AD-A054 955 RAND CORP SANTA MONICA CALIF F/G 9/2

HAND CALCULATOR PROGRAMS FOR STAFF OFFICERS. (U)

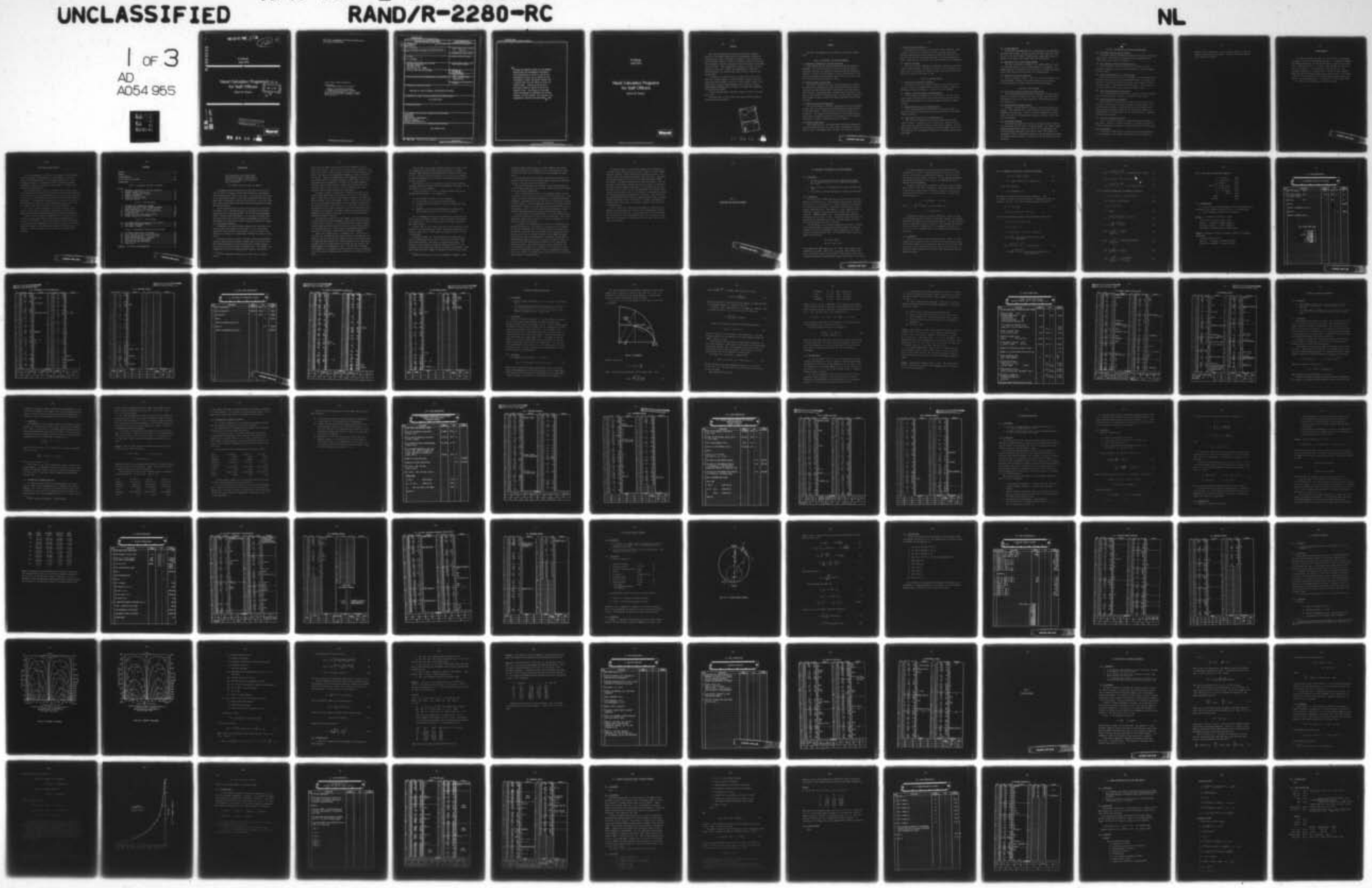
APR 78 E W PAXSON

UNCLASSIFIED RAND/R-2280-RC NL

1 OF 3

AD

A054 955



R-2280-RC
April 1978

Hand Calculator Programs for Staff Officers

Edwin W. Paxson

Rand
SANTA MONICA, CA. 90406

This research was supported by The Rand Corporation as part of its program of public service.

Library of Congress Cataloging in Publication Data

Paxson, Edwin W
Hand calculator programs for staff officers.

([Report] - Rand Corporation ; R-2280-RC)
1. Military art and science--Data processing.
2. HP-67/97 (Calculating-machine) I. Title.
II. Series: Rand Corporation. Rand report ; R-2280-RC.
AS36.R3 R-2280 [U104] 081s [355'.0028'54] 78-7310
ISBN 0-8330-0020-9

Published by The Rand Corporation

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (14) RAND/R-2280-RC	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) Hand Calculator Programs for Staff Officers.		5. TYPE OF REPORT & PERIOD COVERED (9) Interim rept.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) E. W./Paxson		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Rand Corporation 1700 Main Street Santa Monica, Ca. 90406		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE (11) Apr 1978
		13. NUMBER OF PAGES 221 (12) 222 p.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

No restrictions

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Calculators
Computation
Applications of Mathematics
Computer Programs
Hewlett-Packard HP-67/97

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

see reverse side

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE 296600 UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Provides 23 programs written for the Hewlett-Packard HP-67/97 programmable calculators. Full documentation is given to clarify the background of topics and to enable the user to program a subject of special interest for a machine other than the HP-67 but with comparable power. Several programs reduce published volumes of tables to one or two magnetic cards. The programs are grouped under the headings of geographic and orbital programs, military models, cost programs, and mathematical functions and algorithms. (WH)

R-2280-RC
April 1978

Hand Calculator Programs for Staff Officers

Edwin W. Paxson

Rand
SANTA MONICA, CA. 90406

PREFACE

This report documents and discusses twenty-three programs--written for the Hewlett-Packard HP-67/97 programmable calculators--covering a wide range of problems of interest to staff officers in all the military services. Using this material, the user may quickly obtain answers to specific questions arising in meetings, at the desk, or in the field. Full documentation is given to clarify the background of a topic and to enable the programming of a subject of special interest for a machine other than the HP-67, but with comparable power.

In general, the report avoids the "slide-rule" type of topic where only a given formula is to be evaluated. Rather, topics are chosen that would consume too much of a staff officer's time to program because the underlying mathematics may be obscure, because approximating techniques must be sought, or because the programming itself presents problems.

Several programs reduce published volumes of tables to one magnetic card.

The major part of this research was supported by The Rand Corporation from its own funds.

SUMMARY

The report is summarized by an overview of the topics covered in it.

Part I. Geographic and Orbital Programs

1. Geographic Coordinates to UTM and Conversely

Army tactical maps use Universal Transverse Mercator (UTM) coordinates. For joint operations with the Air Force and the Navy, coordinate conversion to geographic coordinates and the converse is essential. Accuracy of this program is better than 10 meters in the *northing* (distance from the equator) and 1 meter in the *easting* (distance from the central meridian of a zone).

2. Sunrise, Sunset, and Twilight

The times of sunrise and of the various categories of twilight are important in planning many types of military operations and activities, although adverse weather conditions all too often vitiate such planning. This program gives twilight times for any day of the year, at any latitude and longitude, and at any altitude. Accuracy is three minutes or less, except under special conditions such as high latitudes.

3. Geodetic Distances and Bearings

The usual formulas of spherical trigonometry that are programmed to give great circle distances and bearings employ a spherical earth of some mean radius. Distances can be in error by as much as 20 kilometers. The program here uses formulas of the National Geodetic Survey based on Bessel's solution for the geodesic on an ellipsoid of revolution. Accuracy is good, about 0.1" or 3 meters.

4. Reentry Trajectories

The program uses Sec. 20 (Fourth-Order Differential Equations). For a body with zero lift and a given "beta" entering the upper atmosphere, find the subsequent range, altitude, and velocity to impact.

PRECEDING PAGE BLANK

5. Satellite Orbital Elements

This program solves two of many possible orbital problems: Given a satellite's injection altitude, velocity, and flight path angle, find the remaining six orbital elements; or given the injection altitude and the altitudes of perigee and apogee, find the remaining elements. Equations are provided so that other problems may be programmed.

6. Satellite Tracking

Given the time and longitude of equatorial crossing of a satellite, select a ground station. Determine if the orbit can be viewed on that pass, and if so, determine its range, bearing, and elevation from local horizon to horizon as functions of time.

Part II. Military Models

7. The Deer Hunt (Defenseless Bombers)

The model assesses the expected outcome of a time-limited battle in which a group of armament-limited interceptors engages a group of defenseless penetrating bombers. A deer hunt is the paradigm.

8. A Bomber Penetration Model (Defended Bombers)

In this model, the bombers are not defenseless. As part of mission planning, bombers divide their payloads between defense missiles and ground attack munitions to maximize weapons delivered to ground targets.

9. Damage Probabilities, PVN and QVN Targets

The program gives damage probabilities for nuclear weapons of given yield and CEP applied against PVN and QVN targets at the optimal airburst altitude.

10. Four Deuces (Precision 4.2-inch Mortar Fire)

This section is an example of data-table replacement by functional fitting. It applies to the 4.2-inch mortar, reducing firing table corrections and meteorological conditions to formulas. The program yields corrected shell charge and corrected azimuth and elevation for precision fire, and permits a difference in altitude between mortar and target.

11. A Laser Equation

The equation programmed applies to propagation in the atmosphere and allows for blockage, thermal blooming, and jitter factors. Given any two of the three primary variables power, range, and average intensity at the target, the program finds the third factor.

12. Shaking the Dice (A War Gaming Example)

This section provides an example of how random numbers are used in a firefight model to assess outcomes quickly in war gaming. The example employs a conceptual mortar round with an on-board heat-seeker sensor that causes the round to home on an armored target.

13. Optimum Allocation of Resources

The title promises too much. This is a topic in nonlinear, convex programming. Military applications arise in search planning, allocating weapons to target classes, and allocating budgets.

Part III. Cost Programs

14. Log-Linear Cumulative Average and Unit Costing

These programs implement the basic assumption of learning curve theory as it applies to production. That is, each time total production doubles, the cost per item reduces to a constant percentage of the previous cost.

15. Time-Phased Procurement Costing

Consider a system, weapon or otherwise, with several major components. Each component has its own lead time and its own, possibly segmented, learning curve. Specify a delivery schedule over future years, and find the New Obligational Authority by fiscal year to support the program.

16. Cost/Benefit Streams

This model deals with the decision to spend money now as opposed to later during the life cycle of a weapon system. For example, should engineering development money be spent now in the expectation that future operating and support costs will be lower? The yardstick is the present value of a discounted stream of cost and benefits (savings). An "internal rate of return" is calculated to provide go-no-go for the decision.

Part IV. Mathematical Functions and Algorithms

17. The Normal Function and Its Inverse

The normal function (probability integral) is pervasive in military calculations. The program is frequently used in conjunction with others, such as that for the Q function.

18. The Q Function (Offset Coverage Function)

The Q function is used in radar detection theory and offset bombing calculations, as well as in calculations of collateral damage to point targets.

19. Linear Programming and 3 × 3 Matrix Games

Many models may be stripped in a meaningful and transparent formulation to three activities as a programming problem, or to three own courses of action pitted against an enemy's three courses of action, in order to make a command decision by game theory. This program uses the pivot method and has some interesting indexing aspects.

20. Fourth-Order Differential Equations

This program supports applications to reentry trajectory determination, Lanchester models of combat, and optimal control theory.

21. Curve Families and Mach Numbers

Military data are frequently presented as sets of tables or as families of curves, with a parameter naming the family member. This section suggests methods of representing these data through curve-fitting, using elementary functions. The methods are applied to the determination of best Mach number for the A-7D aircraft on long-range, constant-altitude cruise.

22. Ten-Point Gaussian Integration

This is a utility program for evaluating definite integrals as they arise. Accuracy is usually excellent. For example, incomplete elliptic integrals are computed to eight decimal places by this method.

23. Truth Tables

A calculus of propositions is tailored for ready implementation by the calculator. The program systematically solves problems in symbolic logic, consisting of a set of logical conditions that the atomistic propositions must satisfy. There are real-world applications, usually overlooked.

ACKNOWLEDGMENTS

H. G. Massey provided the models for the sections on time-phased procurement costing and cost/benefit streams. D. C. Kephart programmed the damage probabilities for PVN and QVN targets, based on his earlier work. Lieutenant Colonel R. S. DeLaney, USAF, brought the laser equation to my attention. I am grateful to these men and also to the technical reviewers, W. B. Graham and R. N. Snow, for their comments. Roger Snow went well beyond a meticulous technical review. He provided a more efficient program for the Q function and prepared several flowcharts to clarify program logic. I am grateful for his collaboration.

Of course, errors found are to be laid to my door.

TO THE USER OF THIS REPORT

If your temperament is like that of the author, this description of the psychology of programming will sound familiar: After the usual time-consuming process of getting the mathematics of a topic in shape, the urge is to program as quickly as possible and make independent checks of the validity of the outputs. It works! And we move on to something else.

Any program, however, certainly including those in this report, can be improved--can be shortened and made more elegant and transparent. The result of this product-improvement effort may be to reduce the running time and to find program and storage space to extend the program's capability. A reexamination of program logic is part of this effort. The program may be made more robust, minimizing operator errors that occur when complex input operations are otherwise required.

If you as a user are interested in a particular topic in this report, you may choose to make this extra effort, which will be repaid with an enriched understanding of hand calculator programming.

Finally, you are invited to communicate to the author any errors you detect, errors and unforeseen restrictions being inevitable in a report of this nature. You are also invited to send to the author, for possible future programming, descriptions of topics that you feel may be of interest to some significant subset of the staff officer community. And by all means, copies of your own programs would be welcomed.

CONTENTS

INTRODUCTION

> *"The general who wins a battle makes many calculations in his temple ere the battle is fought. The general who loses a battle makes but few calculations beforehand."*
>
> \- Sun Tzü Wu, *The Art of War*, ca. 500 B.C.

Programmable hand calculators are little more than five years old, but they are already in their third generation, the gestation period being one and a half to two years. Up to now, they are unique in our inflationary world, in that each new generation has much more power than its predecessor, but sells for much less. This cost trend may reverse should these calculators become more competitive in power with microprocessors.

We can assess the impact in the civilian sector by noting that the number of user programs submitted to the Hewlett-Packard HP-67 program library is approaching 3000. The PPC (Personal Programmers Club),* with more than 2500 members, is a nonprofit worldwide group of people who own and use PPCs (personal programmable calculators). The monthly club newsletter contains a wealth of programs and imaginative programming techniques.

Remembering that modern digital computers were initiated by the military under the pressures of World War II, it is curious that these PPCs, these powerful little animals, are not as equally widespread in the service of the Department of Defense as in the civilian sector.

The PPCs *are* used, of course. The Joint Technical Coordinating Group for Munitions Effectiveness (JTCC/ME) under the JCS has had 25 HP-67 programs prepared for mission planning by squadron ordnance officers in the Air Force, Navy, and Marine Corps. The Strategic Air Command uses the HP-65 in bombing mission planning. Some System

---

*2541 W. Camden Place, Santa Ana, CA 92704; Attn: Richard Nelson.

Project Offices (SPOs), such as the F-16 SPO at Edwards Air Force Base, use the HP-67. Junior officers are using their own funds to purchase PPCs, which in some cases must represent a tradeoff against a new TV set. But there is no recognizable community of users in the military sector. There is no mechanism--no clearinghouses like those in the civilian world--to exchange programs, to share ideas, and to state requirements for new programs. The notion of a loosely organized "national security users group" to achieve these implied objectives naturally comes to mind. We hope that this report may have some catalytic effect in accelerating such a development.

The hand calculator is particularly suited for military use because so many applications can be made in the field or in a meeting where a senior officer wants a quick answer to support a decision, or where a briefer is to be confounded. But for field use the calculator as *currently* designed would probably not meet military specifications. The operating range for the HP-67 is 10° to 40°C (50° to 104°F) and the battery pack life under continuous use is about three hours before recharging or replacement is required. However, current machines are compatible with avionics, producing little or no interference with sensitive electronic circuits.

But powerful as they are in their domain, the PPCs are far from a final answer to personal computing, although this statement depends on their future evolution. In preparing this report, many instances occurred where much more storage than available was needed and where it was frustrating not to have available a programming capability of more lines of code with a higher-level interpretive language.

Again, the civilian sector is leading the way. More than 120 companies are now manufacturing microprocessors with peripherals for home use, and more than 900 home computer dealers in the United States are marketing these machines at relatively modest prices. Memory may be added, there is keyboard input and cathode ray tube display, with BASIC apparently the language of choice. The military is lagging, even though it is a reasonable bet that many staff officers would like to be freed from the computing-center bureaucracy in doing their daily jobs.

But once more, too strong a position should not be taken. Military computing in general requires large main frames to support extremely large data bases and programs with a million or more lines of code. One would certainly hesitate to try to use a microcomputer for logistic management or for solving three-dimensional partial differential equations.

Nevertheless, there is a real gap in the spectrum of required computing capability to meet military requirements, a gap whose filling this report can only adumbrate.

A word of apology is in order. Recorded program cards are not provided with this report. The reasons are:

- No recipient is likely to use all programs;
- The per-copy cost of the report would be high;
- It requires 10 to 20 minutes to key in a program and check it; and
- Hopefully, the keyer will understand the program and be able to modify or tailor it to his or her desires.

It is recommended that users step through the illustrative problems to get the mechanics straight. And it is always a good idea to do a problem twice. Errors in keying are easy to make, especially when under pressure.

Finally, what is to be said to the staff officer who wants to program his or her own problems on a PPC? The natural question for the officer to ask first is: What bounds a problem that can be "fitted" to the machine?

The general answer is: If the problem can be formulated as a *chain* of subproblems, each of which is within the machine's coding and storage capability, then there is in principle no bound. For example, in the prediction of tides by harmonic analysis,* 37 constituents (cosine terms) each with three constants are employed. Since these terms need only be added, one program card and five data cards,

---

*Special Publication No. 98, U.S. Department of Commerce, 1940.

used successively, would suffice. As other examples, six linear algebraic equations in six unknowns can be solved using both sides of two cards, and a Star Trek battle can be programmed with eight cards.

For problems that can be chained, the practical limitation is execution time, which can be long and hardly acceptable if many problems are to be run, as in tidal prediction.

But not all problems can be chained. Operations with matrices of order higher than five, and solutions of partial differential equations, are usually nonchainable.

Even if a problem *should* fit, it is frequently hard to see how to make it actually conform to the calculator's Procrustean bed. This could be because the underlying mathematics, including approximating techniques, is beyond one's reach. The help of a specialist colleague is then essential. Once this mathematical hurdle is cleared, programming--which is really an art form with personal brush strokes--can be exasperating. Advice? Read and understand good programs, as many as possible--something that few of us have the self-discipline to do.

As a postscript to this Introduction, an as yet unexploited area of the military application of PPCs should be mentioned. Two or more people may operate their calculators in parallel, engaging in a cooperative, interactive exercise.

For example, two submarines may be allies in a simulated battle against one enemy boat. The purpose of the exercise is to examine, by repeated simulation runs, the tactical utility of communications between the two friendly boats during the battle--ranging from none, through restricted, to complete information and command exchanges. Each player has his own program which, by sampling from probability distributions, shows the output of his sensor systems in respect to target position and bearing, and the damage, if any, inflicted by ordnance launched. Each player keeps his own log and battle plot. At each battle increment (say, 15 minutes of real time), the calculators may be physically exchanged so that, as appropriate, information can be entered in assigned storage registers, and the calculators then returned to the right boats.

As another example, a War College seminar may be examining the cost implications over the next ten years or more of various possible strategic postures. Weapon systems may be phased in and out. New systems require research and development monies and time. In general, each weapon system has cost profiles of funds required for RDT&E, procurement, and annual maintenance and operating expenses. The cost envelope of each weapons system with respect to time is calculated by the seminar member assigned that system. All programs are the same, differing only in their cost and time parameters. The seminar leader totals the year-by-year costs of all systems in the posture and checks for feasibility against an assumed yearly ceiling. After discussion, the seminar members revise phasing or numbers procured and go through another iteration to see if the ceiling is reached or exceeded, and to determine if the posture is balanced in regard to the threat and required missions.

These examples have indeed been programmed for interactive computing on large computers; but this is time-consuming and facilities may not be readily available. The suggested use of PPCs in parallel is an option that can be implemented quickly and can provide a shakedown for more sophisticated approaches, which in some cases may prove not to be warranted.

