq.2.and.iabs(iout).eq.16) iout=12
if (iabs(iout).eq.11) call fchar (90.,.12,13,13hforce in kips)
if (iabs(iout).eq.12) call fchar (90.,.12,18,18hvelocity in ft/s)
if (iabs(iout).eq.13) call fchar (90.,.12,13,13hstress in ksi)
if (iabs(iout).eq.14) call fchar (90.,.12,17,17hacceleration in g)
if (iabs(iout).eq.15) call fchar (90.,.12,15,15hdispl in inches)
if (iabs(iout).eq.16) call fchar (90.,.12,15,15hpressure in psi)
iout=iouts
call fmove (.75*xs,4.1*ys)
call where (xc,yc,f)
160 call symbol (xc,yc,.15,title,0.,40)
call fmove (-3.0*xs,ymax+5.0*ys)
if (iabs(iout).eq.16.or.iabs(iout).eq.26) ymax=0.0
i9=i9+1
if (iabs(iout).eq.16.and.iseg(i9).ne.0) go to 112
if (iabs(iout).eq.26.and.iseg(i9).ne.0) go to 112
200 if (iabs(iout).lt.20) go to 300
if (iabs(iout).eq.26) go to 300
plenx=4.
pleny=1.5
plenmx=1.
plenmy=2.5

```

```

ymax=0.
do 201 j=1,jpm
do 201 i=1,jjnp
k=jselec(i)
if (jselec(1).eq.0) k=i+2
if (k.le.2.or.k.gt.15) go to 201
k=(k-1)*200+j
if (out(k).gt.ymax) ymax=out(k)
201 continue
ym=ymax
do 202 j=1,15
if (ymax.lt.1.) go to 203
202 ymax=ymax/10.
203 rj=10.**((j-2)*2.0
i=ym/rj+1.
ymax=rj*float(i)
xs=dtpm/plenx
ys=ymax/pleny
call fscale(-9.0*xs,-2.5*ys,xs,ys,ymax,lh ,0.,0.)
nn=n
if (nn.gt.jnp(jjnp)) nn=jnp(jjnp)
salph(1)=0.
do 206 i=2,jjnp
jend=jnp(i)
jbeg=jnp(i-1)+1
salph(i)=salph(i-1)
do 206 j=jbeg,jend
206 salph(i)=salph(i)+alph(j)
do 204 j=1,jjnp
i=jselec(j)
if (jselec(1).eq.0) i=j+2
if (i.le.2.or.i.gt.15) go to 204
pmy=plenmy*salph(j)/salph(nn)
pmx=pmy*plenmx/plenmy
pmy=-pmy*ys
pmx=-pmx*xs
call fmove (pmx,pmy)
do 205 k=1,jpm
kk=(i-1)*200+k
205 call fplot (pmx+float(k)*dtp,pmy+out(kk))
call fplot (pmx+float(k)*dtp,pmy)
204 continue
call fplot (pmx,pmy)
call fmove (pmx+float(k)*dtp,pmy)
call fplot (dtpm,0.)
call fmove (dtpm+xs*.3,0.)
call fchar (0.,.12,12,12h time in msec)
call fmove (dtpm,0.)
call fplot (dtpm,-.05*ys)
call fmove (dtpm-.12*xs,-.2*ys)
call fnumb (0.,.12,2,dtpm)
call fmove (dtpm,0.)
call fplot (0.,0.)

```

c

```

call fplot (pmx,pmy)
call fmove (.55*pmx-.3*xs,.55*pmy+.1*ys)
call fchar (67.5,.12,6,6hlength)
call fmove (-.3*xs,.25*ymax)
if (iabs(iout).eq.21) call fchar (90.,.12,5,5hforce)
if (iabs(iout).eq.22) call fchar (90.,.12,4,4hvel.)
if (iabs(iout).eq.23) call fchar (90.,.12,6,6hstress)
if (iabs(iout).eq.24) call fchar (90.,.12,4,4hacc.)
if (iabs(iout).eq.25) call fchar (90.,.12,6,6hdispl.)
call fmove (-.9*xs,ymax-.05*ys)
call fnumb (0.,.12,2,ymax)
call fmove (0.,ymax)
call fplot (-.05*xs,ymax)
call fmove (0.,ymax)
call fplot (0.,0.)
call fmove (.75*xs,1.6*ys)
call where (xc,yc,f)
210 call symbol (xc,yc,.15,title,0.,40)
call fmove (-3.0*xs,ymax+5.0*ys)
300 if (iult.eq.0) return
do 301 j=1,5
big(j)=0.
do 302 i=1,iult
k=(j-1)*10+i
302 if (abs(res(k)).gt.big(j)) big(j)=abs(res(k))
big(j)=big(j)*1.1
ym=big(j)
do 303 i=1,15
if (ym.lt.1.) go to 304
303 ym=ym/10.