PART I

GEOGRAPHIC AND ORBITAL PROGRAMS

1. GEOGRAPHIC COORDINATES TO UTM AND CONVERSELY

1.1. REFERENCES

a. *Universal Transverse Mercator Grid,* AMS Technical Manual No. 19, Army Map Service, Corps of Engineers, Washington, D.C., 1952.

b. *Map Projections*, P. Richards and R. K. Adler, North-Holland, 1972.

c. *Map Reading,* FM 21-26, Department of the Army, October 1960.

1.2. DISCUSSION

Tactical-scale (1:50000) Army maps use the Universal Transverse Mercator Grid (see Ref. c). The map borders show latitude and longitude ticks, but it is difficult to locate the geographic coordinates of a point with any precision. Conversely, Air Force maps use geographic coordinates only. Consequently, in joint operations such as targeting, coordinate conversion from one system to the other is essential. FORTRAN programs exist. The program used at The Rand Corporation has 132 lines of code. Although perfectly adapted to the preparation of coordinates for a list of agreed targets, it hardly meets the requirements of ad hoc field use.

The UTM system covers the world between 80°S and 84°N. Starting at the 180° meridian of longitude and moving eastward, the globe is divided in zones 6° of longitude in width, numbered 1 to 60. Each zone has a central meridian (CM). The following formulas relate zone number (ZN) to the CM:

$$ZN = (CM + 183)/6$$

$$CM = (6 \cdot ZN) - 183$$

For example, Fort Knox, Kentucky is about 86°W. Hence the ZN is the rounded value of (180 - 86)/6, ZN = 16, and CM = -87 or 87°W. (See Ref. c for further details on lettering 8° zones south to north, and on double-lettering for 100,000 meter squares within each 6° × 8° block.)

The value assigned to the CM in each zone is 500,000 meters, called the *false easting*. Hence locations in a zone west of the CM have an *easting* less than 500,000 and conversely. The *northing* is the distance from the equator in meters. For the Southern Hemisphere, the equator is assigned a *false northing* of 10,000,000 meters and numbers decrease southward.

The major complication in coordinate conversion is that allowance must be made for the earth's oblateness. Hence the equatorial radius a and the polar radius b must be selected. Actually, a and the reciprocal of the flattening $f = (a - b)/a$ are given. For the International Spheroid,

$$a = 6\ 378\ 388 \text{ m} , \quad 1/f = 297 .$$

Since $f = 1 - \sqrt{1 - \varepsilon^2}$, where ε is the eccentricity,

$$\varepsilon^2 = 0.006\ 722\ 67 .$$

Unfortunately, different spheroids (different a and f) are used for different areas of the world, for historical reasons. For example, the Clarke 1866 spheroid is used for North America. The other spheroids used are Clarke 1880, Everest, and Bessel. The International Spheroid is used for Europe. (Consult Ref. a.) Consequently, the data a, ε^2, $n = (a - b)/(a + b)$ used here have to be changed for certain parts of the world.

1.3. EQUATIONS

The full equations for the conversions (Refs. a and b) are quite lengthy because extreme accuracy is desired in surveying applications. For military purposes, it is possible to dock the tails of these formulas and still get accuracies better than 1 meter in the *eastings* (E') and better than 10 meters in the *northings* (N)--the distance from the equator in meters.

1.3.1. Geographic Coordinates to UTM Grid Coordinates

$$N = (I) + (II)p^2 + (III)p^4 \qquad (1)$$

$$E' = (IV)p + (V)p^3 \geq 0, \; E = 500\,000 \pm E' . \qquad (2)$$

South of the equator,

$$\overline{N} = 10\,000\,000 - N . \qquad (3)$$

The given coordinates are latitude ϕ and longitude λ. Then $p = 0.0001 \cdot \Delta\lambda$, where $\Delta\lambda$ is the difference of longitude from the CM, *measured in seconds*. E' is the (positive) distance from the CM.

$$(I) = S \cdot k_0, \text{ where} \qquad (4)$$

$$S = A\,\phi - B \sin 2\,\phi + C \sin 4\,\phi .$$

S is the true meridional distance from the equator in meters and

$$A = a[1 - n + 0.75\,n^2(1 - n)]$$

$$B = 1.5\,an(1 - n)$$

$$C = 0.9375\,an^2(1 - n), \; n = (a - b)/(a + b)$$

$k_0 = 0.9996$, the central scale factor to reduce distortion.

$$(II) = \frac{k_0 \sin^2 1'' \cdot 10^8}{4} \cdot \upsilon \sin 2\,\phi, \text{ where} \qquad (5)$$

$\upsilon = a/\sqrt{1 - \varepsilon^2 \sin^2 \phi}$ is the radius of curvature in the prime vertical.

$$\text{(III)} = \frac{k_0 \sin^4 1'' \cdot 10^{16}}{24} \cdot \upsilon \sin \phi \cos^3 \phi (5 - \tan^2 \phi) \tag{6}$$

$$\text{(IV)} = k_0 \sin 1'' \cdot 10^4 \cdot \upsilon \cos \phi \tag{7}$$

$$\text{(V)} = \frac{k_0 \sin^3 1'' \cdot 10^{12}}{6} \cdot \upsilon \cos \phi (2 \cos^2 \phi - 1) \tag{8}$$

1.3.2. UTM Grid Coordinates to Geographic Coordinates

$$\phi = \phi' - [\text{(VII)}\, q^2 - \text{(VIII)}\, q^4]/3600 \tag{9}$$

$$\Delta\lambda = [\text{(IX)}\, q - \text{(X)}\, q^3]/3600 \tag{10}$$

$$q = E' \cdot 10^{-6} \geq 0$$

$$\phi'' = N/Ak_0 \tag{11}$$

$$\phi' = \frac{N/k_0 + B \sin 2\phi'' - C \sin 4\phi''}{A} \tag{12}$$

$$\upsilon = a/\sqrt{1 - \varepsilon^2 \sin^2 \phi''} \tag{13}$$

$$\text{(VII)} = \frac{10^{12}}{2k_0^2 \sin 1''} \cdot \frac{\tan \phi'}{\upsilon^2} \tag{14}$$

$$\text{(VIII)} = \frac{10^{24}}{24k_0^4 \sin 1''} \cdot \frac{\tan \phi' (5 + 3 \tan^2 \phi')}{\upsilon^4} \tag{15}$$

$$\text{(IX)} = \frac{10^6}{k_0 \sin 1''} \cdot \frac{1}{\upsilon \cos \phi'} \tag{16}$$

$$\text{(X)} = \frac{10^{18}}{6k_0^3 \sin 1''} \cdot \frac{1 + 2 \tan^2 \phi'}{\upsilon^3 \cos \phi'} \tag{17}$$

1.3.3. Data Card (International Spheroid)

a =	6 378 338	STO Ø
ε^2 =	0.006 722 67	STO 1
k_0 =	0.999 6	STO 3
$10^6 \sin 1''$ =	4.848 136 8	STO 4
A =	6 367 645.45	STO A
B =	16 106.99	STO B
C =	16.976	STO C
3600		STO 5
500 000		STO 6

2.4. PROGRAM NOTES

a. It will be noted that the powers of 10 in the program differ from those in the formulas because $10^6 \sin 1''$ is a stored datum.

b. West longitude is prefixed by a minus sign.

Example 1. N 49°48'00", E 08°24'0" to UTMC.

49.48 STO D, 8.24 STO E, 9 STO 7 (CM)
Press A: Northing = 5516670 (5516677.7)
Press R/S: Easting = 456820 (456819.7)
The numbers in parentheses are the AMS values.

Example 2. Northing = 5516677.7, Easting = 456819.7 to geographic coordinates.

N STO D, E STO E, 9 STO 7
Press A: Latitude = 49.4801 (49°48'01")
Press R/S: Longitude = 8.2360 (8°24'00")

1.5.1 USER INSTRUCTIONS

1 — 1.1 GEOGRAPHIC COORDS TO UTM GRID — 2

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	KEY LAT (D.MS) STO D	49.48	STO	D	49.
2	KEY LONG (D.MS) STO E (- FOR W. LONG.)	8.24	STO	E	8.
3	KEY CM STO 7	9	STO	7	9.
4	PRESS A			A	5516670
	OUTPUT IS NORTHING (M) IN D				
5	PRESS R/S			R/S	456820
	OUTPUT IS EASTING (M) IN E				

INT. GEOID DATA CARD

a	6378388.000	0
ϵ^2	0.00672267	1
k_o	0.99960000	3
$10^6 \sin 1''$	4.84813680	4
	3600.000000	5
	500000.0000	6
A	6367645.450	A
B	16106.99000	B
C	16.97600000	C

PRECEDING PAGE BLANK

1.6.1 GEOGRAPHIC TO UTM COORDINATES

STEP	KEY ENTRY		KEY CODE	COMMENTS	STEP	KEY ENTRY		KEY CODE	COMMENTS
001	001	*LBLA	21 11			057	÷	-24	
	002	RCL7	36 07			058	RCL3	36 03	
	003	RCLE	36 15			059	×	-35	
	004	HMS→	16 36	DEC. DEGS.	060	060	RCL4	36 04	
	005	−	-45			061	X^2	53	
	006	ABS	16 31			062	×	-35	
	007	RCL5	36 05			063	EEX	-23	
	008	×	-35	$\Delta\lambda$ IN SECS		064	4	04	
	009	EEX	-23			065	÷	-24	
010	010	4	04			066	RCLI	36 46	
	011	÷	-24			067	X^2	53	
	012	STOI	35 46	p		068	×	-35	
	013	RCLD	36 14			069	RCL9	36 09	
	014	HMS→	16 36		070	070	+	-55	
	015	STOD	35 14	LAT. IN DEC DEGS		071	STOD	35 14	(I) + (II)p^2
	016	4	04			072	RCL2	36 02	
	017	×	-35			073	TAN	43	
	018	SIN	41			074	X^2	53	
	019	RCLC	36 13			075	5	05	
020	020	×	-35			076	−	-45	
	021	RCLD	36 14			077	CHS	-22	
	022	2	02			078	RCL2	36 02	
	023	×	-35			079	COS	42	
	024	SIN	41		080	080	3	03	
	025	RCLB	36 12			081	Y^x	31	
	026	×	-35			082	×	-35	
	027	−	-45			083	RCL2	36 02	
	028	RCLD	36 14			084	SIN	41	
	029	D→R	16 45	RADIANS		085	×	-35	
030	030	RCLA	36 11			086	RCL8	36 08	ν
	031	×	-35			087	×	-35	
	032	+	-55			088	2	02	
	033	RCL3	36 03			089	4	04	
	034	×	-35		090	090	÷	-24	
	035	STO9	35 09	$k_o S$ (I)		091	RCL3	36 03	
	036	RCLD	36 14			092	×	-35	
	037	STO2	35 02	LAT		093	RCL4	36 04	
	038	SIN	41			094	4	04	
	039	X^2	53			095	Y^x	31	
040	040	RCL1	36 01	ϵ^2		096	×	-35	
	041	×	-35			097	EEX	-23	
	042	1	01			098	8	08	
	043	−	-45			099	÷	-24	
	044	CHS	-22		100	100	RCLI	36 46	
	045	√X	54			101	4	04	
	046	RCL0	36 00			102	Y^x	31	
	047	÷	-24			103	×	-35	(III)p^4
	048	1/X	52			104	RCLD	36 14	
	049	STO8	35 08	ν		105	+	-55	
050	050	RCL2	36 02			106	STOD	35 14	NORTHING
	051	2	02			107	R/S	51	
	052	×	-35			108	RCL2	36 02	
	053	SIN	41			109	COS	42	
	054	RCL8	36 08		110	110	RCL8	36 08	
	055	×	-35			111	×	-35	
	056	4	04			112	RCL3	36 03	

REGISTERS

0	1	2	3	4	5	6	7	8	9
a	ϵ^2	LAT. ϕ	k_o'	$10^6 \sin 1''$	3600	500000	CM	ν	$k_o S$
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I
A	B	C	LAT. ϕ, D	LONG. λ, E	p

1.6.1 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
	113 ×	-35	
	114 RCL4	36 04	
	115 ×	-35	
	116 EEX	-23	
	117 2	02	
	118 ÷	-24	
	119 RCLI	36 46	
120	120 ×	-35	
	121 STO9	35 09	(IV)p
	122 RCL2	36 02	
	123 COS	42	
	124 X^2	53	
	125 2	02	
	126 ×	-35	
	127 1	01	
	128 −	-45	
	129 RCL2	36 02	
130	130 COS	42	
	131 ×	-35	
	132 RCL8	36 08	
	133 ×	-35	
	134 6	06	
	135 ÷	-24	
	136 RCL3	36 03	
	137 ×	-35	
	138 RCL4	36 04	
	139 3	03	
140	140 Y^x	31	
	141 ×	-35	
	142 EEX	-23	
	143 6	06	
	144 ÷	-24	
	145 RCLI	36 46	
	146 3	03	
	147 Y^x	31	
	148 ×	-35	
	149 RCL9	36 09	
150	150 +	-55	
	151 STO9	35 09	E' (2)
	152 RCL7	36 07	
	153 RCLE	36 15	
	154 X≤Y?	16-35	LONG ≤ CM ?
	155 GTOB	22 12	
	156 RCL6	36 06	
	157 RCL9	36 09	
	158 +	-55	
	159 STOE	35 15	EASTING
160	160 RTN	24	
	161 *LBLB	21 12	
	162 RCL6	36 06	
	163 RCL9	36 09	
	164 -	-45	
	165 STOE	35 15	EASTING
	166 RTN	24	

LABELS

A	B	C	D	E
USED	USED			

a	b	c	d	e

0	1	2	3	4

5	6	7	8	9

FLAGS: 0, 1, 2, 3

SET STATUS

FLAGS (ON OFF)	TRIG	DISP
0 ☐ ☐	DEG ☐	FIX ☐
1 ☐ ☐	GRAD ☐	SCI ☐
2 ☐ ☐	RAD ☐	ENG ☐
3 ☐ ☐		n____

1.5.2 UTM USER INSTRUCTIONS

1 1.2 UTM GRID TO GEOGRAPHIC COORDS **2**

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	KEY N (D.MS) STO D	5516677.7	STO	D	SAME
2	KEY E (D.MS) STO E	456819.7	STO	E	SAME
3	KEY CM STO 7	9	STO	7	9.0000
4	PRESS A			A	49.4801
	OUTPUT IS LATITUDE (D.MS) IN D				
5	PRESS R/S			R/S	8.2360
	OUTPUT IS LONGITUDE (D.MS) IN E				8°24'00"

1.6.2 UTM TO GEOGRAPHIC COORDINATES

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	001 *LBLA	21 11	
	002 RCL6	36 06	
	003 RCLE	36 15	E
	004 -	-45	
	005 ABS	16 31	
	006 EEX	-23	
	007 6	06	
	008 ÷	-24	
	009 STOI	35 46	q
010	010 RCLD	36 14	
	011 RCLA	36 11	
	012 ÷	-24	
	013 RCL3	36 03	
	014 ÷	-24	
	015 R→D	16 46	DEGS.
	016 STO9	35 09	ϕ'' (11)
	017 4	04	
	018 ×	-35	
	019 SIN	41	
020	020 RCLC	36 13	
	021 ×	-35	
	022 CHS	-22	
	023 RCL9	36 09	
	024 2	02	
	025 ×	-35	
	026 SIN	41	
	027 RCLB	36 12	
	028 ×	-35	
	029 +	-55	
030	030 RCLD	36 14	
	031 RCL3	36 03	
	032 ÷	-24	
	033 +	-55	
	034 RCLA	36 11	
	035 ÷	-24	
	036 R→D	16 46	DEGS
	037 STO9	35 09	ϕ' (12)
	038 SIN	41	
	039 X²	53	
040	040 RCL1	36 01	
	041 ×	-35	
	042 1	01	
	043 -	-45	
	044 CHS	-22	
	045 √X	54	
	046 RCL0	36 00	
	047 ÷	-24	
	048 1/X	52	
	049 STO8	35 08	ν (13)
050	050 RCL9	36 09	
	051 TAN	43	
	052 X²	53	
	053 3	03	
	054 ×	-35	
	055 5	05	
	056 +	-55	
	057 RCL9	36 09	
	058 TAN	43	
	059 ×	-35	
060	060 RCL8	36 08	
	061 4	04	
	062 Yˣ	31	
	063 ÷	-24	
	064 RCL4	36 04	
	065 ÷	-24	
	066 RCL3	36 03	
	067 4	04	
	068 Yˣ	31	
	069 ÷	-24	
070	070 2	02	
	071 4	04	
	072 ÷	-24	
	073 EEX	-23	
	074 3	03	
	075 0	00	
	076 ×	-35	
	077 RCLI	36 46	
	078 4	04	
	079 Yˣ	31	
080	080 ×	-35	
	081 STO2	35 02	(VIII) q^4
	082 RCL9	36 09	
	083 TAN	43	
	084 RCL8	36 08	
	085 X²	53	
	086 ÷	-24	
	087 RCL4	36 04	
	088 ÷	-24	
	089 RCL3	36 03	
090	090 X²	53	
	091 ÷	-24	
	092 2	02	
	093 ÷	-24	
	094 EEX	-23	
	095 1	01	
	096 8	08	
	097 ×	-35	
	098 RCLI	36 46	q
	099 X²	53	
100	100 ×	-35	
	101 RCL2	36 02	
	102 -	-45	
	103 CHS	-22	
	104 RCL5	36 05	
	105 ÷	-24	
	106 RCL9	36 09	
	107 +	-55	
	108 →HMS	16 35	D.MS
	109 STOD	35 14	ϕ
110	110 R/S	51	DISPLAY
	111 RCL9	36 09	
	112 COS	42	

REGISTERS

0	1	2	3	4	5	6	7	8	9
a	ϵ^2	(VIII) q^4	k_o	$10^6 \sin 1''$	3600	500000	CM	ν	ϕ'
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I
A	B	C	N, ϕ	E, λ	q

1.6.2 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	RCL8	36 08	
114	x	-35	
115	RCL4	36 04	
115	x	-35	
117	RCL3	36 03	
118	x	-35	
119	1/X	52	
120	EEX	-23	
121	1	01	
122	2	02	
123	x	-35	
124	RCLI	36 46	
125	x	-35	
126	STO2	35 02	(IX) q
127	RCL9	36 09	
128	TAN	43	
129	X^2	53	
130	2	02	
131	x	-35	
132	1	01	
133	+	-55	
134	RCL9	36 09	
135	COS	42	
136	÷	-24	
137	RCL8	36 08	
138	3	03	
139	Y^X	31	
140	÷	-24	
141	RCL4	36 04	
142	÷	-24	
143	RCL3	36 03	
144	3	03	
145	Y^X	31	
146	÷	-24	
147	6	06	
148	÷	-24	
149	EEX	-23	
150	2	02	
151	4	04	
152	x	-35	
153	RCLI	36 46	
154	3	03	
155	Y^X	31	
156	x	-35	(X) q^3
157	CHS	-22	
158	RCL2	36 02	
159	+	-55	
160	RCL5	36 05	
161	÷	-24	
162	STO2	35 02	$\Delta\lambda$ (10)
163	RCLE	36 15	
164	RCL6	36 06	
165	-	-45	
166	X<0?	16-45	E- 500000 < 0 ?
167	GTOB	22 12	
168	RCL7	36 07	
169	RCL2	36 02	
170	+	-55	
171	→HMS	16 35	D.MS
172	STOE	35 15	LONG
173	RTN	24	
174	*LBLB	21 12	
175	RCL7	36 07	
176	RCL2	36 02	
177	-	-45	
178	→HMS	16 35	D.MS
179	STOE	35 15	LONG
180	RTN	24	

LABELS					FLAGS
A USED	B USED	C	D	E	0
a	b	c	d	e	1
0	1	2	3	4	2
5	6	7	8	9	3

SET STATUS

FLAGS (ON OFF)	TRIG	DISP
0 ☐ ☐	DEG ☐	FIX ☐
1 ☐ ☐	GRAD ☐	SCI ☐
2 ☐ ☐	RAD ☐	ENG ☐
3 ☐ ☐		n ____

2. SUNRISE, SUNSET, AND TWILIGHT

2.1. REFERENCES

a. Russell, Dugan, and Stewart, *Astronomy*, Ginn & Co., New York, 1945.

b. *The American Ephemeris and Nautical Almanac for the Year 1977*, U.S. Government Printing Office, Washington, D.C., 1976.

c. *Explanatory Supplement to the Astronomical Ephemeris*, Her Majesty's Stationery Office, London, 1961.