304 prj(j)=10.**(i-2)
i=big(j)/prj(j)+1.
301 big(j)=prj(j)*float(i)
p1ly(1)=4.
p1ly(2)=6.
p1ly(3)=5.
p1ly(4)=3.0
p1ly(5)=3.
if (big(5).gt.big(4)) prj(4)=prj(5)
if (big(4).gt.big(5)) prj(5)=prj(4)
if (big(5).gt.big(4)) big(4)=big(5)
if (big(4).gt.big(5)) big(5)=big(4)
do 305 j=1,5
pys(j)=big(j)/p1ly(j)
punit(j)=prj(j)/5.
305 iunit(j)=big(j)/prj(j)+1.001
big(1)=big(1)*1.25
call fscale ( 9.0*pys(2),-9.0*pys(3),pys(2),pys(3),big(3),1h ,0.
>0.)
i3=big(2)/punit(2)
i2w=1
do 309 i=1,i3

```

```

xuse=punit(2)*i
call fplot (xuse,0.)
call fplot (xuse,-.05*pys(3))
if (i.ne.i2w*5) go to 309
i2w=i2w+1
call fplot (xuse,.05*pys(3))
call fmove (xuse-.25*pys(2),-.2*pys(3))
call fnumb (0.,.12,0,xuse)
309 call fmove (xuse,0.)
call fplot (big(2),0.)
if (iunit(3).gt.big(3)) iunit(3)=iunit(3)/10.+1
i2=big(3)/iunit(3)
do 308 i=1,i2
yuse=iunit(3)*i
call fplot (big(2),yuse)
call fplot (big(2)+.05*pys(2),yuse)
call fmove (big(2)+.1*pys(2),yuse-.06*pys(3))
call fnumb (0.,.12,0,yuse)
308 call fmove (big(2),yuse)
call fplot (big(2),big(3))
call fmove (big(2)+.3*pys(2),yuse-.25*pys(3))
call fchar (0.,.12,3,3hft.)
ndo = iult
do 310 j=1,iult
call fmove (res(ndo+10)-.06*pys(2),res(ndo+20)-.06*pys(3))
call fchar(0.0,.12,1,lhs)
310 ndo = ndo-1
call fmove (0.0,0.0)
call fscale (-0.0,0.0*pys(1),pys(2),pys(1),big(1),lh ,0.,0.)
il=big(1)/punit(1)
ilw=1
do 306 i=1,il
yuse=punit(1)*i
call fplot (0.,yuse)
call fplot (-.05*pys(2),yuse)
if (i.ne.ilw*5) go to 306
ilw=ilw+1
call fplot (.05*pys(2),yuse)
call fmove (-.9*pys(2),yuse-.05*pys(1))
call fnumb (0.,.12,2,yuse)
306 call fmove (0.,yuse)
call fplot (0.,big(1))
call fmove (-.5*pys(2),(ilw-1)*prj(1)-.2*pys(1))
call fchar (0.,.12,4,4htons)
call fmove(1.3*pys(2),4.6*pys(1))
call fchar (0.,.12,12,12hresistance=x)
call fmove (3.4*pys(2),4.6*pys(1))
call fchar (0.,.12,8,8hstroke=s)
c dummy out original source statement
ndo = iult
do 307 j=1,iult
call fmove (res(ndo+10)-.06*pys(2),res(ndo)-.06*pys(1))
call fchar (0.,.12,1,lhx)

```



```

307  ndo = ndo-1
      call fmove (2.3*pys(2),-.4*pys(1))
      call fchar (0.,.12,14,14hblows per foot)
      call fmove (0.0,0.0)
      call fscale (-0.0,-5.0*pys(4),pys(2),pys(4),big(4),lh ,0.,0.)
      i4=big(4)/punit(4)
      do 311 i=1,i4
        yuse=punit(4)*i
        call fplot (0.,yuse)
        call fplot (-.05*pys(2),yuse)
        do 316 j=1,10
          xuse=j*prj(4)
          if (abs(xuse-yuse).lt.0.001) go to 317
316  continue
        go to 311
317  call fplot (.05*pys(2),yuse)
      call fmove (-.9*pys(2),yuse-.05*pys(4))
      call fnumb (0.,.12,2,yuse)
      fy=yuse
311  call fmove (0.,yuse)
      call fplot (0.,big(4))
      call fmove (-.4*pys(2),fy-.2*pys(4))
      call fchar (0.,.12,3,3hksi)
      call fmove (1.2*pys(2),2.8*pys(4))
      call fmove (.9*pys(2),2.8*pys(4))
      call fchar(0.,.12,20,20hcompressive stress=c)
c    dummy out original source statement
      call fmove (3.4*pys(2),2.8*pys(4))
      call fchar (0.,.12,16,16htensile stress=*)
      ndo = iult
      do 313 j=1,iult
        call fmove (res(ndo+10)-.06*pys(2),abs(res(ndo+30))-.06*pys(4))
        call fchar (0.0,0.12,1,1h*)
313  ndo = ndo-1
      call fmove (0.,0.)
      i2w=1
      do 314 i=1,i3
        xuse=punit(2)*i
        call fplot (xuse,0.)
        call fplot (xuse,-.05*pys(4))
        if (i.ne.i2w*5) go to 314
        i2w=i2w+1
        call fplot (xuse,.05*pys(4))
        call fmove (xuse-.25*pys(2),-.2*pys(4))
        call fnumb (0.,.12,0,xuse)
314  call fmove (xuse,0.)
        call fplot (big(2),0.)
        call fplot (big(2),big(4))
        call fplot (0.,big(4))
        do 315 j=1,iult
          call fmove (res(j+10)-.06*pys(2),res(j+40)-.06*pys(4))
315  call fchar (0.0,.12,1,1hc)
        call fmove (.75*pys(2),3.2*pys(4))

```



```

    call where (xc,yc,f)
320 call symbol (xc,yc,.15,title,0.,40)
    call plot (4.5,0.0,999)
    return
end
subroutine fnumb (ang,size,ndec,var)
    call where(xc,yc,f)
    call number (xc,yc,size,var,ang,ndec)
    return
end
subroutine fplot (x,y)
    common/chiplt/xino,yino,xsc,ysc,ymaxo
    xn=x/xsc
    yn=y/ysc
    call plot (xn,yn,2)
    return
end
subroutine fmove (x,y)
    common/chiplt/xino,yino,xsc,ysc,ymaxo
    xn=x/xsc
    yn=y/ysc
    call plot (xn,yn,3)
    return
end
subroutine fchar (ang,size,length,istr)
    dimension istr(10)
    call where(xc,yc,f)
    call symbol (xc,yc,size,istr,ang,length)
    return
end
subroutine fschar (ang,size,no)
    dimension ia(1)
    if (no.lt.10) ia(1)=0
    if (no.lt.50.and.no.gt.9) ia(1)=1
    if (no.lt.71.and.no.gt.49) ia(1)=2
    if (no.gt.70) ia(1)=5
    call symbol (xc,yc,size,ia,ang,-1)
    return
end
subroutine fscale (xin,yin,xs,ys,ymax,iaxis,xun,yun)
    common/chiplt/xino,yino,xsc,ysc,ymaxo
    xsc=xs
    ysc=ys
    xino=xin/xsc
    yino=yin/ysc
    call plot (-xino,-yino,-3)
    ymaxo=ymax/ysc
    return
end
subroutine sumout(result,ityph,iult,iw,iex)
    dimension result(100),iex(40)
11 format(50h      r ult  bl ct stroke (ft)   minstr i,j   maxstr,
1      20h  i,j   energy bl rt)

```

```

12  format(50h      tons      bpf  down  up      ksi      ksi,
1    20h          ft-kip  bpm)
21  format(50h      r ult  bl ct stroke  bcp  minstr  i,j  maxstr,
1    20h      i,j  energy bl rt)
22  format(50h      tons      bpf  ft  psi      ksi      ksi,
1    20h          ft-kip  bpm)
31  format(50h      r ult  bl ct stroke(eq.)  minstr  i,j  maxstr,
1    14h      i,j  energy)
32  format(50h      tons      bpf          ft      ksi      ksi,
1    14h          ft-kip)
13  format(1h ,f7.1,i7,2f6.1,f8.2,1h(,i2,1h,,i3,1h),f7.2,1h(,i2,1h,,
1    i3,1h),2f6.1)
33  format(1h,f7.1,i7,f12.2,f8.2,1h(,i2,1h,,i3,1h),f7.2,1h(,i2,1h,,