2.2. DISCUSSION

Charts are prepared and issued for each major military operation or operational area giving sunlight, moonlight, and tidal data. Nautical twilight (sun's zenith angle from 102° to 96°) provides enough illumination for most types of ground activity, although bomb loading and repair work require artificial light. Civil twilight (sun's zenith angle from 96° to 90°) permits normal day activities such as observed artillery fire and visual bombing. Sunrise occurs when the sun's upper limb has a zenith angle of 90°. This makes the zenith distance of the sun's center 90°50' (90.83°), allowing 34' for horizontal refraction and 16' for the sun's semidiameter. For some aircraft applications, a correction of $1'.17\sqrt{h}$ is added, where h is the altitude in feet. If H is in kilofeet and decimal degrees are used, this correction is $0.617\sqrt{H}$ degrees.

2.3. EQUATIONS

The fundamental relation (Ref. c, p. 403) is

$$\cos h = -\tan\phi \tan\delta + \sec\phi \sec\delta \cos z \ , \qquad (1)$$

where h and δ are the hour angle and declination of the sun at the time of the phenomenon, ϕ is the latitude, and z is the zenith angle. (The correction of the declination from ephemeris noon to approximate rising or setting is at most 0.1°, a refinement we will neglect.)

The sun's declination is the number of degrees the earth's axis departs from a plane that is normal to the sun's direct rays. The declination is tabulated in the annual Ephemeris. An approximate formula is derived and checked against the tables.

In Fig. 2.1, θ is the true anomaly on day $\bar{D}$ and α is the eccentric anomaly.

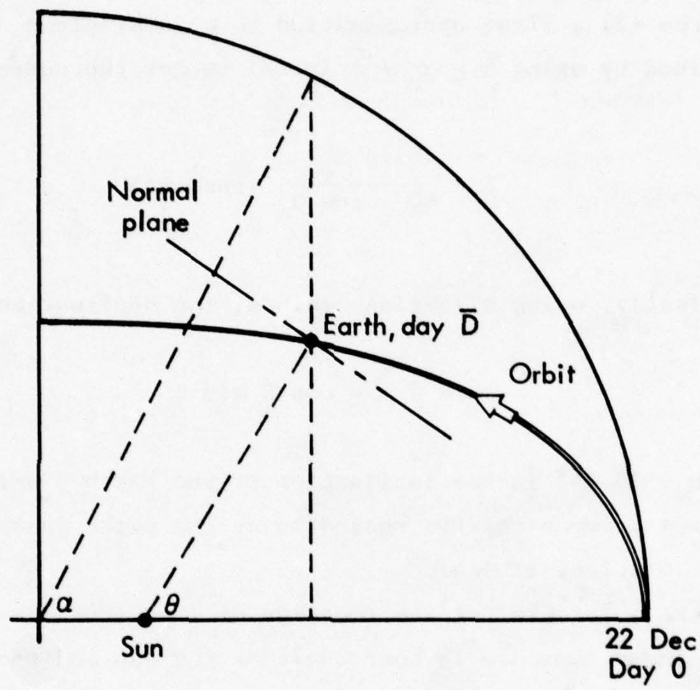


Fig. 2.1— Anomalies

Kepler's equation is

$$\alpha - \varepsilon \sin \alpha = \frac{2\Pi \bar{D}}{365} , \qquad (2)$$

where $\varepsilon \doteq 1/60$ is the eccentricity of the earth's orbit. Also,

$$\sin \alpha = \frac{\sqrt{1 - \varepsilon^2} \sin \theta}{1 + \varepsilon \cos \theta} . \qquad (3)$$

Neglecting $\sqrt{1-\varepsilon^2}$ (= 0.99986), and solving for θ yields

$$\sin\theta = \frac{\sin\alpha}{1 \pm \varepsilon\cos\alpha}, \qquad (4)$$

where the + branch is used from the vernal equinox, 21 March (Day 89), to the autumnal equinox, 23 September (Day 275).

From (2) a first approximation to α is simply $\alpha_o = 2\Pi D/365$. This is refined by using $\alpha = \alpha_o + \Delta$ in (2) to get the correction

$$\Delta = \frac{\sin\alpha_o}{60 - \cos\alpha_o} \text{ (radians)} . \qquad (5)$$

Finally, using direction cosines, the declination becomes

$$\sin\delta = -\cos\theta\sin\upsilon , \qquad (6)$$

where $\upsilon = 23.44°$ is the inclination of the earth's axis. The sign becomes + between the two equinoxes as the earth passes through the summer solstice, 21 June.

Next a formula for the Equation of Time (EOT) is required. The EOT is the difference in hour angle of the sun and the fictitious mean sun used for ordinary time. The difference owes to two causes: (1) the variable motion of the sun because of the eccentricity of the earth's orbit, and (2) the obliquity of the ecliptic.

The figure on p. 147 of Ref. a suggests that

$$EOT = -A\sin(\theta - a) - B\sin(2\theta + b) . \qquad (7)$$

We get from that figure the approximate values A = 8, a = 5.92°, B = 10, b = 4.73°. Here Day 0 (D = 0) is 25 December, since the EOT is 0 on that date.

The extrema of the EOT are:

12 February	D = 49	EOT = -14.29 min
14 May	D = 140	EOT = +3.72 min
26 July	D = 213	EOT = -6.46 min
3 November	D = 313	EOT = +16.41 min .

Replace A by A + ΔA, etc., substitute in (7), take only first-order terms, and get four linear equations in the unknowns ΔA, Δa, ΔB, Δb. These are solved quickly by Program 7 of the Hewlett-Packard Math Pac 1. The corrected values of the parameters are

$$A = 7.4447 \; , \quad a = 5.935 \; , \quad B = 9.894 \; , \quad b = 4.941 \; .$$

The resulting mean absolute error throughout the year with respect to the tabulated values of the EOT is 24 sec.

Rising and setting times are now computed by

$$\begin{aligned} \text{Rising} &= 12 - \text{EOT} - h \\ \text{Setting} &= \text{Rising} + 2h \; . \end{aligned} \tag{8}$$

These are local mean times with respect to the central meridian (CM) of a given time zone. To correct for other longitudes, subtract 4 min for each degree east of the CM, since the sun is earlier, and add 4 min for each degree west of the CM. The correction is programmed.

2.4. PROGRAM NOTES

(1) The day number D ($\bar{D}$ = D + 3) for a given date is needed, counting from Christmas as Day 0. Subtract 1 from the month number and multiply by 30.42, the average number of days in a month. Take the integral part. For month numbers 1, 8, 9, 10, 11, 12, add 6; for months 3, 4, 5, 6, 7, add 5; and for month 2, add 7. Finish by adding the days of the date.

(2) For the longitude correction (f LBL 0) add 360° if West longitude (entered negative). Then obtain the correct central meridian by checking whether the fractional part of the longitude divided by 15 is less than or greater than 0.5 (1/2 hr).

(3) At dates when the sun's declination is close to 0, formula (4) can yield a number very slightly greater in absolute value than 1. This would generate an Error signal. Such a number is replaced by 1 in f LBL 5.

(4) This program required 220 steps. There was not space to program rising and setting in the Southern Hemisphere. To do this, proceed as follows:

- Find the sun's declination on the desired date.
- Change the sign and use formula (6) to get a new θ and a new day number ($365 \cdot \theta/360$), differing by about 6 months.
- Find the rising time for the latter day in the Northern Hemisphere.
- Add $EOT_2 - EOT_1$.

Example. Find sunrise on 5 May at 38°S (Central Meridian). Run program with +38°. RCL 2 to get $\delta = 15.48°$ (the true declination is 16.15°). RCL 1 to get $EOT_1 = .0600$ hr. By (6) with $\delta = -15.48$, $\theta = 47.9$ or 312.1. Choose the latter. D = 317, or 7 November. Run program with date 11.07. Rising time is 6 h 32 m (6.53), $EOT_2 = 0.26$. Rising time is 6.53 + 0.26 - 0.06 = 6.73 or 6 h 43 m. The value for this example given on p. 566 of Ref. (b) is 6 h 45 m.

(5) Program running time is about 25 sec. Compared with the values tabulated in Ref. b (pp. 434ff), errors are 0 to 3 min with 0 and 1 the most likely values. The errors can be greater, however, at high latitudes at dates close to those with twilight lasting all night.

Example. Astronomical twilight, 50°N, 14 July: The computed value is 0 h 31 m versus the actual 0 32 for beginning, and 23 39 versus 23 32 for end of twilight.

2.5 USER INSTRUCTIONS

2. SUNRISE, SUNSET AND TWILIGHT
ZENITH A, DATE B, LAT. C, LONG. D PRESS A, R/S

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	LOAD BOTH SIDES OF PRGM AND DATA CARDS				
2	ZENITH DESIRED STO A				
	SUNRISE (UPPER LIMB) 90.83	90.83		ENT	90.83
	CIVIL TWILIGHT 96				
	NAUTICAL TWILIGHT 102				
	ASTRONOMICAL TWILIGHT 108				
	TO CORRECT FOR ALTITUDE IN KFT				
	ADD 0.617 √H TO THE ABOVE (40 KFT)	3.90		+	94.73
			STO	A	94.73
4	DATE AS MM.DD STO B				
	(N.B. OCT 9 IS 10.09)	10.09	STO	B	10.09
5	LATITUDE (NORTH ONLY)				
	DD.MM STO C 36° 22' N	36.22	STO	C	36.22
6	LONGITUDE ± DDD.MM STO D				
	(- FOR W. LONG) 121° 8' W	-121.08	STO	D	-121.08
7	PRESS A TO GET RISING HH.MM (5h 45m)				5.45
8	PRESS R/S TO GET SETTING HH.MM (17h 57m)				17.57
9	DAY NUMBER OF DATE		RCL	0	291
	RCL 0 AND SUBTRACT 9	9		-	282
10	EQUATION OF TIME				
	DSP 4, RCL 1, g H.MS		RCL	1	0.2128
	TO GET .MMSS (12m 46s)		g	H.MS	.1246
11	DECLINATION OF SUN		RCL	2	-5.9663
	RCL 2, g H.MS to get + DD.MMSS (-5° 58')		g	H.MS	-5.5759
12	RISING IS STORED IN 3		RCL	3	5.7632
	SETTING IS STORED IN 4		g	H.MS	5.4548
	(g H.MS)				
13	ERROR MEANS TWILIGHT LASTS ALL NIGHT				

2.6 SUNRISE, SUNSET, AND TWILIGHT

STEP	KEY ENTRY		KEY CODE	COMMENTS
001	001	*LBLA	21 11	PRGM NOTE 1.
	002	RCLB	36 12	
	003	INT	16 34	MONTH
	004	1	01	
	005	−	-45	
	006	RCL7	36 07	
	007	x	-35	
	008	INT	16 34	D*
	009	STO0	35 00	
010	010	RCLB	36 12	
	011	8	08	
	012	X≤Y?	16-35	
	013	GTO1	22 01	
	014	RCLB	36 12	
	015	3	03	
	016	X≤Y?	16-35	
	017	GTO2	22 02	
	018	RCLB	36 12	
	019	2	02	
020	020	X≤Y?	16-35	MONTH 2 ?
	021	GTO3	22 03	
	022	GTO1	22 01	MONTH 1
	023	*LBL1	21 01	MONTHS 8 to 12
	024	RCL0	36 00	AND 1
	025	6	06	
	026	ST+0	35-55 00	ADD 6
	027	GTO4	22 04	
	028	*LBL2	21 02	MONTHS 3 TO 7
	029	RCL0	36 00	
030	030	5	05	
	031	ST+0	35-55 00	ADD 5
	032	GTO4	22 04	
	033	*LBL3	21 03	MONTH 2
	034	7	07	
	035	ST+0	35-55 00	ADD 7
	036	GTO4	22 04	
	037	*LBL4	21 04	
	038	RCLB	36 12	
	039	FRC	16 44	
040	040	1	01	
	041	0	00	
	042	0	00	
	043	x	-35	
	044	ST+0	35-55 00	DAY D
	045	RCL0	36 00	EQN. OF TIME
	046	RCL9	36 09	
	047	x	-35	θ
	048	P⇄S	16 51	SEC
	049	RCL3	36 03	$-a$
050	050	+	-55	
	051	SIN	41	
	052	RCL1	36 01	−A
	053	x	-35	
	054	P⇄S	16-51	PRI
	055	RCL0	36 00	
	056	RCL9	36 09	

STEP	KEY ENTRY		KEY CODE	COMMENTS
	057	x	-35	
	058	2	02	
	059	x	-35	2θ
060	060	P⇄S	16-51	SEC
	061	RCL4	36 04	
	062	+	-55	
	063	SIN	41	
	064	RCL2	36 02	− B
	065	x	-35	
	066	+	-55	
	067	6	06	
	068	0	00	
	069	÷	-24	
070	070	P⇄S	16-51	
	071	STO1	35 01	EOT IN HRS.
	072	RCL0	36 00	(5)
	073	3	03	
	074	+	-55	
	075	STO0	35 00	$\bar{D}$
	076	RCL9	36 09	
	077	x	-35	
	078	STOE	35 15	ϕ
	079	SIN	41	
080	080	6	06	
	081	0	00	
	082	RCLE	36 15	
	083	COS	42	
	084	−	-45	
	085	÷	-24	
	086	R→D	16 46	Δ RADIANS (5)
	087	RCLE	36 15	
	088	+	-55	
	089	STOE	35 15	ϕ
090	090	GSB7	23 07	
	091	RCLE	36 15	
	092	SIN	41	
	093	RCL6	36 06	
	094	÷	-24	$\sin\theta$ (4)
	095	GSB5	23 05	
	096	SIN⁻¹	16 41	θ
	097	COS	42	
	098	RCL8	36 08	$\sin\nu = \sin 23.44°$
	099	x	-35	
100	100	SIN⁻¹	16 41	(6)
	101	RCLI	36 46	±1
	102	x	-35	$\pm\delta$
	103	STO2	35 02	
	104	RCLA	36 11	z (1)
	105	COS	42	
	106	RCL2	36 02	
	107	COS	42	
	108	÷	-24	
	109	RCLC	36 13	
110	110	HMS→	16 36	ϕ DEC.DEGS
	111	COS	42	
	112	÷	-24	

REGISTERS

0	1	2	3	4	5	6	7	8	9
D*,D,D	EOT (HRS)	$\pm\delta$	RISING	SETTING	h(HRS)	$1\pm\epsilon\cos\phi$	30.42	.3978	360/365

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
	-7.447	-9.894	-5.935	4.941					

A	B	C	D	E	I
ZENITH Z	DATE MM.DD	LAT. DD.MM	LONG. ± DDD.MM	ϕ_0, ϕ	±1, (157)

2.6 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
	113 RCL2	36 02	
	114 TAN	43	tan δ
	115 RCLC	36 13	
	116 HMS→	16 36	
	117 TAN	43	
	118 ×	-35	
	119 −	-45	cos h (1)
120	120 COS⁻¹	16 42	
	121 1	01	
	122 5	05	
	123 ÷	-24	
	124 STO5	35 05	h IN HRS
	125 1	01	
	126 2	02	
	127 RCL1	36 01	
	128 −	-45	RISING (8)
	129 RCL5	36 05	
130	130 −	-45	
	131 STO3	35 03	
	132 GSB9	23 09	
	133 →HMS	16 35	
	134 R/S	51	CORRECTED RISING
	135 RCL3	36 03	
	136 RCL5	36 05	
	137 2	02	
	138 ×	-35	
	139 +	-55	
140	140 STO4	35 04	
	141 →HMS	16 35	CORRECTED
	142 RTN	24	SETTING
	143 *LBL0	21 00	PRGM NOTE 2
	144 RCLD	36 14	
	145 HMS→	16 36	
	146 X>0?	16-44	LONG POS ? (EAST)
	147 GTOa	22 16 11	
	148 3	03	IF NOT, ADD 360
	149 6	06	
150	150 0	00	
	151 +	-55	
	152 GTOa	22 16 11	
	153 *LBLa	21 16 11	
	154 1	01	
	155 5	05	
	156 ÷	-24	
	157 FRC	16 44	FRAC OF LONG/15
	158 STOI	35 46	
	159 •	-62	
160	160 5	05	
	161 X>Y?	16-34	
	162 GTOb	22 16 12	(GTO fb)
	163 RCL3	36 03	E OF CENTRAL MER
	164 RCLI	36 46	(BECAUSE OF DIVISION
	165 1	01	BY 15, ABOVE APPLIES-
	166 −	-45	4 MIN/DEG CORREC-
	167 −	-45	TION.)
	168 STO3	35 03	

STEP	KEY ENTRY	KEY CODE	COMMENTS
	169 RTN	24	(gLBLb)
170	170 *LBLb	21 16 12	
	171 RCL3	36 03	
	172 RCLI	36 46	
	173 −	-45	
	174 STO3	35 03	
	175 RTN	24	
	176 *LBL7	21 07	BRANCH CHECK (4)
	177 RCL0	36 00	$\bar{D}$
	178 9	09	
	179 0	00	
180	180 X>Y?	16-34	$90 > \bar{D}$?
	181 GTO8	22 08	
	182 2	02	
	183 7	07	
	184 5	05	
	185 RCL0	36 00	
	186 X>Y?	16-34	$\bar{D} > 275$?
	187 GTO8	22 08	
	188 GTO9	22 09	
	189 *LBL8	21 08	
190	190 1	01	
	191 RCLE	36 15	
	192 COS	42	
	193 6	06	
	194 0	00	
	195 ÷	-24	
	196 −	-45	USE - SIGN
	197 STO6	35 06	$1 - \epsilon \cos \phi$
	198 1	01	
	199 CHS	-22	
200	200 STOI	35 46	-1 IN R_I
	201 RTN	24	(TO SHOW BRANCH)
	202 *LBL9	21 09	
	203 1	01	
	204 RCLE	36 15	
	205 COS	42	
	206 6	06	
	207 0	00	
	208 ÷	-24	
	209 +	-55	USE + SIGN
210	210 STO6	35 06	$1 + \epsilon \cos \phi$
	211 1	01	
	212 STOI	35 46	+1 IN R_I
	213 RTN	24	(TO SHOW BRANCH)
	214 *LBL5	21 05	PRGM NOTE 3.
	215 ABS	16 31	
	216 1	01	
	217 X⇄Y	-41	
	218 X>Y?	16-34	$\lvert \sin \theta \rvert > 1$?
	219 1	01	REPLACE BY 1
220	220 RTN	24	

NOTE: DATA CARD ENTRIES ARE MARKED ☐ IN REGISTERS.

LABELS		FLAGS	SET STATUS		
D	E	0	FLAGS	TRIG	DISP
d	e	1	ON OFF 0 ☐ ☐	DEG ☐	FIX ☐
3	4	2	1 ☐ ☐	GRAD ☐	SCI ☐
8	9	3	2 ☐ ☐ 3 ☐ ☐	RAD ☐	ENG ☐ n___

3. GEODETIC DISTANCES AND BEARINGS

3.1. REFERENCES

a. Erwin Schmid, *Triangulation Position Computation Without Tabulations*, unpublished Ms., National Geodetic Survey, July 1972.

b. P. A. Smith and H. G. Massey, *A JOSS Program for the Geodetic Inverse Computation*, The Rand Corporation, P-4950, January 1973.

3.2. DISCUSSION

The programs usually written for great circle distances and bearings assume a spherical earth of some mean radius and employ the elementary formulas of spherical trigonometry. The results can be in error by as much as 20 kilometers. To the geodesist (which I am not), such programs are to be anathematized. But beyond the evident demands of surveying, there *are* applications in the military and international domains where much greater precision is required.

In 1828, F. W. Bessel gave the general solution for the geodesic on an ellipsoid of revolution. Two differential equations must be solved as part of Bessel's rigorous procedure.

Unfortunately for rigor, the geodesist's world is neither perfect nor static. Periodically, the semi-major and semi-minor axes of the geoid are changed somewhat in value. We recommend using WGS 72 (World Geodetic System 1972), now rather generally accepted, which assumes

$$a = 6\ 378\ 135.0 \text{ m} \qquad b = 6\ 356\ 750.233 \text{ m},$$

and hence a squared eccentricity of

$$\varepsilon^2 = 1 - (b/a)^2 = 0.006\ 694\ 407 .$$

(The reciprocal of the flattening f is 298.256.)

The *forward* problem in geodesy takes the latitude and longitude of a station, an azimuthal bearing measured clockwise from north, and a distance or geodetic segment, and asks for the geographical coordinates of the terminus of the segment. The *inverse* problem wants the geodetic distance between two stations, given their geographical coordinates, as well as the bearings of each station from the other.

3.3. EQUATIONS

The equations for the solution of these two problems are rather lengthy. To conserve space in this report, they are not reproduced here. They are found in Ref. b, available from The Rand Corporation, Publications Department, Santa Monica, CA 90406.* For those readers who delve into this reference, note that in the *forward* solution programmed here there is a replacement in formula (8) of

$$\frac{1 - k}{1 + k^2/4} \quad \text{by} \quad 1 - k \text{ ,}$$

since $k^2/4$ is less than 1×10^{-6}, and there is a deletion in formula (10) of the term

$$\frac{29}{48} k^3 \cdot \cos 6\ S' \cdot \sin 3\ \Delta S' \text{ ,}$$

which is of the order 1×10^{-9}.