1    i3,1h),2f6.1)
c    subheadings; main heading printed in main program
    if(ityph.gt.1)go to 101
    write (6,11)
    write (6,12)
    go to 200
101  if(ityph.gt.2)go to 102
    write (6,21)
    write (6,22)
    go to 200
102  write (6,31)
    write (6,32)
200  continue
c    result contains (in order) rult,blct,downstroke,sexmin,
c    sexmax,blowrate,enthru,b.c.p,upstroke
c
c    open end diesel
    do 300 i=1,iult
    ib = result(i+10)+.5
    if(ib.lt.0.or.ib.gt.9999)ib=9999
    if(ityph.eq.1)write (6,13)result(i),ib,result(i+20),
1    result(i+80),result(i+30),iex(i*4-3),iex(i*4-2),result(i+40),
2    iex(i*4-1),iex(i*4),result(i+60),result(i+50)
c    closed end diesel
    if(ityph.eq.2)write (6,13)result(i),ib,result(i+20),
1    result(i+70),result(i+30),iex(i*4-3),iex(i*4-2),result(i+40),
2    iex(i*4-1),iex(i*4),result(i+60),result(i+50)
c    alrstm
    if(ityph.eq.3)write (6,33)result(i),ib,result(i+20),
1    result(i+30),iex(i*4-3),iex(i*4-2),result(i+40),
2    iex(i*4-1),iex(i*4),result(i+60)
300  continue
    return
    end
    subroutine number
    return
    end
    subroutine plot
    return
    end

```

```
subroutine plots
return
end
subroutine where
return
end
subroutine symbol
return
end
#eor
#eof
```

Appendix B - PILE Input Form

A complete input form is provided. The necessary parameters must be inputted within the specified spaces and in the sequence provided. Additional copies of the form should be made for future use.

PILE-Static and Dynamic Analysis of Piles Input Form

1.000	NSTRAT	B	APNT	ASHFT	LEN	QPRMT	SHEAR	PRCNT	INSTL	NEQN	WAV

1.101	SOIL	XL	N	NCORR	PHI	NQ	CU	GAM	ZC	KS	DEL	ALPH	NC	CONSL	OCR
.															
1.110															

2.001	EQN	W	H	C	EF	N	WP	C1	C3	EP	FS
.											
2.005											

3.000	NSEG	NLAYER	ATP	E	RP

3.101	NELM	NSTRAT	ALPH	A	QS	SJ
.						
3.199						

[illegible]

Title

3.200

[illegible]

IPMI

SYLL

HOLMES

RCS

AM 15

381 SS

NCR

ISP

2

IPEL

FUEL

STR

MR IC

III

2

noi

3.300

[illegible]

Wt. Cap (klps)	Stiff Cap (K/in)	Stiff Cushion (K/in)
10	10	10
20	20	20
30	30	30
40	40	40
50	50	50
60	60	60
70	70	70
80	80	80
90	90	90
100	100	100
110	110	110
120	120	120
130	130	130
140	140	140
150	150	150
160	160	160
170	170	170
180	180	180
190	190	190
200	200	200
210	210	210
220	220	220
230	230	230
240	240	240
250	250	250
260	260	260
270	270	270
280	280	280
290	290	290
300	300	300
310	310	310
320	320	320
330	330	330
340	340	340
350	350	350
360	360	360
370	370	370
380	380	380
390	390	390
400	400	400
410	410	410
420	420	420
430	430	430
440	440	440
450	450	450
460	460	460
470	470	470
480	480	480
490	490	490
500	500	500
510	510	510
520	520	520
530	530	530
540	540	540
550	550	550
560	560	560
570	570	570
580	580	580
590	590	590
600	600	600
610	610	610
620	620	620
630	630	630
640	640	640
650	650	650
660	660	660
670	670	670
680	680	680
690	690	690
700	700	700
710	710	710
720	720	720
730	730	730
740	740	740
750	750	750
760	760	760
770	770	770
780	780	780
790	790	790
800	800	800
810	810	810
820	820	820
830	830	830
840	840	840
850	850	850
860	860	860
870	870	870
880	880	880
890	890	890
900	900	900
910	910	910
920	920	920
930	930	930
940	940	940
950	950	950
960	960	960
970	970	970
980	980	980
990	990	990
1000	1000	1000

3.400

Coefficiente di Restituzione

[illegible]

TDEL
(00C)

TEMAX
(280C)

Cushion

Pile Top

(D1000)

(Diesel
r/Steam)

3.500

Quake
Skin (ln)

Quake
Too (In)

Damping Skin

Damping Too

RULT
(tons)r. of Rest.
of SoilStroke
(11)

**Hammer
Efficiency**

Steam
Pressure (p)

Reaction wt.
(kips)

3.600

Appendix C - Examples

Two examples are provided which illustrate most of the options available to the user of PILE. The examples may also be used to verify computations after initial implementation of the program.

Example 1 - Cohesionless Soils

The pile considered is a close-end steel-pipe pile, 80 ft long and 12.75 in. in diameter driven into cohesionless soil strata.

The subsurface conditions are modeled into four strata with the following parameters assumed or derived from established correlations:

(a) stratum thickness (XL):

$$XL_1 = 8.0 \text{ ft}$$

$$XL_2 = 52.0 \text{ ft}$$

$$XL_3 = 8.0 \text{ ft}$$

$$XL_4 = 30.0 \text{ ft}$$

(b) standard penetration values (N):

$$N_1 = 4$$

$$N_2 = 16$$

$$N_3 = 10$$

$$N_4 = 20$$

(c) angle of internal friction (PHI):

$$\phi_1 = 25$$

$$\phi_2 = 32$$

$$\phi_3 = 30$$

$$\phi_4 = 33$$

(d) unit weight of soil (GAM)

$$\gamma_1 = 110 \text{ lbs/ft}^3$$

(e) critical depth ratio (ZC)

$$ZC_1 = 5.0$$

The pile is driven with a Vulcan II hammer. The five dynamic formulas will be computed for final set and blow counts.

A wave equation analysis will be performed. The pile-soil system is discretized into 20 equal elements of 4.0 ft. The following parameters are assumed:

(a) elastic modulus (E):

$$E (\text{steel}) = 30000. \text{ ksi}$$

(b) unit weight of material (RP):

$$RP = 492. \text{ lbs/ft}^3$$

(c) soil quake (QS):

$$QS = 0.10 \text{ in. (all layers + point)}$$

(d) Smith damping factor (SJ):

$$SJ_1 = .10$$

$$SJ_2 = .05$$

$$SJ_3 = .10$$

$$SJ_4 = .15$$

(e) cushion stiffness

$$\text{plywood} = 3192. \text{ ksi}$$

(f) hammer efficiency (EF)

$$EF = 0.60$$

NEON WAV

OCR

OCR

5
4

22

5

[illegible]

IOUT	IJJ	IHAMR	IOSTR	IFUEL	IPEL	N	ISPL	NCROSS	IBEDAM	IPERCS	ISMITH	ITYS	IPH1
					2	06.		0	1-		✓	2-	

Wt. Cap (klps)	Stiff Cap (K/in)	Stiff Cushion (K/in)
1.0		3.75

	Anvil (Diesel)	Cap (Diesel)	Pile Top	Cushion	TEMAX (msec)	TDEL (sec)
Cap (Air/Steam) Assembly (A/S)						

Cap (Air/Steam) Assembly (A/S)	Anvil (Diesel)	Cap (Diesel)	Pile Top	Cushion	TEMAX (msec)	TDEL (sec)
✓	✓	✓	1.0	0.25		

Quake Skin (in)	Quake Toe (in)	Damping Skin	Damping Toe	RULT (tees)	Coeff. of Rest. of Soil	Stroke (ft)	Hammer Efficiency	Steam Pressure (psi)	Reaction wt. (klps)
							0.20		

purdue university

ultimate static pile capacity

pile - soil model

soil data

shear strength : drained - long term condition

number of strata = 4

1 stratum thickness = 8.00 ft. angle of internal friction = 25.00 deg. undrained shear strength = 0 tsf.