This is done to save scarce program space and to accelerate execution slightly. In comparing the results with the more exact values found by the National Geodetic Survey, the resulting latitude may be in error up to 0.2" (6 meters), and the longitude by considerably less than this.

3.4. PROGRAM NOTES (FORWARD SOLUTION)

(1) There are three places in the program where care must be taken to ensure that the arctangent gives an angle in the correct quadrant. The 'g $\tan^{-1}$' function on the HP-67 produces angles only in quadrants I and IV, but the rectangular-to-polar-coordinate conversion provides the correct quadrant for +/+, +/-, -/-, and -/+. Key

---

*Price to private individuals: $3.00 postpaid.

in the y value, the numerator with *its* sign. Press ENTER. Key in the x value, the denominator with its sign. Press g → P. Press h x ↔ y to get the angle.

(2) Stack manipulation is used in two places to hold values in stack storage until they can be placed in primary storage, avoiding the extra steps involving 'f P ↔ S' if secondary storage were used. It is good practice in such programming to use 'SST' in run mode and get successive traces of the stack contents by 'g STK'. (This is *pre*-bugging rather than *de*-bugging.)

(3) Because of program size (218 steps), the user is asked to make, if necessary, a simple final correction to the displayed longitude (8 and 9 under 3.5.1, User Instructions). It is important to use '360, CHS, h H.MS+' rather than straightforward subtraction in step 8.

Examples. The following comparisons with National Geodetic Survey values use their geoid constants of

$$a = 6\ 378\ 145.00 \text{ m} \qquad b = 6\ 356\ 759.76 \text{ m}$$

$$\varepsilon^2 = 0.006\ 694\ 545\ .$$

Moreover, their *inverse* solutions (geoid distances as segment lengths) are used to check *forward* solutions. (Note: To convert kilometers to nautical miles, divide by 1.852 exactly.)

Let the station of origin be the municipal airport at Fairbanks, Alaska--64°49'08.95", -147°51'51.36". Then we find

Place	Los Angeles	Jakarta	Tel Aviv
Azimuth	135°12'18.42"	281°25'57.06"	357°45'26.32"
Kilometers	3961.6444	-11344.3480	-9259.2287
Latitude	34°02'59.93"	-6°10'00.16"	32°04'59.79"
Δ lat	0.07"	0.16"	0.21"
Longitude	-118°13'59.96"	106°49'59.93"	34°46'0000"
Δ long	0.04"	0.07"	0.00"

If the azimuth (bearing) is greater than 180°, the distance is entered as a minus quantity. The Δ's show the difference with respect to the NGS's more accurate value (0.10" in latitude is about 10 feet).

3.5. PROGRAM NOTES (INVERSE SOLUTION)

(1) In this program also, the 2-parameter arctangent procedure must be followed to get angles in the correct quadrant.

(2) Again because of program space, we neglect terms involving k^3 and except for one case those involving k^2.

(3) On the first R/S if the number appearing (the difference in radians of the longitudes of the second and first stations) is greater than π, then key CHS, hπ, 2, x, +, R/S. Otherwise we get the greater of the two geodesic distances between the stations. Of course, this is readily programmed but we do not have remaining available space.

We now repeat the above examples using the inverse solution.

Place	Los Angeles	Jakarta	Tel Aviv
Latitude	34°03'	-6°10'	32°05'
Longitude	-118°14'	106°50'	34°46'
Kilometers	3961.6406	11344.3418	9259.2221
Δ (meters)	3.8	6.2	6.6
Azimuth	135°12'18.62"	78°34'03.00"	02°14'33.65"
Δ AZ	0.20"	0.06"	0.03"
Back AZ	158°45'02.66"	155°07'39.00"	178°52'19.11"
Δ B/AZ	0.08"	0.01"	0.01"

In this program, azimuths are less than 180° and give the angle from N to the eastbound portion of the geodesic.

The accuracy should be acceptable to all but the most demanding user. We pay for this accuracy in the usual coin of increased computing time. The reader may wish to program using a mean earth radius and the formulas of spherical trigonometry, as mentioned in Sec. 3.2, to persuade himself.

Because of lack of program space, the user must take account of three procedures:

- If both given latitudes are negative (Southern Hemisphere), change signs of *both* latitudes and *both* longitudes.
- If both stations have the same longitude (meridional arc), the program will show an error at step 044 (1/tan 0). In this case add 0.01' to one longitude (0.01) and run from the beginning, although the solution is unstable for such small differences in longitude.
- Registers A, B, C, D are used both for initial and intermediate storage. If a new problem is to be run with the same first station, its coordinates must be restored. You can, however, store the coordinates also in S6 and S7, being careful to precede and follow by f P ↔ S.

3.5.1 USER INSTRUCTIONS

3.1 GEOGRAPHIC COORDINATES OF A GEODETIC SEGMENT FROM GIVEN POINT (FORWARD SOLUTION)

1 2

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	LOAD DATA AND PROGRAM CARDS				
2	KEY LAT OF STATION (D.MS) STO A	64.4909	STO	A	
	(-FOR S. LAT)				
3	KEY LONG OF STATION (D.MS) STO B	-147.5151	STO	B	
	(-FOR W. LONG)				
4	KEY AZIMUTH (D.MS) CLOCKWISE FROM	135.1218	STO	C	
	NORTH. STO C				
5	KEY SEGMENT DISTANCE IN KMS, CHS				
	IF AZ > 180 , STO D (IF INPUT IS IN				
	NAUT. MILES KEY IN, MULTIPLY BY				
	1.852, STO D)	3961.64	STO	D	
6	PRESS A TO GET LAT (D.MS)			A	34.0300
7	PRESS R/S TO GET LONG (D.MS)			R/S	-118.1359
8	IF LONG > 180 , KEY 360,				
	CHS, h H.MS +				
9	IF LONG < -180 KEY 360, h H.MS +				
	<u>DATA CARD</u>				
	ϵ STO 0 0.081 819 356		STO	0	
	$\sqrt{1-\epsilon^2}$ STO 1 0.996 647 176		STO	1	
	b STO 2 (IN KMS) 6 356 .75023		STO	2	
	(WGS 72)				

3.6.1 FORWARD SOLUTION

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	001 *LBLA	21 11	
	002 DSP6	-63 06	
	003 RAD	16-22	RADIAN MODE
	004 Pi	16-24	
	005 RCLC	36 13	
	006 HMS→	16 36	
	007 D→R	16 45	
	008 X>Y?	16-34	
	009 GSB0	23 00	
010	010 STOE	35 15	
	011 RCLA	36 11	
	012 HMS→	16 36	
	013 D→R	16 45	
	014 TAN	43	
	015 RCL1	36 01	
	016 ×	-35	
	017 TAN⁻¹	16 43	
	018 STO4	35 04	
	019 COS	42	
020	020 RCLE	36 15	
	021 SIN	41	
	022 ×	-35	
	023 COS⁻¹	16 42	
	024 STO5	35 05	
	025 RCLE	36 15	
	026 COS	42	
	027 CHS	-22	
	028 ENT↑	-21	
	029 RCL4	36 04	
030	030 TAN	43	POLAR COORDS
	031 →P	34	σ_1 IN CORR. QUAD
	032 X⇄Y	-41	
	033 STO6	35 06	
	034 RCLE	36 15	
	035 COS	42	
	036 CHS	-22	
	037 ENT↑	-21	
	038 RCL4	36 04	
	039 SIN	41	
040	040 RCLE	36 15	
	041 SIN	41	
	042 ×	-35	TO GET λ_1, σ_1
	043 →P	34	IN SAME QUAD
	044 X⇄Y	-41	
	045 STO7	35 07	
	046 RCL0	36 00	
	047 RCL1	36 01	
	048 ÷	-24	
	049 RCL5	36 05	
050	050 SIN	41	
	051 ×	-35	
	052 TAN⁻¹	16 43	
	053 2	02	
	054 ÷	-24	
	055 TAN	43	
	056 X²	53	
	057 STO8	35 08	
	058 RCL6	36 06	
	059 4	04	
060	060 ×	-35	
	061 SIN	41	
	062 8	08	
	063 ÷	-24	
	064 CHS	-22	
	065 RCL8	36 08	
	066 X²	53	
	067 ×	-35	
	068 RCL6	36 06	
	069 2	02	
070	070 ×	-35	
	071 SIN	41	
	072 RCL8	36 08	
	073 ×	-35	
	074 +	-55	
	075 RCL6	36 06	
	076 2	02	
	077 ×	-35	
	078 +	-55	
	079 STO9	35 09	
080	080 1	01	
	081 RCL8	36 08	
	082 -	-45	
	083 RCLD	36 14	
	084 ×	-35	
	085 RCL2	36 02	
	086 ÷	-24	
	087 STO3	35 03	
	088 ST+9	35-55 09	
	089 RCL3	36 03	
090	090 2	02	
	091 ×	-35	
	092 SIN	41	
	093 RCL9	36 09	
	094 2	02	
	095 ×	-35	
	096 COS	42	
	097 ×	-35	
	098 RCL8	36 08	
	099 X²	53	
100	100 ×	-35	
	101 5	05	
	102 ×	-35	
	103 8	08	
	104 ÷	-24	
	105 RCL9	36 09	
	106 COS	42	
	107 RCL3	36 03	
	108 SIN	41	
	109 ×	-35	
110	110 RCL8	36 08	
	111 ×	-35	
	112 -	-45	

REGISTERS

0	1	2	3	4	5	6	7	8	9
ϵ	$\sqrt{1-\epsilon^2}$	b	Δs^1, σ	ψ_1, ψ_2	ψ_m	σ_1, σ_2	λ_1, λ_2	k	$2S_1^1$, $2S^1$, $\Delta\sigma$
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I
ϕ_1	ℓ_1	α_1	$\pm\Delta S$	α_1^*, $\Delta\lambda$	

3.6.1 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	RCL3	36 03	
114	+	-55	
115	STO9	35 09	$\Delta\sigma$
116	RCL6	36 06	
117	+	-55	
118	ENT↑	-21	STACK
119	ENT↑	-21	MANIPULATION
120	RCL6	36 06	
121	+	-55	
122	STO3	35 03	2σ
123	X⇄Y	-41	
124	STO6	35 06	σ_2
125	2	02	
126	ST÷3	35-24 03	σ
127	RCL6	36 06	
128	SIN	41	
129	ENT↑	-21	TO GET λ_2, σ_2
130	RCL6	36 06	IN SAME QUAD
131	COS	42	
132	RCL5	36 05	
133	COS	42	
134	×	-35	
135	→P	34	
136	X⇄Y	-41	
137	ENT↑	-21	
138	RCL7	36 07	
139	-	-45	
140	STOE	35 15	$\Delta\lambda$
141	X⇄Y	-41	
142	STO7	35 07	λ_2
143	RCL6	36 06	
144	COS	42	
145	RCL5	36 05	
146	SIN	41	
147	×	-35	
148	SIN⁻¹	16 41	ψ_2
149	STO4	35 04	
150	TAN	43	
151	RCL1	36 01	
152	÷	-24	
153	TAN⁻¹	16 43	ϕ_2
154	R→D	16 46	
155	→HMS	16 35	
156	R/S	51	DSP LAT
157	RCL9	36 09	
158	2	02	
159	×	-35	
160	SIN	41	
161	RCL3	36 03	
162	4	04	
163	×	-35	
164	COS	42	
165	×	-35	
166	RCL8	36 08	
167	×	-35	
168	4	04	

STEP	KEY ENTRY	KEY CODE	COMMENTS
169	÷	-24	
170	CHS	-22	
171	RCL9	36 09	
172	SIN	41	
173	RCL3	36 03	
174	2	02	
175	×	-35	
176	COS	42	
177	×	-35	
178	+	-55	
179	RCL8	36 08	
180	2	02	
181	÷	-24	
182	1	01	
183	+	-55	
184	RCL9	36 09	
185	×	-35	
186	+	-55	
187	RCL5	36 05	
188	COS	42	
189	×	-35	
190	RCL8	36 08	
191	×	-35	
192	4	04	
193	÷	-24	
194	RCL0	36 00	
195	X²	53	
196	×	-35	
197	1	01	
198	RCL1	36 01	
199	-	-45	
200	RCL5	36 05	
201	COS	42	
202	×	-35	
203	RCL9	36 09	
204	×	-35	
205	-	-45	
206	RCLE	36 15	
207	+	-55	
208	RCLB	36 12	
209	HMS→	16 36	
210	D→R	16 45	
211	+	-55	
212	R→D	16 46	
213	→HMS	16 35	DSP LONG
214	RTN	24	
215	*LBL0	21 00	
216	X⇄Y	-41	
217	-	-45	
218	RTN	24	

LABELS					FLAGS
A	B	C	D	E	0
a	b	c	d	e	1
0	1	2	3	4	2
5	6	7	8	9	3

SET STATUS		
FLAGS (ON OFF)	TRIG	DISP
0 ☐ ☐	DEG ☐	FIX ☐
1 ☐ ☐	GRAD ☐	SCI ☐
2 ☐ ☐	RAD ☐	ENG ☐
3 ☐ ☐		n____

3.5.2 USER INSTRUCTIONS

3.2 GEODETIC DISTANCE BETWEEN TWO STATIONS, AND THEIR BEARINGS (INVERSE SOLUTION)

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	LAT. OF 1ST STATION, D.MS STO A	64.4909	STO	A	
	- IF S. LAT				
2	LONG. OF 1ST STATION, D.M.S. STO B	-147.5151	STO	B	
	- IF W. LONG.				
3	LAT OF 2ND STATION, STO C	34.03	STO	C	
4	LONG. OF 2ND STATION, STO D	-118.1359	STO	D	
5	PRESS A			A	
6	ON R/S, IF > π (3.1416),				
	KEY CHS, hπ, 2, x, +, R/S				
7	ON NEXT R/S, SEE DISTANCE (KMS.)			R/S	3961.64
8	ON NEXT R/S, SEE BEARING STATION			R/S	135.1218
	1 TO STATION 2 (< 180, FROM N TO				
	EASTBOUND PORTION OF GEODESIC) D.MS				
9	ON LAST R/S, SEE BEARING FROM STATION			R/S	158.4502
	2 TO STATION 1 (AS ABOVE) D.MS				
10	FOR S HEMISPHERE SEE NOTES				
	DATA CARD				
	ϵ STO ϕ 0.081 819 356				
	$\sqrt{1-\epsilon^2}$ STO 1 0.996 647 176				
	b STO 2 6 356.750 23				
	(WGS 72)				

3.6.2 INVERSE SOLUTION

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	001 *LBLA	21 11	
	002 RAD	16-22	RADIAN MODE
	003 RCLA	36 11	
	004 HMS→	16 36	
	005 D→R	16 45	
	006 TAN	43	
	007 RCL1	36 01	
	008 ×	-35	
	009 STOA	35 11	$\tan\psi_1$
010	010 RCLB	36 12	
	011 HMS→	16 36	
	012 D→R	16 45	
	013 STOB	35 12	
	014 RCLC	36 13	
	015 HMS→	16 36	
	016 D→R	16 45	
	017 TAN	43	
	018 RCL1	36 01	
	019 ×	-35	
020	020 STOC	35 13	$\tan\psi_2$
	021 RCLD	36 14	
	022 HMS→	16 36	
	023 D→R	16 45	
	024 STOD	35 14	
	025 RCLA	36 11	
	026 RCLC	36 13	
	027 -	-45	
	028 STO3	35 03	
	029 RCLA	36 11	
030	030 RCLC	36 13	
	031 +	-55	
	032 STO4	35 04	
	033 RCLD	36 14	
	034 RCLB	36 12	
	035 -	-45	
	036 R/S	51	
	037 STOE	35 15	$\ell_2-\ell_1$
	038 2	02	
	039 ÷	-24	
040	040 STO5	35 05	$\Delta\lambda^\circ/2$
	041 *LBLB	21 12	
	042 RCL5	36 05	
	043 TAN	43	
	044 1/X	52	
	045 RCL3	36 03	
	046 ×	-35	2 PARAM $\tan^{-1}$
	047 ENT↑	-21	
	048 RCL4	36 04	
	049 →P	34	
050	050 X⇄Y	-41	
	051 ENT↑	-21	
	052 ENT↑	-21	
	053 RCL5	36 05	
	054 -	-45	
	055 STO7	35 07	
	056 X⇄Y	-41	
	057 RCL5	36 05	
	058 +	-55	
	059 STO8	35 08	
060	060 RCLA	36 11	
	061 RCL7	36 07	
	062 COS	42	
	063 ÷	-24	
	064 TAN⁻¹	16 43	
	065 STO9	35 09	
	066 COS	42	
	067 RCL7	36 07	
	068 SIN	41	
	069 ×	-35	
070	070 ENT↑	-21	
	071 RCL7	36 07	
	072 COS	42	
	073 →P	34	
	074 X⇄Y	-41	
	075 P⇄S	16-51	
	076 STO1	35 01	
	077 P⇄S	16-51	
	078 RCL9	36 09	
	079 COS	42	
080	080 RCL8	36 08	
	081 SIN	41	
	082 ×	-35	
	083 ENT↑	-21	
	084 RCL8	36 08	
	085 COS	42	
	086 →P	34	
	087 X⇄Y	-41	
	088 P⇄S	16-51	
	089 STO2	35 02	
090	090 RCL1	36 01	
	091 -	-45	
	092 STO3	35 03	
	093 RCL1	36 01	
	094 RCL2	36 02	
	095 +	-55	
	096 2	02	
	097 ÷	-24	
	098 STO4	35 04	
	099 P⇄S	16-51	
100	100 RCL9	36 09	
	101 SIN	41	
	102 RCL0	36 00	
	103 ×	-35	
	104 RCL1	36 01	
	105 ÷	-24	
	106 TAN⁻¹	16 43	
	107 2	02	
	108 ÷	-24	
	109 TAN	43	
110	110 X²	53	
	111 P⇄S	16-51	
	112 STO5	35 05	

REGISTERS

0	1	2	3	4	5	6	7	8	9
ϵ	$\sqrt{1-\epsilon^2}$	b	$\tan\psi_1 - \tan\psi_2$	$\tan\psi_1$ $\tan\psi_2$	$\Delta\lambda^i/2$	$\Delta\lambda^{i+1/2}$	$\lambda_1{}^i$	$\lambda_2{}^i$	$\psi_m{}^i$
S0	S1 $\sigma_2{}^i$	S2 $\sigma_2{}^i$	S3 $\Delta\sigma^i$	S4 σ^i	S5 k^i	S6	S7	S8	S9

A	B	C	D	E	I
$\phi_1/\tan\psi_1$	ℓ_1	$\phi_2/\tan\psi_2$	ℓ_2	$\Delta\ell=\ell_2-\ell_1$	

3.6.2 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	CHS	-22	
114	2	02	
115	÷	-24	
116	RCL4	36 04	
117	2	02	
118	x	-35	
119	COS	42	
120	x	-35	
121	RCL3	36 03	
122	SIN	41	
123	x	-35	
124	2	02	
125	P⇄S	16-51	
126	RCL1	36 01	
127	1	01	
128	+	-55	
129	÷	-24	
130	P⇄S	16-51	
131	RCL5	36 05	
132	2	02	
133	÷	-24	
134	-	-45	
135	RCL3	36 03	
136	x	-35	
137	+	-55	
138	P⇄S	16-51	
139	RCL9	36 09	
140	COS	42	
141	x	-35	
142	2	02	
143	÷	-24	
144	RCL8	36 08	
145	X²	53	
146	x	-35	
147	RCLE	36 15	
148	+	-55	
149	2	02	
150	÷	-24	
151	STO6	35 06	
152	RCL5	36 05	
153	-	-45	
154	DSP6	-63 06	
155	RND	16 24	
156	X≠Y?	16-42	LOOP
157	GTOC	22 13	
158	P⇄S	16-51	
159	RCL3	36 03	
160	2	02	
161	x	-35	
162	SIN	41	
163	RCL4	36 04	
164	4	04	
165	x	-35	
166	COS	42	
167	x	-35	
168	8	08	

STEP	KEY ENTRY	KEY CODE	COMMENTS
169	÷	-24	
170	CHS	-22	
171	RCL5	36 05	
172	X²	53	
173	x	-35	
174	RCL3	36 03	
175	SIN	41	
176	RCL4	36 04	
177	2	02	
178	x	-35	
179	COS	42	
180	x	-35	
181	RCL5	36 05	
182	x	-35	
183	+	-55	
184	RCL3	36 03	
185	+	-55	
186	1	01	
187	RCL5	36 05	
188	−	-45	
189	÷	-24	
190	P⇄S	16-51	
191	RCL2	36 02	
192	x	-35	
193	R/S	51	DISTANCE
194	1	01	
195	ENT↑	-21	
196	RCL7	36 07	
197	TAN	43	
198	RCLA	36 11	
199	GSBD	23 14	
200	R/S	51	AZIMUTH
201	1	01	
202	ENT↑	-21	
203	RCL8	36 08	
204	TAN	43	
205	RCLC	36 13	
206	GSBD	23 14	
207	RTN	24	BACK AZIMUTH
208	*LBLD	21 14	
209	TAN⁻¹	16 43	
210	SIN	41	
211	x	-35	
212	CHS	-22	
213	→P	34	
214	X⇄Y	-41	
215	R→D	16 46	
216	→HMS	16 35	
217	RTN	24	
218	*LBLC	21 13	
219	RCL6	36 06	
220	STO5	35 05	
221	GTOB	22 12	

LABELS					FLAGS	SET STATUS		
A	B	C	D	E	0	FLAGS	TRIG	DISP
a	b	c	d	e	1	ON OFF		
0	1	2	3	4	2	0 ☐ ☐ 1 ☐ ☐	DEG ☐ GRAD ☐	FIX ☐ SCI ☐
5	6	7	8	9	3	2 ☐ ☐ 3 ☐ ☐	RAD ☐	ENG ☐ n ____

4. REENTRY TRAJECTORIES

4.1. REFERENCES

a. M. M. Moe, "An Approximation to the Re-Entry Trajectory," *ARS Journal*, January 1960, pp. 50-53.

b. R. Blum, "Re-Entry Trajectories: Flat Earth Approximation," *ARS Journal*, April 1962, pp. 616-620.