average n value = 4 b/ft. (4) bearing capacity factor, nc = 0
unit weight = 110.00 lbs/cu.ft. coef. of earth pressure, ks = .58 coef. of earth pressure, ko = 0
critical depth r = 5.00 angle of shaft friction = 25.00 deg. adhesion coefficient = 0

2 stratum thickness = 52.00 ft. angle of internal friction = 32.00 deg. undrained shear strength = 0 tsf.
average n value = 16 b/ft. (16)

3 stratum thickness = 8.00 ft. angle of internal friction = 30.00 deg. undrained shear strength = 0 tsf.
average n value = 10 b/ft. (10)

4 stratum thickness = 30.00 ft. angle of internal friction = 33.00 deg. undrained shear strength = 0 tsf.
average n value = 20 b/ft. (17)

pile data

method of installation : driven

point width = 1.062500 ft. point area = .886641 sq.ft. shaft area = 3.337940 sq.ft.

ultimate point capacity = 60.29 tons

ultimate shaft capacity = 64.08 tons

ultimate total pile capacity = 124.38 tons total depth of penetration = 80.00 feet

1

formula : gates

weight of hammer, $w = 5000$. lbs.
drop of hammer, $h = 36.00$ in. efficiency factor for hammer, $ef = .60$

ultimate dynamic resistance, $ru = 124$. tons
permanent set, $s = .01$ in.
driving resistance, $1.0/s = 1375$. blows/ft.

formula : engineering news

weight of hammer, $w = 5000$. lbs.
drop of hammer, $h = 36.00$ in. factor for energy losses, $c = .10$

ultimate dynamic resistance, $ru = 373$. tons
permanent set, $s = 2.07$ in.
driving resistance, $1.0/s = 6$. blows/ft.

formula : janbu

weight of hammer, $w = 5000$. lbs.
drop of hammer, $h = 36.00$ in. weight of pile, $wp = 34900$. lbs.
modulus of elasticity of pile, $ep = 30000000$. psi. cross-section of pile, $a = 127.68$ sq.in.
 pile length, $l = 960.00$ in.

ultimate dynamic resistance, $ru = 187$. tons
permanent set, $s = .10$ in.
driving resistance, $1.0/s = 116$. blows/ft.

formula : danish

weight of hammer, $w = 5000$. lbs.
drop of hammer, $h = 36.00$ in. cross-section of pile, $a = 127.68$ sq.in.
efficiency factor for hammer, $ef = .60$ pile length, $l = 960.00$ in.
 modulus of elasticity of pile, $ep = 30000000$. psi.

ultimate dynamic resistance, $r_u = 249$. tons
 permanent set, $s = .75$ in.
 driving resistance, $1.0/s = 16$. blows/ft.

formula : hiley

weight of hammer, $w = 5000$. lbs.
 drop of hammer, $h = 36.00$ in.
 efficiency factor for hammer, $ef = .60$
 efficiency factor for impact, $n = .85$

weight of pile, $wp = 34900$. lbs.
 elastic compression of head, $c1 = .16$ in.
 elastic compression of pile, $c2 = .09$ in.
 elastic compression of ground, $c3 = .10$ in.

ultimate dynamic resistance, $r_u = 187$. tons
 permanent set, $s = .32$ in.
 driving resistance, $1.0/s = 38$. blows/ft.

1

w e a p - wave equation analysis for piles

1980 version

example 1

pile description

x bel. top (ft) 0 80.0
 a (sq. in.) 127.7 127.7
 e (ksi) 30000. 30000.
 gamma (lb/cu ft) 492.0 492.0
 stroke (equiv.) 3.0 ft, efficiency .60 , impact velocity 10.8 ft/s

hammer model vulc. 1
 element weight stiffness coeff.
 number (kips) (k/in) restitution
 1 5.000
 cap 1.000 129520.3 1.000
 cushion 3192.0 .250
 pile top 1.000

1

pile properties

pile length= 80. ft., area(at top)= 127.7 s-in
 e. modul(at top)=30000. ksi., spec. wt.(at top)= 492. lbs/cu ft

	weight (kips)	stiffn. (k/in)	pdamp. (ks/ft)	splice (kips)	soil-s	soil-d (s/ft)	quake (in)	l.b.t. (ft)
1	1.745	79797.	0	0	.003	.100	.100	4.0
2	1.745	79797.	0	-5000.	.006	.100	.100	8.0
3	1.745	79797.	0	-5000.	.011	.050	.100	12.0
4	1.745	79797.	0	-5000.	.022	.050	.100	16.0
5	1.745	79797.	0	-5000.	.032	.050	.100	20.0
6	1.745	79797.	0	-5000.	.034	.050	.100	24.0
7	1.745	79797.	0	-5000.	.034	.050	.100	28.0
8	1.745	79797.	0	-5000.	.034	.050	.100	32.0
9	1.745	79797.	0	-5000.	.034	.050	.100	36.0
10	1.745	79797.	0	-5000.	.034	.050	.100	40.0
11	1.745	79797.	0	-5000.	.034	.050	.100	44.0
12	1.745	79797.	0	-5000.	.034	.050	.100	48.0
13	1.745	79797.	0	-5000.	.033	.050	.100	52.0
14	1.745	79797.	0	-5000.	.026	.050	.100	56.0
15	1.745	79797.	0	-5000.	.020	.050	.100	60.0
16	1.745	79797.	0	-5000.	.014	.100	.100	64.0
17	1.745	79797.	0	-5000.	.016	.100	.100	68.0
18	1.745	79797.	0	-5000.	.019	.050	.100	72.0
19	1.745	79797.	0	-5000.	.031	.050	.100	76.0
20	1.745	79797.	0	-5000.	.041	.050	.100	80.0
toe					.485	.150	.100	

coefficient of restitution of soil 1.000

options and specifications

phi	1.60	s-damping	smith	rwt (kips)	-0
iout	-0	p-damping	0	soil dist. no.	-2
ifuel	1	j skin	0	tdel (sec.)	-0
iostr	0	j toe	0	temax (ms)	-0
		time incr. (ms)	.088		

1 rult= 124.4, at toe= 60.3 tons

0. table of extrema, time step

Ono.	fmin, jmn (k)	fmax, jmx (k)	strmin, jsn (ksi)	strmax, jsx (ksi)	vmax, jvx (f/s)	dmax, jdx (in)
1	0, 0	493.8, 66	0, 0	3.87, 66	2.5, 153	.204, 201
2	-118.5, 147	496.0, 69	-.93, 147	3.88, 69	2.3, 72	.203, 201
3	-211.8, 144	494.4, 72	-1.66, 144	3.87, 72	2.3, 72	.202, 195
4	-276.7, 144	493.7, 75	-2.17, 144	3.87, 75	2.3, 75	.201, 195
5	-302.0, 144	491.2, 78	-2.37, 144	3.85, 78	2.3, 78	.199, 195
6	-319.0, 141	487.9, 81	-2.50, 141	3.82, 81	2.3, 81	.195, 198
7	-316.9, 138	490.2, 81	-2.48, 138	3.84, 81	2.3, 84	.192, 201
8	-298.8, 135	493.2, 84	-2.34, 135	3.86, 84	2.3, 87	.188, 201
9	-274.5, 132	495.5, 87	-2.15, 132	3.88, 87	2.2, 90	.184, 204
10	-295.6, 153	497.2, 90	-2.32, 153	3.89, 90	2.2, 93	.179, 207
11	-297.2, 150	497.6, 93	-2.33, 150	3.90, 93	2.2, 93	.175, 192
12	-280.2, 147	496.0, 96	-2.19, 147	3.89, 96	2.5, 120	.173, 234
13	-263.7, 144	488.7, 99	-2.07, 144	3.83, 99	2.8, 120	.174, 237
14	-239.2, 141	456.2, 102	-1.87, 141	3.57, 102	2.6, 120	.175, 237
15	-229.9, 138	423.3, 192	-1.80, 138	3.32, 192	3.2, 108	.174, 231
16	-166.9, 138	423.7, 195	-1.31, 138	3.32, 195	3.5, 111	.175, 231
17	-77.9, 153	404.2, 90	-.61, 153	3.17, 90	3.6, 111	.175, 231
18	-82.3, 150	372.5, 90	-.64, 150	2.92, 90	3.3, 111	.174, 234
19	-43.6, 150	300.3, 117	-.34, 150	2.35, 117	3.1, 102	.174, 240
20	0, 0	282.9, 117	0, 0	2.22, 117	3.4, 120	.174, 240

1 example

0 summary

r ult	bl ct	stroke(eq.)	minstr i,j	maxstr i,j	energy
tons	bpf	ft	ksi	ksi	ft-kip
, 124.4	162	3.00	-2.50(6,141)	3.90(11, 93)	3.2

The results of the first example for predominantly cohesionless soils show the following:

- (1) Ultimate capacity. The ultimate point, shaft, and total capacity, and total penetration are given as a result of the static analysis on pg. 325.