4.2. DISCUSSION

For a body with zero lift and given weight-to-drag ratio (beta), entering the upper atmosphere at time 0 with assigned altitude, velocity, and path angle, find at desired time intervals the subsequent range, altitude, velocity, and path angle to impact.

An "exact" solution (an analytic solution is not possible) requires the numerical integration of two second-order differential equations, both highly nonlinear. The technique is to merge a program for the functions of the differential equations with a slightly retailored version of Program 20, which solves fourth-order differential equations, also providing on a separate card an initialization program to determine values for the variables at the first time interval, $t = h$. This merging, retailoring, and initialization is a useful model for similar applications in other work, when a particular set of equations is to be solved frequently.

The equations adopted are those for a nonrotating round earth. There are three assumptions:

1. In the weight-to-drag ratio $\beta = mg/C_D A$, the drag coefficient is held constant, although it actually varies with Mach number;
2. Surface gravity g is used uncorrected for altitude by $(r_e/r)^2$, where r_e is the radius of the earth and r is the distance to the body from the earth's center;
3. The density of the atmosphere is approximated by $\rho(y) = 0.00237 \exp\{-y/24{,}000\}$ slug/ft^3, where $y = r - r_e$ is the altitude in feet (Ref. a).

We have at Rand an on-line, time-sharing program in JOSS, written by D. C. Kephart, which performs the numerical integration using the fourth-order Runge-Kutta formulas and a time interval of 1 sec. Hence a convenient base of comparison with the simpler and coarser predictor-corrector approach of Program 20 is available.

4.3. EQUATIONS

The basic variables are:

- r = distance from earth's center (ft)
- ϕ = polar angle from initial vector to current vector
- V = speed of the body (ft/sec)
- θ = angle measured positively downward from the local horizontal to the velocity vector.

In these variables, the equations of motion are

$$\frac{1}{r}\frac{d}{dt}\left(r^2\frac{d\phi}{dt}\right) = -\frac{F_D}{m}\cos\theta \tag{1}$$

$$\frac{d^2r}{dt^2} = r\left(\frac{d\phi}{dt}\right)^2 - g + \frac{F_D}{m}\sin\theta\ , \tag{2}$$

where the drag force due to air is $F_D = \rho(y)V^2C_DA/2$. We also have

$$\begin{gathered} r(d\phi/dt) = V\cos\theta\ ,\quad dr/dt = -V\sin\theta \\ V^2 = (dr/dt)^2 + (r\ d\phi/dt)^2\ . \end{gathered} \tag{3}$$

Using the variables

$$x = r_e\phi\ \text{(range)} \qquad y = r - r_e\ \text{(altitude)}$$

$$u = x' \qquad v = y'\ ,$$

the system of four first-order differential equations is

$$x' = u \qquad y' = v$$

$$u' = -\left[\frac{2v}{y + r_e} + \frac{g\rho(y)V}{2\beta}\right] u \tag{4}$$

$$v' = \frac{y + r_e}{r_e^2} u^2 - g - \frac{g\rho(y)V}{2\beta} v ,$$

where

$$V^2 = (1 + y/r_e)^2 u^2 + v^2 .$$

As a matter of interest, it will be found by computing dV_0/dt and $d\theta_0/dt$ that for some reentry altitudes V can *increase* initially and θ can *decrease* (pitch-up). An example will illustrate these phenomena.

Turn now to the determination of x_1, y_1, u_1, v_1 at $t = h$, the first time increment, which are values needed for the method of Program 20. This will be programmed to initialize the numerical integration. We have

$$x_1 = u_0 h + u_0' h^2/2 \qquad y_1 = y_0 + v_0 h + v_0' h^2/2$$
$$u_1 = u_0 + u_0' h \qquad v_1 = v_0 + v_0' h , \tag{5}$$

where h is the interval of integration (10 sec seems suitable).

The equations for the flat-earth approximation are found from (4) by putting $r_e = \infty$. The interested reader may wish to try programming this case along the lines of this section to compare the answers with those for a round earth.

4.4. PROGRAM NOTES

Program 20 is modified as follows:

1. At the beginning of LBL A, 4 is stored in registers D and E, replacing V_0 and θ_0 used during the initialization.
2. Most of LBL 4 is deleted, since we are working with 4 equations and we want to compute velocity and path angle for display.
3. The subroutine programming of u' and v' is straightforward, but we have had to use program steps for g/2 and g because storage registers are not available.

Example. Reentry conditions at t = 0 are

$$y_0 = 250{,}000 \text{ ft} \quad V_0 = 30{,}000 \text{ ft/sec} \quad \beta = 1000 \quad \theta_0 = 5° \ .$$

The tabulation below shows values at intervals of 30 sec, although the step size used was 10. The numbers in parentheses are the outputs of Kephart's more refined integration procedure. His density function was used,

$$0.0027 \exp \{-y/23{,}500\} \ ,$$

instead of

$$0.00237 \exp \{-y/24{,}000\} \ ,$$

whose constants are stored on our data card (in S8 and S9).

After the "knee" of the trajectory the altitudes are in error by about 1000 ft low. The reason is that 10 sec is too great an interval in this region. If h = 2 sec is used, the maximum error in altitude reduces to about 80 ft and the other values are sensibly exact. And if h = 1 sec, the value used in Kephart's calculation, the maximum difference in altitude is 3.5 ft.

This is entirely of theoretical interest as a comparison of two methods of numerical integration: the fourth-order Runge-Kutta and the extended Hamming predictor-corrector method employed here. In the practical sense, the variation of the drag coefficient with Mach

TIME (sec)	x RANGE (n mi)	y ALTITUDE (ft)	ν VELOCITY (ft/sec)	θ ANGLE (deg)
20	97.33	199 751	29 993	4.60
	(97.37)	(199 753)	(29 987)	(4.58)
50	242.03	132 807	28 692	4.05
	(242.07)	(133 228)	(28 730)	(4.00)
80	363.79	80 636	19 009	4.27
	(364.40)	(81 490)	(19 176)	(4.19)
110	422.90	46 342	6 657	8.26
	(424.67)	(47 349)	(6 786)	(8.06)
140	441.84	19 889	2 109	23.19
	(444.03)	(20 860)	(2 218)	(22.36)
160	446.25	3 474	1 154	43.38
	(448.64)	(4 328)	(1 207)	(41.82)
170	447.37	-4 333	921	55.10
	(--)	(--)	(--)	(--)

number, the departure of the atmosphere from the ideal one used in the calculation and, above all, the probably asymmetric ablation of the nose cone can vitiate such accuracy. Taking h = 10 sec is a reasonable choice that minimizes computing time while yielding acceptable accuracy for the purposes to which this program should be put.

4.5 USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	LOAD DATA CARD (BOTH SIDES)				0.00
2	TIME INTERVAL h, STO D, STO 1	10	STO	0	
			STO	1	
3	β STO B, V STO D, θ_0 STO E	1000	STO	B	1000.00
	(θ IS POSITIVE DOWNWARDS)	30000	STO	D	30000.00
		5	STO	E	5.00
4	y_0 STO 6, STO 7	250000	STO	6	250000
			STO	7	250000
5	LOAD INITIALIZATION CARD				250000
6	PRESS E y_1			E	224358.95
7	LOAD PROGRAM CARD				
8	PRESS A			A	
9	SEE t (h PAUSE)				20.00
	SEE RANGE n.mi. (f-x-)				57.32
	SEE ALT ft (f-x-)				199752.44
	SEE VEL ft/SEC (f-x-)				29989.44
	SEE ANGLE (f-x-)				4.60
10	IF MORE TIME NEEDED TO RECORD, USE R/S				30.00
	(NOTE: VALUES DO NOT AGREE				146.00
	WITH EXAMPLE IN TEXT BECAUSE				176225.76
	A DIFFERENT DENSITY FUNCTION				29870.78
	IS USED HERE.)				4.41

4.6 RE- ENTRY TRAJECTORIES (INITIALIZATION)

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLB	21 12	(IN PRI.)	057	RCLA	36 11	(LBL9 NOT NEEDED
002	GSB9	23 09	V_o(COULD BE RCLD)	058	÷	-24	FOR INITIALIZATION
003	1	01		059	1	01	--SEE STEP 002.
004	6	06	g/2	060	+	-55	BUT WILL BE NEEDED
005	.	-62		061	P⇄S	16-51	FOR RE-ENTRY PRGM.)
006	1	01		062	RCL1	36 01	
007	×	-35		063	×	-35	
008	RCLB	36 12	β	064	X²	53	
009	÷	-24		065	RCL5	36 05	
010	RCL7	36 07		066	X²	53	
011	P⇄S	16-51	(SEC)	067	+	-55	
012	RCL9	36 09		068	√X	54	
013	÷	-24		069	P⇄S	16-51	
014	eˣ	33		070	RTN	24	V_o
015	×	-35		071	*LBLE	21 15	
016	RCL8	36 08		072	RCLE	36 15	
017	×	-35		073	COS	42	
018	STOC	35 13	$g\rho(y)\,V_o/2\beta$	074	RCLD	36 14	
019	RCL5	36 05		075	×	-35	$V_o \cos\theta_o$
020	P⇄S	16-51	(PRI)	076	RCL6	36 06	
021	2	02		077	RCLA	36 11	
022	×	-35		078	÷	-24	
023	RCL7	36 07		079	1	01	
024	RCLA	36 11		080	+	-55	$1 + y_o/r_e$
025	+	-55		081	÷	-24	
026	÷	-24		082	P⇄S	16-51	(SEC)
027	+	-55		083	STO0	35 00	
028	CHS	-22		084	STO1	35 01	$V_o \cos\theta/(1+y_o/r_e)$
029	P⇄S	16-51		085	RCLE	36 15	
030	RCL1	36 01	u_o	086	SIN	41	
031	×	-35		087	RCLD	36 14	
032	P⇄S	16-51		088	×	-35	
033	RTN	24	u_o^1	089	CHS	-22	
034	*LBLa	21 16 11	(IN PRI.)	090	STO4	35 04	$-V_o \sin\theta$
035	RCL7	36 07		091	STO5	35 05	
036	RCLA	36 11		092	P⇄S	16-51	(PRI)
037	+	-55	$y_o + r_e$	093	GSBB	23 12	
038	P⇄S	16-51	(SEC)	094	P⇄S	16-51	(SEC)
039	RCL1	36 01		095	STO2	35 02	u_o^1
040	RCLA	36 11		096	P⇄S	16-51	(PRI)
041	÷	-24		097	GSBa	23 16 11	
042	X²	53		098	P⇄S	16-51	
043	×	-35		099	STO6	35 06	v_o^1
044	3	03		100	P⇄S	16-51	
045	2	02	g	101	RCL0	36 00	
046	.	-62		102	×	-35	
047	2	02		103	P⇄S	16-51	(SEC)
048	-	-45		104	RCL4	36 04	
049	RCLC	36 13		105	+	-55	
050	RCL5	36 05		106	STO5	35 05	v_1
051	×	-35		107	RCL0	36 00	
052	-	-45		108	RCL2	36 02	
053	P⇄S	16-51		109	P⇄S	16-51	(PRI)
054	RTN	24	v_o^1	110	RCL0	36 00	
055	*LBL9	21 09		111	×	-35	
056	RCL7	36 07	y_o	112	+	-55	

REGISTERS

0	1	2	3	4	5	6	7	8	9
h	h					y_o	y_o		

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
u_o	u_o, u_1	u_o^1	X_1	v_o	v_o, v_1	v_o	Y_1	.00237	-24000

A	B	C	D	E	I
20903040	β	(018)	V_o	θ_o	

4.6 PROGRAM LISTING

STEP	KEY ENTRY		KEY CODE	COMMENTS
	113	P⇄S	16-51	(SEC)
	114	STO1	35 01	u_1
	115	RCL0	36 00	
	116	RCL2	36 02	
	117	P⇄S	16-51	(PRI)
	118	RCL0	36 00	
	119	×	-35	
120	120	2	02	
	121	÷	-24	
	122	+	-55	
	123	RCL0	36 00	
	124	×	-35	
	125	STO3	35 03	x_1
	126	P⇄S	16-51	(SEC)
	127	RCL4	36 04	
	128	RCL6	36 06	
	129	P⇄S	16-51	(PRI)
130	130	RCL0	36 00	
	131	×	-35	
	132	2	02	
	133	÷	-24	
	134	+	-55	
	135	RCL0	36 00	
	136	×	-35	
	137	RCL6	36 06	
	138	+	-55	
	139	STO7	35 07	y_1
140	140	RTN	24	
	141	R/S	51	
150				
160				
180				
190				
200				

DATA CARD

SECONDARY STORAGE

Value	Register	Comment
0.00	0	
0.00	1	
0.00	2	
0.00	3	
0.00	4	
0.00	5	
0.00	6	
0.00	7	
2.370000000-03	8	DENSITY CONSTANT
-24000.00	9	ALT. CONSTANT
20903040.00	A	EARTH RADIUS (FT)
0.00	B	
0.00	C	
0.00	D	
0.00	E	
0.00	I	

LABELS					FLAGS
A	B USED	C	D	E INITIALIZE	0
a USED	b	c	d	e	1
0	1	2	3	4	2
5	6	7	8	9 USED	3

SET STATUS

FLAGS	ON	OFF	TRIG		DISP	
0	☐	☐	DEG	☐	FIX	☐
1	☐	☐	GRAD	☐	SCI	☐
2	☐	☐	RAD	☐	ENG	☐
3	☐	☐			n ____	

4.6 RE-ENTRY TRAJECTORY PROGRAM (ROUND EARTH)

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	(PRGM NOTE 1)
002	DSP2	-63 02	
003	4	04	
004	STOD	35 14	
005	STOE	35 15	
006	3	03	
007	STOI	35 46	
008	GTOC	22 13	
009	*LBLC	21 13	(SEE PRGM 20)
010	GSBi	23 45	
011	ISZI	16 26 46	
012	STOi	35 45	
013	RCL0	36 00	
014	x	-35	
015	2	02	
016	x	-35	
017	DSZI	16 25 46	
018	DSZI	16 25 46	
019	RCLi	36 45	
020	+	-55	
021	ISZI	16 26 46	
022	RCLi	36 45	
023	DSZI	16 25 46	
024	STOi	35 45	
025	R↓	-31	
026	ISZI	16 26 46	
027	ISZI	16 26 46	
028	ISZI	16 26 46	
029	STOi	35 45	
030	RCLD	36 14	
031	1	01	
032	−	-45	
033	STOD	35 14	
034	X=0?	16-43	
035	GTO0	22 00	
036	ISZI	16 26 46	
037	ISZI	16 26 46	
038	GTOC	22 13	
039	*LBL0	21 00	(SEE PRGM 20)
040	RCLE	36 15	
041	STOD	35 14	
042	RCL0	36 00	
043	ST+1	35-55 01	
044	GSB2	23 02	
045	3	03	
046	STOI	35 46	
047	GTO1	22 01	
048	*LBL1	21 01	(SEE PRGM 20)
049	GSBi	23 45	
050	ISZI	16 26 46	
051	RCLi	36 45	
052	+	-55	
053	RCL0	36 00	
054	x	-35	
055	2	02	
056	x	-35	
057	DSZI	16 25 46	
058	RCLi	36 45	
059	+	-55	
060	DSZI	16 25 46	
061	RCLi	36 45	
062	4	04	
063	x	-35	
064	+	-55	
065	5	05	
066	÷	-24	
067	ISZI	16 26 46	
068	ISZI	16 26 46	
069	ISZI	16 26 46	
070	STOi	35 45	
071	RCLD	36 14	
072	1	01	
073	−	-45	
074	STOD	35 14	
075	X=0?	16-43	
076	GTO4	22 04	
077	ISZI	16 26 46	
078	ISZI	16 26 46	
079	GTO1	22 01	
080	*LBL4	21 04	(PRGM 20 STEPS)
081	RCLE	36 15	077
082	STOD	35 14	
083	GSB2	23 02	
084	RCL1	36 01	080
085	PSE	16 51	t
086	RCL3	36 03	082
087	6	06	
088	0	00	(TO GET N.MI.)
089	8	08	
090	0	00	
091	÷	-24	
092	PRTX	-14	083 x
093	RCL7	36 07	089
094	PRTX	-14	y
095	GSB9	23 09	
096	STOD	35 14	V
097	PRTX	-14	
098	P⇄S	16-51	
099	RCL5	36 05	v
100	P⇄S	16-51	
101	RCLD	36 14	
102	÷	-24	
103	CHS	-22	
104	SIN⁻¹	16 41	θ
105	PRTX	-14	
106	GTOA	22 11	
107	*LBL3	21 03	
108	P⇄S	16-51	(SEC)
109	RCL1	36 01	
110	P⇄S	16-51	(PRI)
111	RTN	24	u
112	*LBL7	21 07	

REGISTERS

0	1	2	3	4	5	6	7	8	9
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I

4.6 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	P⇄S	16-51	
114	RCL5	36 05	
115	P⇄S	16-51	
116	RTN	24	V
117	*LBLB	21 12	(PRGM FOR u_o^1
118	GSB9	23 09	SEE PREVIOUS
119	1	01	LISTING.)
120	6	06	
121	.	-62	
122	1	01	
123	x	-35	
124	RCLB	36 12	
125	÷	-24	
126	RCL7	36 07	
127	P⇄S	16-51	
128	RCL9	36 09	
129	÷	-24	
130	e^x	33	
131	x	-35	
132	RCL8	36 08	
133	x	-35	
134	STOC	35 13	
135	RCL5	36 05	
136	P⇄S	16-51	
137	2	02	
138	x	-35	
139	RCL7	36 07	
140	RCLA	36 11	
141	+	-55	
142	÷	-24	
143	+	-55	
144	CHS	-22	
145	P⇄S	16-51	
146	RCL1	36 01	
147	x	-35	
148	P⇄S	16-51	
149	RTN	24	
150	*LBLa	21 16 11	(PRGM FOR v^1)
151	RCL7	36 07	
152	RCLA	36 11	
153	+	-55	
154	P⇄S	16-51	
155	RCL1	36 01	
156	RCLA	36 11	
157	÷	-24	
158	X^2	53	
159	x	-35	
160	3	03	
161	2	02	
162	.	-62	
163	2	02	
164	−	-45	
165	RCLC	36 13	
166	RCL5	36 05	
167	x	-35	
168	−	-45	
169	P⇄S	16-51	
170	RTN	24	
171	*LBL9	21 09	(PRGM FOR V)
172	RCL7	36 07	
173	RCLA	36 11	
174	÷	-24	
175	1	01	
176	+	-55	
177	P⇄S	16-51	
178	RCL1	36 01	
179	x	-35	
180	X^2	53	
181	RCL5	36 05	
182	X^2	53	
183	+	-55	
184	√X	54	
185	P⇄S	16-51	
186	RTN	24	
187	*LBL2	21 02	
188	P⇄S	16-51	
189	RCL7	36 07	
190	STO5	35 05	
191	RCL3	36 03	
192	STO1	35 01	
193	P⇄S	16-51	
194	RCL9	36 09	
195	STO7	35 07	
196	RCL5	36 05	
197	STO3	35 03	
198	RTN	24	
199	R/S	51	

LABELS					FLAGS
A START	B	C	D	E	0
a	b	c	d	e	1
0	1	2	3	4	2
5	6	7	8	9	3

SET STATUS

FLAGS (ON OFF)	TRIG	DISP
0 ☐ ☐	DEG ☐	FIX ☐
1 ☐ ☐	GRAD ☐	SCI ☐
2 ☐ ☐	RAD ☐	ENG ☐
3 ☐ ☐		n____

5. SATELLITE ORBITAL ELEMENTS

5.1. REFERENCES

a. R. H. Frick, W. I. Rumer, and E. H. Sharkey, *Trajectory and Orbit Plotter Instruction Manual*, The Rand Corporation, R-418-PR, October 1963.

b. *Space Planners Guide*, USAF Air Force Systems Command, 1 July 1965 (For Official Use Only).