$$Q_p = 60.29 \text{ tons}$$

$$Q_s = 64.08 \text{ tons}$$

$$Q_u = 124.38 \text{ tons}$$

$$L = 80.0 \text{ feet}$$

- (2) Dynamic formulas. The available formulas are listed on pages 326 and 327 with accompanying criteria for driving which include permanent set and blow counts.
- (3) Wave equation analysis. The output illustrates the model used and the results. The most pertinent results consist of the maximum stresses per pile element and the blow counts for the specified resistance. These results are shown are shown on page 329.

Example 2 - Cohesive Soils

The pile considered is a precast concrete pile, 12 in. x 12 in. and 65 ft long, driven into cohesive soil strata. A static analysis is only considered.

The subsurface conditions consists of a top layer of soft clay underlain by a stiff marl. Standard penetration values (N) were not provided. Undrained shear strength conditions were assumed and the following parameters used:

(a) stratum thickness (XL):

$$XL_1 = 45.0 \text{ ft}$$

$$XL_2 = 60.0 \text{ ft}$$

(b) undrained shear strength (CU):

$$CU_1 = 0.2 \text{ ton/ft}^3$$

$$CU_2 = 2.0 \text{ ton/ft}^3$$

(c) α adhesion factor (ALPHA):

$$ALPHA_1 = 1.0$$

$$ALPHA_2 = 0.5$$

(d) bearing capacity factor (NC):

$$NC = 9.0 \text{ (both layers)}$$

PILE-Static and Dynamic Analysis of Piles Input Form

page 1

NSTRAT	B	APNT	ASHFT	LEN	QPRMT	SHEAR	PRCNT	INSTL	NEQN	WAV
2	1.0	1.0	1.0	0.0		UU		HEI.VEN		

1.000

SOIL	XL	N	NCORR	PHI	NQ	CU	GAM	ZC	KS	DEL	ALPH	NC	CONSL	OCR
ARG	45.0					0.2					1.0MC			
ARG	60.0					2.0					0.5MC			

1.101

1.110

EQN	W	H	C	EF	N	WP	CI	C3	EP	FS

2.001

2.005

NSEG	NLAYER	ATP	E	RP

3.000

NELM	NSTRAT	ALPH	A	QS	SJ

3.101

3.199

ultimate static pile capacity

ultimate static pile capacity

ultimate static pile capacity

ultimate static pile capacity

ultimate static pile capacity

ultimate static pile capacity

ultimate static pile capacity

ultimate static pile capacity

ultimate static pile capacity

ultimate static pile capacity

ultimate static pile capacity

ultimate static pile capacity

ultimate static pile capacity

ultimate static pile capacity

ultimate static pile capacity

Similarly, the ultimate capacity is divided into point, shaft, and total ultimate capacity. Values of total penetration, ultimate point, shaft, and total pile capacity are shown as output on page 333.

$$L = 65.0 \text{ feet}$$

$$Q_p = 18.0 \text{ tons}$$

$$Q_s = 95.6 \text{ tons}$$

$$Q_u = 113.6 \text{ tons}$$

COVER DESIGN BY ALDO GIORGINI