5.2. DISCUSSION

The orbital elements are

1.	Injection altitude	(n mi)	H_I
2.	Injection velocity	(ft/sec)	V_I
3.	Injection flight path angle	(deg)	γ_I
4.	Period	(min)	T
5.	Eccentricity	(dimensionless)	ε
6.	Semi-major axis	(n mi)	A
7.	Perigee altitude	(n mi)	H_p
8.	Apogee altitude	(n mi)	H_a
9.	True anomaly (at injection measured from apogee)	(deg)	ν_I

The program below solves two of many possible problems:

A. Given 1, 2, 3 find the remaining elements.

B. Given 1, 7, 8 find the remaining elements.

Reference b uses a sequence of nomograms to solve these problems. Some of these nomograms are difficult to read with any precision because of close interval spacing for relatively large increments.

5.3. EQUATIONS

In Fig. 5.1 distances are measured in units of earth's radius (3437.9 n mi) and velocity in units of $\sqrt{R_E g_0}$ (25,943 ft/sec), the

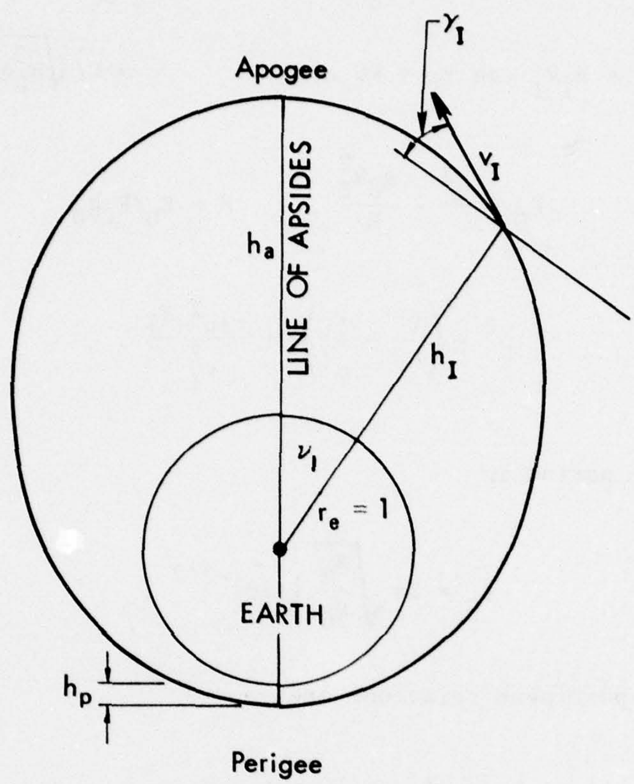


Fig. 5.1 — In-plane orbital elements

surface orbital velocity, and g_0, the surface gravitational acceleration (32.174 ft/sec^2).

$$r = R/R_E \ , \qquad v = V/\sqrt{R_E g_0} \tag{1}$$

$$C = R_I V_I \cos \gamma_I = RV \cos \gamma \ , \qquad c = C/\sqrt{R_E^3 g_0} \tag{2}$$

$$E_0 = \frac{V^2}{2} - \frac{g_0 R_E^2}{R} \ , \qquad E = E_0/R_E g_0 \tag{3}$$

$$\rho^2 = \left(\frac{1}{r_I} - \frac{1}{c^2}\right)^2 + \frac{\tan^2 \gamma_I}{r_I^2} \ . \tag{4}$$

The orbital period is

$$T_0 = 2\pi \sqrt{\frac{R_E}{g_0}} \, |-2E|^{-3/2} \ . \tag{5}$$

Other pertinent relations are

$$\frac{1}{r} = \frac{1}{c^2} - \rho \cos(\theta - \nu_I) \tag{6}$$

$$\tan \gamma = -r \rho \sin(\theta - \nu_I) \tag{7}$$

$$\frac{1}{r_a} = \frac{1}{c^2} - \rho \ , \qquad \frac{1}{r_p} = \frac{1}{c^2} + \rho \tag{8a,8b}$$

where r_a, r_p are the apogee and perigee distances,

$$\varepsilon = \frac{r_a - r_p}{r_a + r_p} \text{ (eccentricity)} \tag{9}$$

$$a = \frac{r_a + r_p}{2} \text{ (semi-major axis) } . \tag{10}$$

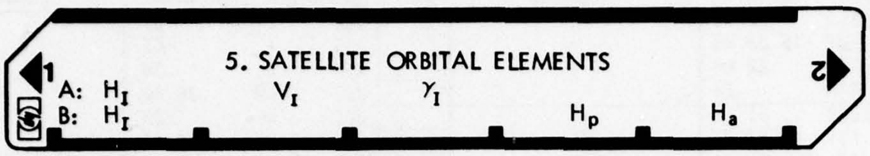
5.4. PROGRAM NOTES

Flag F3 (set by digit entry, cleared by test) is used to direct the program to Problems A or B. The program flow for Problem A (equation numbers in parentheses) is:

1. H_I, LBL A, GTO LBL 0 (1), r_I
2. V_I, LBL B, GTO LBL 1 (1), v_I
3. γ_I, LBL C, GTO LBL 2, γ_I
4. LBL e (2), $1/C^2$, (3), $|E|$ (total energy/non-dim.)
5. LBL D (8b), Hp
6. LBL E (8a), H_a
7. LBL a (5) T_0
8. LBL b (9), ε
9. LBL c (10), A
10. LBL d (7), ν_I

This program can be appreciably shortened by a judicious use of subroutines. A suggested exercise is to rewrite it to see how many problems other than A and B can be packed on one card.

5.5 USER INSTRUCTIONS



STEP	PROBLEM A INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	KEY H_I (n. mi.), PRESS A, SEE r_I	100		1.029
2	KEY V_I (f.p.s.), PRESS B, SEE v_I	26000		1.003
3	KEY γ_I (degs), PRESS C, SEE γ_I	2		2.000
4	PRESS f e			.046
5	PRESS D, SEE H_p (n.mi.)			46.534
6	PRESS E, SEE H_a (n.mi.)			405.442
7	PRESS f a, SEE T_O (mins)			92.866
8	PRESS f b, SEE ϵ			.049
9	PRESS f c, SEE A (n.mi.)			3663.9
10	PRESS f d, SEE ν_I (degs)			47.442
	PROBLEM B			
1	KEY H_I, PRESS A, SEE r_I	150		1.044
2	KEY H_p, PRESS D, SEE r_p	100		1.029
3	KEY H_a, PRESS E, SEE r_a	600		1.175
4	PRESS C, SEE γ_I			2.273
5	PRESS B, SEE V_I			26047
6	PRESS R/S			.454
7	PRESS f a, SEE T_O			97.62
8	PRESS f b, SEE ϵ			.066
9	PRESS f c, SEE A			3787.9
10	PRESS f d, SEE ν_I			39.203
	DATA CARD 0.000000000 0			
	0.000000000 1			
	0.000000000 2			
	0.000000000 3			
	0.000000000 4			
	0.000000000 5			
	0.000000000 6			
	3437.900000 7			
	25933.32000 8			
	29.84260000 9			
	0.000000000 A			
	0.000000000 B			
	0.000000000 C			
	0.000000000 D			
	0.000000000 E			
	0.000000000 I			

PRECEDING PAGE BLANK

5.6 SATELLITE ORBITAL ELEMENTS

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
001	001 *LBLA	21 11			057 X^2	53	
	002 F3?	16 23 03			058 +	-55	
	003 GTO0	22 00			059 √X	54	
	004 RTN	24		060	060 RCL0	36 00	
	005 *LBLB	21 12			061 ×	-35	
	006 F3?	16 23 03			062 TAN-1	16 43	
	007 GTO1	22 01			063 STO2	35 02	γ_I
	008 RCLC	36 13			064 RTN	24	
	009 √X	54			065 *LBLD	21 14	
010	010 1/X	52	c		066 F3?	16 23 03	
	011 RCL0	36 00			067 GTO3	22 03	
	012 ÷	-24	c/r_I		068 RCLC	36 13	
	013 RCL2	36 02			069 RCLA	36 11	
	014 COS	42		070	070 +	-55	
	015 ÷	-24	$c/r_I \cos \gamma_I$		071 1/X	52	
	016 RCL8	36 08			072 STO3	35 03	
	017 ×	-35			073 1	01	
	018 STO1	35 01	V_I		074 -	-45	
	019 R/S	51			075 RCL7	36 07	H_p
020	020 RCL0	36 00			076 ×	-35	
	021 1/X	52	$1/r_I$		077 RTN	24	
	022 RCL1	36 01			078 *LBLE	21 15	
	023 RCL8	36 08			079 F3?	16 23 03	
	024 ÷	-24		080	080 GTO4	22 04	
	025 X^2	53			081 RCLC	36 13	
	026 2	02			082 RCLA	36 11	
	027 ÷	-24			083 -	-45	
	028 -	-45			084 1/X	52	
	029 STOE	35 15	|E|		085 STO4	35 04	
030	030 RTN	24			086 1	01	
	031 *LBLC	21 13			087 -	-45	
	032 F3?	16 23 03			088 RCL7	36 07	
	033 GTO2	22 02			089 ×	-35	H_a
	034 RCL3	36 03	r_p	090	090 RTN	24	
	035 1/X	52			091 *LBL0	21 00	
	036 RCL4	36 04	r_a		092 RCL7	36 07	
	037 1/X	52			093 ÷	-24	
	038 +	-55			094 1	01	
	039 2	02			095 +	-55	
040	040 ÷	-24			096 STO0	35 00	r_I
	041 STOC	35 13	$1/c^2$		097 RTN	24	
	042 RCL3	36 03			098 *LBL1	21 01	
	043 1/X	52			099 RCL8	36 08	
	044 RCL4	36 04		100	100 ÷	-24	
	045 1/X	52			101 STO1	35 01	v_I
	046 -	-45			102 RTN	24	
	047 2	02			103 *LBL2	21 02	
	048 ÷	-24			104 STO2	35 02	γ_I
	049 STOA	35 11	ρ		105 RTN	24	
050	050 RCL0	36 00			106 *LBL3	21 03	
	051 1/X	52			107 RCL7	36 07	
	052 RCLC	36 13			108 ÷	-24	
	053 -	-45			109 1	01	
	054 X^2	53		110	110 +	-55	
	055 CHS	-22			111 STO3	35 03	r_p
	056 RCLA	36 11			112 RTN	24	

REGISTERS

0	1	2	3	4	5	6	7	8	9
r_I	v_I	γ_I	r_p	r_a			R_E	$\sqrt{R_E g_o}$	$\frac{\pi}{60}\sqrt{\frac{R_E}{2 g_o}}$
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I
P		$1/c^2$		|E|	

5.6 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
	113 *LBL4	21 04	
	114 RCL7	36 07	
	115 ÷	-24	
	116 1	01	
	117 +	-55	
	118 STO4	35 04	r_a
	119 RTN	24	
	120 *LBLa	21 16 11	
	121 RCLE	36 15	
	122 1	01	
	123 .	-62	
	124 5	05	
	125 Y^X	31	
	126 1/X	52	
	127 RCL9	36 09	T_O
	128 ×	-35	
	129 RTN	24	
	130 *LBLb	21 16 12	
	131 RCL4	36 04	
	132 RCL3	36 03	
	133 -	-45	
	134 RCL4	36 04	
	135 RCL3	36 03	
	136 +	-55	
	137 ÷	-24	ϵ
	138 RTN	24	
	139 *LBLc	21 16 13	
	140 RCL3	36 03	
	141 RCL4	36 04	
	142 +	-55	
	143 2	02	
	144 ÷	-24	
	145 RCL7	36 07	
	146 ×	-35	A
	147 RTN	24	
	148 *LBLd	21 16 14	
	149 RCL2	36 02	
	150 TAN	43	
	151 RCL0	36 00	
	152 ÷	-24	
	153 RCLA	36 11	
	154 ÷	-24	ν_I
	155 SIN^{-1}	16 41	
	156 RTN	24	
	157 *LBLe	21 16 15	
	158 RCL0	36 00	
	159 RCL1	36 01	
	160 ×	-35	
	161 RCL2	36 02	
	162 COS	42	
	163 ×	-35	
	164 X^2	53	
	165 1/X	52	
	166 STOC	35 13	$1/c^2$
	167 RCL0	36 00	
	168 1/X	52	
	169 RCL1	36 01	
	170 X^2	53	
	171 2	02	
	172 ÷	-24	
	173 -	-45	
	174 STOE	35 15	$\|E\|$
	175 RCL0	36 00	
	176 1/X	52	
	177 RCLC	36 13	
	178 -	-45	
	179 X^2	53	
	180 RCL2	36 02	
	181 TAN	43	
	182 RCL0	36 00	
	183 ÷	-24	
	184 X^2	53	
	185 +	-55	
	186 $\sqrt{X}$	54	
	187 STOA	35 11	ρ
	188 RTN	24	

LABELS

A H_I	B V_I	C γ_I	D H_p	E H_a
a T_O	b ϵ	c A	d ν_I	e $1/c^2$, $\|E\|$
0 r_I	1 v_I	2 γ_I	3 r_p	4 r_a
5	6	7	8	9

FLAGS: 0, 1, 2, 3

SET STATUS

FLAGS (ON OFF)	TRIG	DISP
0 ☐ ☐	DEG ☐	FIX ☐
1 ☐ ☐	GRAD ☐	SCI ☐
2 ☐ ☐	RAD ☐	ENG ☐
3 ☐ ☐		n____

6. SATELLITE TRACKING

6.1. REFERENCE

a. R. Henson, "Computerized Satellite Tracking," *73 Magazine*, February 1977.

6.2. DISCUSSION

Amateur radio operators make extensive use of OSCARS* (Orbiting Satellites Carrying Amateur Radio) for long-range communications. The OSCARS are in near-circular, sun-synchronous polar orbits at an altitude of about 1500 km. OSCAR comes within range of any given spot on earth twice each day, local morning heading south and local evening heading north, and may be within range for as much as 25 min.

An ephemeris provides time and longitude of orbital equatorial crossings (EQX). The problem is: Given the latitude and longitude of a ground station, determine whether or not a given orbit can be viewed and, if so, find the time the bird rises over the horizon and from then on determine its range, bearing, and elevation at desired times until it disappears below the horizon. Figures 6.1 and 6.2, prepared by W. B. Graham (K6QB) of The Rand Corporation, show an elegant solution for his station in Pacific Palisades, California. (See examples below.)

The problem is clearly of military as well as amateur interest.

6.3. EQUATIONS

Notation:

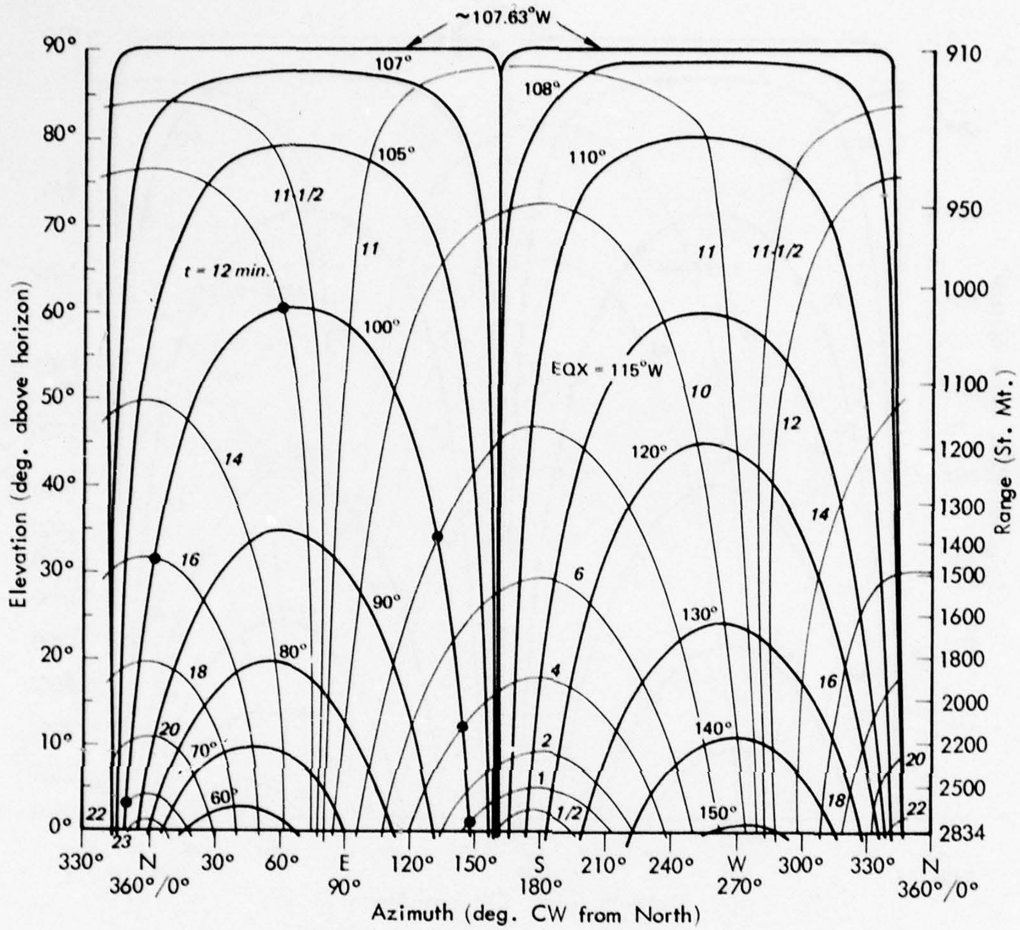
A = station latitude (north only)

B = station longitude (- if west)

β = orbital inclination measured counterclockwise from the equation (<90° posigrade, >90° retrograde)

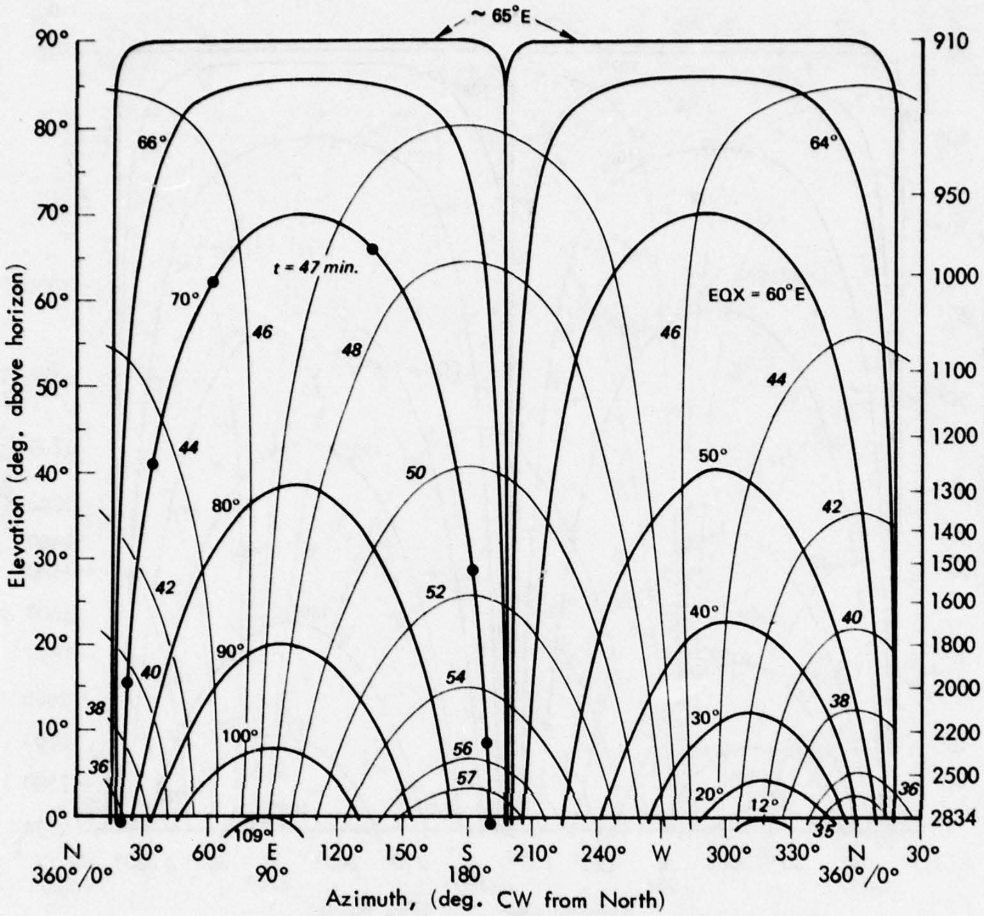
---

*For details on present and future OSCARS, see a series of articles in *QST* magazine (ARRL) starting with the January 1977 issue (Vol. 61, No. 1).



Azimuth and elevation of OSCAR 7 from Pacific Palisades, Ca.
as a function of time (t, minutes) after
Northbound Equatorial Crossing (EQX, W°).
For: 60°W < EQX < 150°W

Fig. 6.1 — OSCAR 7 ascending



Azimuth and elevation of OSCAR 7 from Pacific Palisades, Ca.
as a function of time (t, minutes) after
Northbound Equatorial Crossing (EQX, E°)
For: 12°E < EQX < 110°E

Fig. 6.2— OSCAR 7 descending

H = orbital altitude (n mi)

T = orbital period (min)

α = latitude of subsatellite point (SSP) at time t

γ = longitude of SSP at t

t = time from EQX (min)

γ_0 = longitude of EQX northbound (- if west, t = 0 at this node)

R_E = earth's radius (3437.9 n mi)

D_0 = arc from station to ascending node (EQX)

D_1 = arc to point of tangency of suborbit with latitude β

D_* = arc to SSP of closest approach

D = arc to SSP

t_* = time of closest approach

$t_* \mp \Delta t_*$ = time to rise above (go below) the horizon

θ = bearing from station north

ϕ = elevation above horizon

R = range from station to satellite (n mi).

By the law of sines,

$$\alpha(t) = \sin^{-1}\left[\sin \beta \cdot \sin (360\ t/T)\right] . \tag{1}$$

By the law of cosines,

$$\gamma(t) = \cos^{-1}\left[\cos (360\ t/T)/\cos \alpha(t)\right] - \frac{t}{4} + \gamma_0 , \tag{2}$$

where t/4 is the correction for the earth's rotation. Also by the law of cosines,

$$D(t) = \cos^{-1}\left[\sin A \cdot \sin \alpha + \cos A \cdot \cos \alpha \cdot \cos (B - \gamma)\right] . \tag{3}$$

The equations for θ, ϕ, and R are:

$$\theta(t) = \cos^{-1}\left[\frac{\sin\alpha - \sin A \cdot \cos D}{\cos A \cdot \sin D}\right] . \quad (4)$$

$$\phi(t) = \tan^{-1}\left[\frac{\cos D - 1/(1 + H/R_E)}{\sin D}\right] . \quad (5)$$

$$R(t) = (H + R_E) \sin D/\cos \phi . \quad (6)$$

The above equations are essentially those of Ref. a, except for (6).

Critical times are dealt with by the following equations (which are somewhat in error because their derivation assumes the suborbital trace of the satellite is a great circle, which is not true on a rotating earth). For the time t_* of closest approach,

$$t_* = \frac{T}{360} \tan^{-1} (\cos D_1/\cos D_0) , \quad (7)$$

and the incremental times to zero elevation are

$$\Delta t_* = \mp \frac{T}{360} \cos \bar{D}/\cos D_* , \quad (8)$$

where $\bar{D}$ is the arc length to the SSP at zero elevation and

$$\cos \bar{D} = 1/(1 + H/R_E) . \quad (9)$$

Finally, the period in minutes is

$$T = \frac{2\pi}{60}\sqrt{\frac{R_E}{g}}\left(1 + \frac{H}{R_E}\right)^{3/2} . \quad (10)$$

6.4. PROGRAM NOTES

1. 045, 046: Change β to first quadrant to get latitude of orbit tangency.

2. 115, 116: For retrograde orbit, subtract in (2).

3. 117, 118, 119: These steps produce the sgn of t (± 1), since the HP-67 does not have this signum function.

4. 120, 121: For $t < 0$, south of the equator, move time backward so that posigrade and retrograde orbits reverse in getting SSP longitude by (2).

5. LBL ϕ corrects longitude if outside (-180, +180), $\gamma > 180$ becomes $-360 + \gamma$, and $\gamma < -180$ becomes $360 + \gamma$.

6. LBL 1 gets bearing clockwise from station north.

Example 1. In Fig. 6.1, the ground station is at 34°03'N and 118°33'W. The satellite is retrograde at an altitude of 790 n mi and has an inclination of 102°. Track the satellite when in view if the EQX is 110°W at time 0.

Solution.

34.03 f H (34.05) STO A; 118.33 f H (118.55) CHS STO B; 790 STO C; 102 STO D; 100 CHS STO E; -1 h STI.

A: $T = 115.11$, $t_* = 11.68$, -29.74 (will come in view).

R/S: RCL 5, $\gamma = -100.39$; RCL 4, $\alpha = 1.31$ (SSP)

R/S: RCL 3, $t = 0.043$; RCL 7, $\theta = 148.72$, RCL 8, $\phi = -1.27$

Time t for first appearance is slightly small. Try $t = 1$ STO 3, PRESS B, RCL 8 and get $\phi = 0.51$. Try $t = 0.8$ and get -0.12, which is good enough. The following table is prepared by successively storing t in 3 and recalling 7, 8, 9:

t(3)	θ(7)	φ(8)	R(9) (n mi = 1.1515 stat mi)
0.8	148.40	-0.12	2842
4	144.56	11.47	2154
8	132.20	34.11	1380
12	62.63	60.68	1014
16	1.62	31.70	1435
22	348.65	2.69	2654.

These values are plotted as solid bullets in Fig. 6.1.

Example 2. All values are those of Example 1, except the EQX is 70°E. Hence the satellite will approach Pacific Palisades from the north.

Solution. Proceed as before. Note $T/2 \doteq 57.5$ min and STO 3, Press B, RCL 7, $\theta = -124.39$ is the descending node (EQX, moving south). Time will now be measured positive to the south of the equator. STO -124.39 in register E. Change the sign of the inclination β to minus and STO D. Now go back to square one and redo the problem from LBL A on. The results are tabulated below and again plotted as solid bullets in Fig. 6.2 for comparison.

t	t(corr.)	θ	φ	R (stat mi)
-22	35.5	19.14	-.19	2847
-18	39.5	23.88	14.95	1991
-14	43.5	36.65	41.33	1241
-12	45.5	61.85	61.46	1008
-10	47.5	136.24	65.43	981
-6	51.5	181.04	28.90	1506
-2	55.5	188.51	8.23	2323
+.5	58.0	190.72	-.47	2866

Ephemeris EQXs are given in ZULU time (GMT or UTC - Universal Coordinated Time for the radio amateur). Correct for local station time.

6.5 USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	STO DATA IN R_0, R_1, R_2			
2	STATION LATITUDE, STO A (CONVERT TO DEC DEGS BY fH IF NECESSARY)			
3	STATION LONGITUDE STO B (- FOR W. LONG, CONVERT TO DEC DEGS BY fH IF NEC)			
4	ALTITUDE (n. mi.), STO C			
5	ORBITAL INCLINATION, STO D (CC FROM EQUATOR)			
6	EQX LONGITUDE, STO E			
7	FOR POSIGRADE, 1 h ST I FOR RETROGRADE, -1 h ST I			
8	PRESS A, FIRST f-x-SHOWS T			
9	SECOND f-x-SHOWS TIME OF CLOSEST APPROACH			
10	ON R/S, IF NUMBER IS POSITIVE SATELLITE WILL NOT COME IN VIEW.			
11	PRESS R/S. ON HALT, RCL 5 TO GET APPROX LONG OF SSP ON FIRST APPEARANCE. RECORD. RCL 4 TO GET LATITUDE OF SSP.			
12	PRESS R/S. ON HALT, RANGE IS DISPLAYED. RCL 3 TO GET TIME, RCL 7 TO GET BEARING, RCL 8 TO GET ELEVATION			

6.5 USER INSTRUCTIONS

6. SATELLITE TRACKING

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
13	ELEVATION WILL NOT BE EXACTLY 0.				
	CHOOSE t TO ADJUST, STO 3, PRESS B				
	TO GET ADJUSTED ELEVATION.				
	REPEAT THIS REFINEMENT AS DESIRED.				
14	THEN t STO 3 ($t \neq 0$).				
	PRESS B. RCL 7 TO GET θ .				
	RCL 8 TO GET ϕ. RCL 9 TO GET R.				
	TRACK SATELLITE TO DISAPPEARANCE.				
15	SEE EXAMPLE 2, SECTION 6.4, FOR				
	DESCENDING ORBITS				
16	THE SSP LAT AND LONG ARE FOUND				
	BY RCL 4, RCL 5				

PRECEDING PAGE BLANK

6.6 SATELLITE TRACKING

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11		057	STO6	35 06	t_* (7)
002	RCLC	36 13		058	P⇄S	16-51	(PRI)
003	RCL0	36 00	R_E	059	STO3	35 03	
004	÷	-24		060	PRTX	-14	DSP t_*
005	1	01		061	GSB4	23 04	
006	+	-55	$1+H/R_E$	062	GSB5	23 05	
007	P⇄S	16-51	(SEC)	063	GSB6	23 06	
008	STO3	35 03		064	P⇄S	16-51	(SEC)
009	1/X	52		065	STO5	35 05	
010	COS^{-1}	16 42		066	RCL0	36 00	
011	STO0	35 00	$\bar{D}$ (9)	067	-	-45	IF > 0, SAT WILL NOT COME IN VIEW
012	RCL3	36 03		068	R/S	51	
013	1	01		069	RCL0	36 00	
014	.	-62		070	COS	42	
015	5	05		071	RCL5	36 05	
016	Y^X	31		072	COS	42	(8)
017	P⇄S	16-51	(PRI)	073	÷	-24	
018	RCL1	36 01		074	COS^{-1}	16 42	
019	×	-35		075	RCL2	36 02	
020	P⇄S	16-51	(SEC)	076	÷	-24	
021	STO1	35 01	T (10)	077	STO7	35 07	Δt_*
022	3	03		078	P⇄S	16-51	(PRI)
023	6	06		079	ST-3	35-45 03	
024	0	00		080	GTOB	22 12	
025	÷	-24		081	RTN	24	
026	1/X	52		082	*LBLB	21 12	
027	STO2	35 02	360/T	083	GSB4	23 04	
028	RCL1	36 01		084	GSB5	23 05	
029	PRTX	-14	DSP T (PERIOD)	085	GSB6	23 06	
030	P⇄S	16-51	(PRI)	086	GSB7	23 07	
031	0	00		087	GSB8	23 08	
032	STO4	35 04	α_0	088	GSB9	23 09	
033	RCLE	36 15		089	RCL8	36 08	DISP. RANGE
034	STO5	35 05	γ_0	090	RTN	24	
035	GSB6	23 06		091	*LBL4	21 04	(1)
036	COS	42		092	RCL3	36 03	
037	P⇄S	16-51	(SEC)	093	P⇄S	16-51	(SEC)
038	STO4	35 04	COS D_0	094	RCL2	36 02	
039	RCL1	36 01		095	×	-35	
040	4	04		096	SIN	41	
041	÷	-24		097	RCLD	36 14	
042	P⇄S	16-51	(PRI)	098	SIN	41	
043	STO3	35 03	T/4	099	×	-35	
044	RCLD	36 14		100	SIN^{-1}	16 41	
045	SIN	41	(PRGM NOTE 1)	101	P⇄S	16-51	(PRI)
046	SIN^{-1}	16 41		102	STO4	35 04	
047	STO4	35 04		103	RTN	24	
048	GSB5	23 05		104	*LBL5	21 05	(2)
049	GSB6	23 06		105	RCL3	36 03	
050	COS	42		106	P⇄S	16-51	(SEC)
051	P⇄S	16-51	(SEC)	107	RCL2	36 02	
052	RCL4	36 04		108	×	-35	
053	÷	-24		109	COS	42	
054	TAN^{-1}	16 43		110	P⇄S	16-51	(PRI)
055	RCL2	36 02		111	RCL4	36 04	
056	÷	-24		112	COS	42	

REGISTERS

0	1	2	3	4	5	6	7	8	9
3437.9	84.41	360	t	α	γ	D	θ	ϕ	R

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
$\bar{D}$	T	360/T	$1+H/R_E$	COS D_0	D_*	t_*	Δt_*		

A	B	C	D	E	I
A	B	H	β	γ_0	±1

6.6 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	÷	-24	
114	COS⁻¹	16 42	
115	RCLI	36 46	± (POSI, RETRO)
116	×	-35	
117	RCL3	36 03	
118	RCL3	36 03	(PRGM NOTE 3)
119	ABS	16 31	
120	÷	-24	SGN
121	×	-35	
122	RCL3	36 03	
123	4	04	
124	÷	-24	
125	-	-45	
126	RCLE	36 15	
127	+	-55	
128	STO5	35 05	γ
129	ABS	16 31	
130	1	01	
131	8	08	
132	0	00	
133	X⇄Y	-41	
134	X>Y?	16-34	$\lvert\gamma\rvert > 180^\circ$
135	GTO0	22 00	
136	RTN	24	
137	*LBL0	21 00	(PRGM NOTE 5)
138	RCL5	36 05	
139	RCL5	36 05	
140	ABS	16 31	
141	÷	-24	SGN
142	RCL2	36 02	
143	×	-35	
144	CHS	-22	
145	RCL5	36 05	
146	+	-55	
147	STO5	35 05	CORRECTED LONG.
148	RTN	24	
149	*LBL6	21 06	(3)
150	RCLB	36 12	
151	RCL5	36 05	
152	-	-45	
153	COS	42	
154	RCL4	36 04	
155	COS	42	
156	×	-35	
157	RCLA	36 11	
158	COS	42	
159	×	-35	
160	RCL4	36 04	
161	SIN	41	
162	RCLA	36 11	
163	SIN	41	
164	×	-35	
165	+	-55	
166	COS⁻¹	16 42	
167	STO6	35 06	D (t)
168	RTN	24	

STEP	KEY ENTRY	KEY CODE	COMMENTS
169	*LBL7	21 07	(4)
170	RCL4	36 04	
171	SIN	41	
172	RCLA	36 11	
173	SIN	41	
174	RCL6	36 06	
175	COS	42	
176	×	-35	
177	-	-45	
178	RCL6	36 06	
179	SIN	41	
180	÷	-24	
181	RCLA	36 11	
182	COS	42	
183	÷	-24	
184	COS⁻¹	16 42	
185	STO7	35 07	θ
186	RCL5	36 05	
187	RCLB	36 12	
188	X>Y?	16-34	
189	GTO1	22 01	
190	RTN	24	
191	*LBL8	21 08	(5)
192	RCL6	36 06	
193	COS	42	
194	P⇄S	16-51	(SEC)
195	RCL3	36 03	
196	1/X	52	
197	-	-45	
198	P⇄S	16-51	(PRI)
199	RCL6	36 06	
200	SIN	41	
201	÷	-24	
202	TAN⁻¹	16 43	
203	STO8	35 08	
204	RTN	24	
205	*LBL9	21 09	(6)
206	RCL6	36 06	
207	SIN	41	
208	RCL8	36 08	
209	COS	42	
210	÷	-24	
211	RCLC	36 13	
212	RCL0	36 00	
213	+	-55	
214	×	-35	
215	STO9	35 09	
216	RTN	24	
217	*LBL1	21 01	(PRGM NOTE 6)
218	RCL2	36 02	
219	RCL7	36 07	
220	-	-45	
221	STO7	35 07	
222	RTN	24	

LABELS					FLAGS
A	B	C	D	E	0
a	b	c	d	e	1
0	1	2	3	4	2
5	6	7	8	9	3

SET STATUS		
FLAGS (ON OFF)	TRIG	DISP
0 ☐ ☐	DEG ☐	FIX ☐
1 ☐ ☐	GRAD ☐	SCI ☐
2 ☐ ☐	RAD ☐	ENG ☐
3 ☐ ☐		n_____

PART II

MILITARY MODELS

7. THE DEER HUNT (DEFENSELESS BOMBERS)

7.1. REFERENCES

a. C. H. Builder, *The Penetration Integral and Tables*, The Rand Corporation, R-1257-PR, June 1973.

b. N.J.J. Bailey, *The Elements of Stochastic Processes*, John Wiley and Sons, New York, 1964.

c. A. T. Bharucha-Reid, *Elements of the Theory of Markov Processes and Their Applications*, McGraw-Hill, New York, 1960.

7.2. DISCUSSION

Reference a deduces an integral and constructs tables to assess the *expected* outcome of a one-sided, time-limited battle in which a set of armament-limited interceptors engages a set of defenseless penetrating bombers. The formulation also applies to a set of vessels transiting a minefield where each mine has a fixed number of warheads of some description. There are undoubtedly other military applications.

Builder's original paradigm is preferred. At the beginning of a hunt, there are A deer and B hunters, each of the hunters armed with m rounds of ammunition. Encounters are at random with parameter λ. A hunter can expend only one round on each engagement, with kill probability p. The hunt lasts T units of time.

In Ref. a, the parameters adopted are

$$j = \lambda pBT \qquad k = pmB/A , \tag{1}$$

which provide a bridge to this discussion. The parameter j is the potential number of lethal encounters per deer during the hunt, based upon the expected encounter rate, while k is the potential number of lethal encounters per deer based upon the total hunter armament. The parameters λ and p are perhaps the more natural ones to employ.

A much simpler approach to finding the expected values than that of Ref. a is adopted here. This approach is adequately illustrated by the cases m = 1, m = 2.

For $m = 1$,

$$\frac{da}{dt} = -\lambda pab \qquad \frac{db}{dt} = -\lambda ab \ , \tag{2}$$

where a and b are, respectively, the number of deer and the number of armed hunters remaining at time t. Division gives a first integral $a = pb + \alpha$, where $\alpha = A - pB$. Whence

$$a(t) = \frac{A(A - pB)}{A - pB \exp\{-(A - pB)\ \lambda t\}} \ , \tag{3}$$

and $a(t)/A$ is precisely the P_s of Ref. a if (1) is used.

The a and b found by (2) were called the *expected* values. Actually they are the *deterministic* values. Using the methods of Ref. b (p. 118), which set up a partial differential equation for the moment-generating function,

$$\frac{d\mu_{10}}{dt} = -\lambda p\mu_{11} \qquad \frac{d\mu_{01}}{dt} = -\lambda\mu_{11} \ , \tag{4}$$

where μ_{10}, μ_{01} are, respectively, the means or expected values of a and b, and μ_{11} is the correlation between a and b. Hence for (4) to agree with (2) we must have

$$\mu_{11} = \mu_{10} \cdot \mu_{01}$$

or Exp (ab) = Exp (a) · Exp (b). This is not true because the population sizes are mutually dependent. (See Ref. c, p. 184.) Consequently, we proceed with the understanding that we are dealing with deterministic values rather than expected values.

Turn to the case $m = 2$. At time t the values are a, b_1, b_2, where b_1 hunters have one round left and b_2 still have one round pouched. The deterministic equations are:

$$\frac{da}{dt} = -\lambda pa(b_1 + b_2) \ , \quad \frac{db_1}{dt} = -\lambda ab_1 + \lambda ab_2 \ , \quad \frac{db_2}{dt} = -\lambda ab_2 \ . \tag{5}$$

Two integrals are found immediately:

$$a - pb_1 - 2pb_2 = A - 2pb$$
$$b_1 = -b_2 \ln b_2/B \ . \qquad (6)$$

Whence,

$$\frac{db_2}{dt} = -\lambda p b_2^2 [2 - \ln b_2/B] - \lambda b_2 (A - 2pB) \ . \qquad (7)$$

For particular values of the parameters, (7) is readily integrated by Program 11 of the HP-67 Math Pac 1. The results agree exactly with those of the table for $m = 2$ in Ref. a.

In principle, for $m > 2$ numerical integration is possible. In practice this would be very time-consuming, if not infeasible, for the HP-67.* The next section provides a completely different, although *heuristic*, approach.

7.3. EQUATIONS

Again take $m = 2$. (The symmetric equations to be derived are readily extended to a general m.) On each encounter, A decreases on the average by p. Hence $a(n) = A - np$ is the expected number of deer just after the n*th* encounter. The average time to the next encounter is $\Delta t = 1/(\lambda a(n) \cdot b(n))$, where $b(n) = b_1(n) + b_2(n)$ is the remaining number of armed hunters after the n*th* encounter. Then

$$b_1(n + 1) = b_1(n) - 1 \ , \qquad b_2(n + 1) = b_n$$

with probability $b_1(n)/b(n)$, and

$$b_1(n + 1) = b_1(n) + 1 \ , \qquad b_2(n) = b_2(n) - 1$$

with probability $b_2(n)/b(n)$.

---

*But see Sec. 20 for m = 3, 4 application.

Hence the *expected* next state is

$$b_1(n + 1) = [b_1(n) - 1] \cdot b_1(n)/b(n)$$

$$+ [b_1(n) + 1] \cdot b_2(n)/b(n) .$$

$$b_2(n + 1) = b_2(n)\, b_1(n)/b(n)$$

$$+ [b_2(n) - 1]\, b_2(n)/b(n) .$$

These relations reduce to

[illegible]

[illegible]

[illegible] (8)

[illegible]

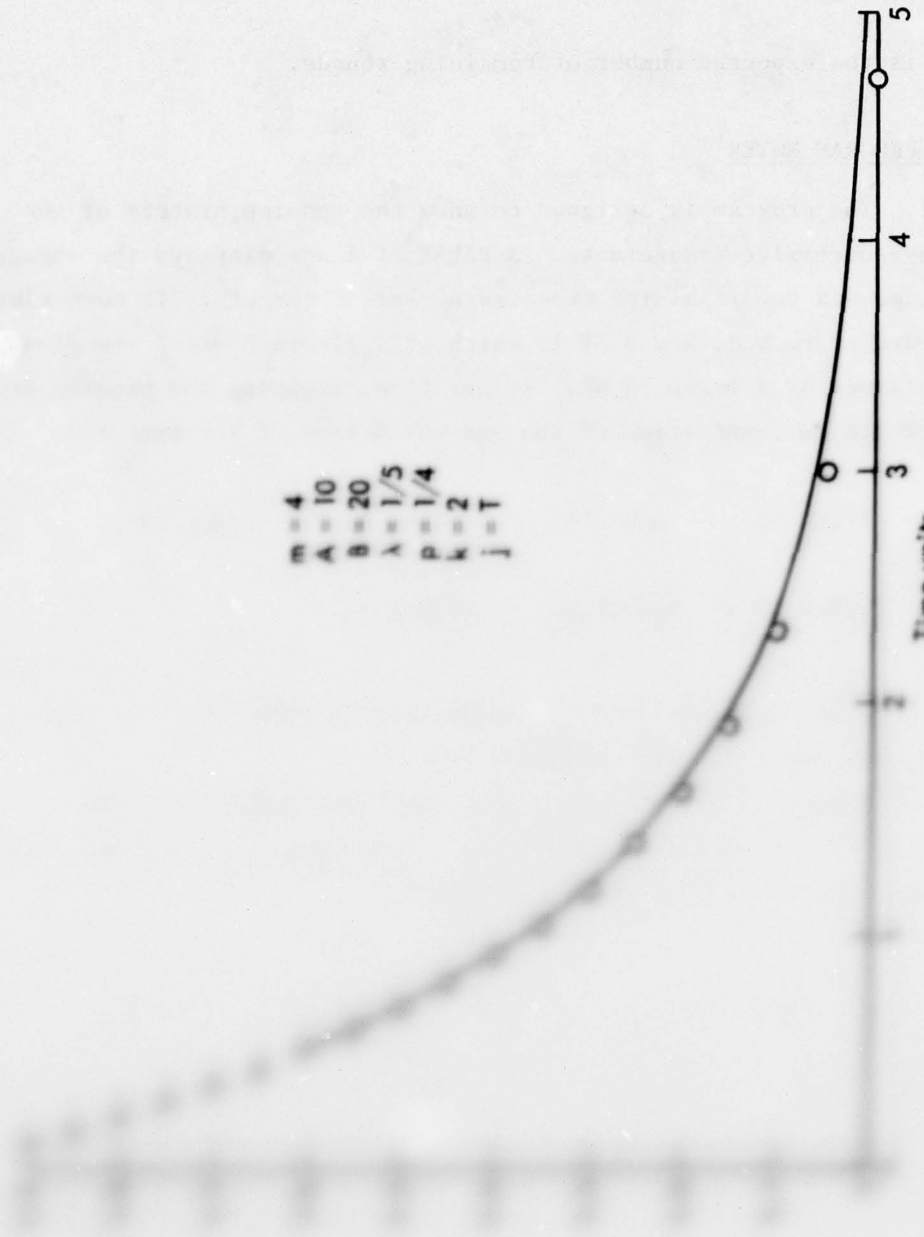


Fig. 7.1—Program vs Ref. (a)

and

$$b_1 + 2b_2 + 3b_3 + 4b_4 = 40.00 \ ,$$

which is the expected number of remaining rounds.

7.4. PROGRAM NOTES

1. The program is designed to show the running history of the hunt by successive engagements. A PAUSE of 1 sec displays the engagement time and the remaining expected number of targets. If more time is needed to record, key h SF 1, which will give a 5-sec f -x- flashing followed by a 1-sec PAUSE. At any time, stopping the program by R/S permits an examination of the current status of the hunt by

t(RCL 9) , a(RCL A) , b(RCL B) , b_4(RCL 8) ,

b_3(RCL 6) , b_2(RCL 4) , b_1(RCL 2) .

2. f LBL A directs action to the correct label for m by dissecting the input #,m and using GTO (i).

3. Extensive use of subroutines leads to program economy.

4. It will also be remarked that a systematic use of label and register addresses eases the programming.

7.5 USER INSTRUCTIONS

1 **7. THE DEER HUNT ($m \leq 4$)** 2

A (STO A) B.m (STO B) λ STO C p STO D

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	f CL REG (IMPORTANT)			
2	INITIALIZE BY STORING AS SHOWN ON THE CARD. (20 HUNTERS, EACH WITH 4 ROUNDS IS STORED AS 20.4)			
3	PRESS A			
4	ON EACH PAUSE, t AND THE REMAINING TARGETS a ARE DISPLAYED, TO MONITOR THE HUNT			
5	IF MORE TIME FOR RECORDING IS DESIRED, KEY h SF 1 TO GET A 5 SECOND PAUSE			
6	THE HUNT STATUS MAY BE REVIEWED AT ANY TIME BY R/S THEN FOR			
	t (RCL 9)			
	a (RCL A)			
	b (RCL B)			
	[illegible]			
	[illegible]			
	[illegible]			
	[illegible]			

7.6 THE DEER HUNT

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	001 *LBLA	21 11	
	002 RCLB	36 12	B.m
	003 STOI	35 46	
	004 INT	16 34	B
	005 STOB	35 12	
	006 RCLI	36 46	
	007 FRC	16 44	
	008 1	01	
	009 0	00	
010	010 ×	-35	
	011 STOI	35 46	m
	012 GTOi	22 45	GTO (i)
	013 *LBL4	21 04	m = 4
	014 RCLB	36 12	
	015 STO0	35 00	
	016 STO7	35 07	
	017 GTO9	22 09	
	018 *LBL9	21 09	
	019 RCLA	36 11	
020	020 STOI	35 46	a(n)
	021 1	01	
	022 RCL0	36 00	
	023 1/X	52	
	024 -	-45	
	025 STOE	35 15	1 - 1/b(n)
	026 RCL7	36 07	
	027 ×	-35	
	028 STO8	35 08	$b_4(n+1)$
	029 GSBE	23 15	
030	030 RCL7	36 07	
	031 RCL0	36 00	
	032 ÷	-24	
	033 +	-55	
	034 STO6	35 06	$b_3(n+1)$
	035 GSBD	23 14	
	036 RCL5	36 05	
	037 RCL0	36 00	
	038 ÷	-24	
	039 +	-55	
040	040 STO4	35 04	$b_2(n+1)$
	041 GSBC	23 13	
	042 RCL3	36 03	
	043 RCL0	36 00	
	044 [illegible]	-24	
	045 [illegible]	-55	
	[illegible]	[illegible]	[illegible]
050	[illegible]	[illegible]	RESET VALUES
	057 STO0	35 00	
	058 GTO9	22 09	NEXT ENCOUNTER
	059 *LBL3	21 03	m = 3
060	060 RCLB	36 12	
	061 STO0	35 00	
	062 STO5	35 05	
	063 GTO8	22 08	
	064 *LBL8	21 08	
	065 RCLA	36 11	
	066 STOI	35 46	
	067 1	01	
	068 RCL0	36 00	
	069 1/X	52	
070	070 −	-45	
	071 STOE	35 15	
	072 RCL5	36 05	
	073 ×	-35	
	074 STO6	35 06	SAME PATTERN
	075 GSBD	23 14	
	076 RCL5	36 05	
	077 RCL0	36 00	
	078 ÷	-24	
	079 +	-55	
080	080 STO4	35 04	
	081 GSBC	23 13	
	082 RCL3	36 03	
	083 RCL0	36 00	
	084 ÷	-24	
	085 +	-55	
	086 STO2	35 02	
	087 GSBB	23 12	
	088 RCL6	36 06	
	089 STO5	35 05	
090	090 RCL4	36 04	RESET VALUES
	091 STO3	35 03	
	092 RCL2	36 02	
	093 STO1	35 01	
	094 RCLB	36 12	
	095 STO0	35 00	
	096 GTO8	22 08	
	[illegible]	21 02	m = 2
	[illegible]	36 12	
	[illegible]	35 00	
100	[illegible]	[illegible]	
	[illegible]	[illegible]	SAME PATTERN

[illegible]

7.6 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	GSBC	23 13	
114	RCL3	36 03	
115	RCL0	36 00	
116	÷	-24	
117	+	-55	
118	STO2	35 02	
119	GSBB	23 12	
120	RCL4	36 04	RESET VALUES
121	STO3	35 03	
122	RCL2	36 02	
123	STO1	35 01	
124	RCLB	36 12	
125	STO0	35 00	
126	GTO7	22 07	
127	*LBL1	21 01	m = 1
128	RCLA	36 11	a(n) < 0 ?
129	X<0?	16-45	
130	R/S	51	
131	RCLB	36 12	b(n+1)
132	STO0	35 00	
133	1	01	
134	−	-45	
135	STOB	35 12	
136	RCLA	36 11	a(n+1)
137	STOI	35 46	
138	RCLD	36 14	
139	−	-45	
140	STOA	35 11	
141	RCLI	36 46	
142	RCL0	36 00	
143	×	-35	
144	RCLC	36 13	
145	×	-35	
146	1/X	52	
147	ST+9	35-55 09	CURRENT t
148	RCL9	36 09	
149	F1?	16 23 01	
150	PRTX	-14	f-x- t
151	PSE	16 51	
152	RCLA	36 11	
153	F1?	16 23 01	
154	PRTX	-14	f-x- a(n+1)
155	PSE	16 51	b(n) < 0 ?
156	RCLB	36 12	
157	X<0?	16-44	
158	GTO1	22 01	
159	RTN	24	
160	[illegible]	[illegible]	COMMON SUBROUTINE
161	[illegible]	[illegible]	
162	[illegible]	[illegible]	[illegible]
163	[illegible]	[illegible]	
164	[illegible]	[illegible]	
165	[illegible]	[illegible]	[illegible]
166	[illegible]	[illegible]	
167	[illegible]	[illegible]	
168	[illegible]	[illegible]	

STEP	KEY ENTRY	KEY CODE	COMMENTS
169	RCL0	36 00	
170	÷	-24	
171	-	-45	
172	STOB	35 12	b(n+1)
173	RCLI	36 46	
174	RCL0	36 00	
175	×	-35	
176	RCLC	36 13	
177	×	-35	
178	1/X	52	1/λ a(n) b(n)
179	ST+9	35-55 09	CURRENT t
180	RCLI	36 46	
181	RCLD	36 14	a(n) - p
182	−	-45	
183	STOA	35 11	a(n+1)
184	RCL9	36 09	t
185	F1?	16 23 01	CHECK FOR PSE
186	PRTX	-14	f-x- t
187	PSE	16 51	
188	RCLA	36 11	
189	F1?	16 23 01	CHECK FOR PSE
190	PRTX	-14	f-x- a(n+1)
191	PSE	16 51	
192	RTN	24	
193	*LBLC	21 13	
194	RCLE	36 15	
195	RCL1	36 01	
196	×	-35	
197	RTN	24	
198	*LBLD	21 14	
199	RCLE	36 15	
200	RCL3	36 03	
201	×	-35	
202	RTN	24	
203	*LBLE	21 15	
204	RCLE	36 15	
205	RCL5	36 05	
206	×	-35	
207	RTN	24	

8. A BOMBER PENETRATION MODEL (DEFENDED BOMBERS)

8.1. REFERENCES

None.

8.2. DISCUSSION

This penetration model is offered solely as an example of how the HP-67 can assist the analyst in his preliminary study of the factors bearing on a problem and how these factors interact. Model-making is, or rather should be, an art form drawing the essential elements from reality and illuminating their articulation. The author holds to the view that an initial model should be economical and transparent. The ornaments come later.

A group of bombers with integrated fire control for mutually supporting self-defense against interceptors makes a corridor penetration to a set of targets. The payload of a bomber can be divided at pleasure between defense missiles (AAM) and ground attack munitions (ASM). This loading is decided prior to the mission by choosing the number of AAM to be fired against each interceptor based on the expected number to be encountered. At equal intervals of time during the penetration, a clump of interceptors comes within range of the bombers' AAMs. *The bombers fire first.* Each of the surviving interceptors of the clump makes a single pass, allocating fire uniformly over the bombers, and then withdraws from the battle. As the battle progresses, more and more AAMs are fired per surviving bomber at the new interceptor clumps.

What bomber loading maximizes ASMs delivered on target?

8.3. EQUATIONS

[illegible]

[illegible]

[illegible]

[illegible]

$K = A + r \cdot S$, the payload constant

r = ratio of ASM to AAM weights

k = AAMs expended per interceptor encountered

Q = probability an AAM will miss an interceptor

I = interceptors per clump

q = probability an interceptor will miss its target

N = duration of the penetration with unit of time the interval between interceptor mass attacks

T = ASMs on target

$y^{**}x = y^x$.

Then

$$B_{n+1} = B_n \cdot q^{**}(I \cdot Q^{**}k/B_n) \qquad (1)$$

because $I \cdot Q^{**}k$ *interceptors survive and this divided by* B_n *is the* expected number of passes per bomber.

The number of missiles fired per bomber on the n*th* engagement equals kI/B_n. Hence the number of AAMs per surviving bomber to target is

$$A = kI(1/B_1 + 1/B_2 + \ldots + 1/B_N) , \qquad (2)$$

if the planning assumptions were realized in combat. The number of bombers surviving to target equals B_{N+1}, since after the last engagement by B_N they in turn receive return fire. Finally,

$$T = B_{N+1}(K - A)/r . \qquad (3)$$

As k increases, B_{N+1} increases but $K - A$ decreases. This tradeoff implies the existence of a k to maximize T.

For our purposes, no attempt is made to determine analytically [illegible] quickly, and (2) this exploration is informative since it shows the sensitivity of the outcomes to k (as well as to the other parameters).

Example.

B = 10, K = 30, r = 3, Q = 1/2, I = 10, q = 1/2, N = 3

k	A	B_4	T
2	7.30	5.31	40.19
1	4.72	1.95	16.40
3	9.87	7.52	50.47
4	12.55	8.73	50.78
3.5	11.19	8.22	51.54

Note how flat the bombs-on-target curve is for k = 3 to 4. Going from k = 3.5 to 4 decreases bombs on target by 1.5 percent, but *increases* bombers saved by 5 percent of the original force. This saving is not trivial if bombers are to be recycled for follow-on attacks. It illustrates the insight that quickly prepared "toy" models can provide.

8.4. PROGRAM NOTES

None.

8.5 USER INSTRUCTIONS

8. A BOMBER PENETRATION MODEL

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS	
1	B STO B	10			
2	KEY K PRESS A	30		30.00	
3	KEY r PRESS R/S	3		3.00	
4	KEY Q PRESS R/S	.5		0.50	
5	KEY I PRESS R/S	10		10.00	
6	KEY q PRESS R/S	.5		0.50	
7	KEY N PRESS R/S	3		3.00	
8	KEY k PRESS B	3.5		51.54	T
9	THE REGISTER CONTENTS OF THE PROGRAM LISTING SHOW HOW TO RECALL OR REPLACE ANY PAPAMETERS OR VALUES				
10	RCL 0			8.22	B_4
11	RCL A			11.19	A

8.6 BOMBER PENETRATION

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	001 *LBLA	21 11	INPUTS
	002 STO1	35 01	K
	003 R/S	51	
	004 STO2	35 02	r
	005 R/S	51	
	006 STO3	35 03	Q
	007 R/S	51	
	008 STO4	35 04	I
	009 R/S	51	
010	010 STO5	35 05	q
	011 R/S	51	
	012 STO6	35 06	N
	013 RTN	24	
	014 *LBLB	21 12	
	015 STO9	35 09	k
	016 RCLB	36 12	
	017 STO0	35 00	B_1
	018 1/X	52	
	019 STO7	35 07	$1/B_1$
020	020 1	01	
	021 STOI	35 46	1 h st I
	022 GTOC	22 13	
	023 *LBLC	21 13	
	024 ISZI	16 26 46	
	025 RCL3	36 03	
	026 RCL9	36 09	
	027 Y^x	31	
	028 RCL4	36 04	
	029 ×	-35	
030	030 RCL0	36 00	B_n
	031 ÷	-24	
	032 RCL5	36 05	g
	033 X⇄Y	-41	TO GET y^x
	034 Y^x	31	
	035 RCL0	36 00	
	036 ×	-35	
	037 STO0	35 00	B_{n+1}
	038 1/X	52	
	039 ST+7	35-55 07	Σ
040	040 RCL6	36 06	
	041 1	01	
	042 +	-55	TO GET N + 1
	043 RCLI	36 46	
	044 X=Y?	16-33	PEN. COMPLETE ?
	045 GTOD	22 14	
	046 GTOC	22 13	LOOP
	047 *LBLD	21 14	
	048 RCL0	36 00	B_{N+1}
	049 1/X	52	
050	050 ST-7	35-45 07	
	051 RCL7	36 07	
	052 RCL4	36 04	
	053 ×	-35	
	054 RCL9	36 09	
	055 ×	-35	
	056 STOA	35 11	A (2)
	057 CHS	-22	
	058 RCL1	36 01	
	059 +	-55	
060	060 RCL2	36 02	(3)
	061 ÷	-24	
	062 RCL0	36 00	
	063 ×	-35	
	064 STOC	35 13	T IS DISPLAYED
	065 RTN	24	

REGISTERS

0	1	2	3	4	5	6	7	8	9
B_{n+1}	K	r	Q	I	q	N	$\Sigma 1/B_n$		k

A	B	C	D	E	I
A	B	T			

9. DAMAGE PROBABILITIES, PVN AND QVN TARGETS

9.1. REFERENCES

a. D. C. Kephart, *Some Aids for Estimating Damage Probabilities in Attacks Against Targets with P and Q Vulnerability Numbers*, The Rand Corporation, R-1168-PR, March 1973 (For Official Use Only).

b. *Physical Vulnerability Handbook--Nuclear Weapons*, Defense Intelligence Agency, 1975 update.

9.2. DISCUSSION

The program in this section was prepared by D. C. Kephart of The Rand Corporation. He has also written the program for Texas Instruments' SR-52 hand calculator.

The program gives damage probabilities for nuclear weapons applied against PVN and QVN point targets at the optimal airburst height. For these two classes of targets 'psi' is given by:

$$\text{Overpressure, psi} = 1.1216 \times 1.2^{v} \quad (v = \text{adjusted PVN})$$

$$\text{Dynamic pressure, psi} = 0.02893 \times 1.44^{v} \quad (v = \text{adjusted QVN})$$

9.3. EQUATIONS

Notation

VN.K = Vulnerability number

V = Integer part of VN.K

K = Fractional part of VN.K = [K-factor]/10

w = Warhead yield in kilotons

C = Weapon CEP in feet

A = VN adjustment

v = V + A adjusted vulnerability number

R = Weapon radius in feet

P = Single-shot probability of damage (SSPD)

Formulas for PVN

$$S = \frac{K}{2}\left(\frac{20}{w}\right)^{1/3} + \left\{\left[\frac{K}{2}\left(\frac{20}{w}\right)^{1/3}\right]^2 + 1 - K\right\}^{1/2}$$

$$A = \ln(S^2)/\ln(1.2)$$

$$v = V + A$$

$$R = w^{1/3}[6383.35 \times 0.8836^v] \quad \text{if } v \leq 20.5$$

$$R = w^{1/3}[1900.05 \times 0.9368^v] \quad \text{if } v > 20.5$$

$$P = 1 - \exp\left\{-R^2/2/[C^2/\ln(4) + 0.04R^2]\right\}$$

Formulas for QVN

S satisfies the cubic equation

$$S = [SK\left(\frac{20}{w}\right)^{1/3} + 1 - K]^{1/3}$$

$$A = \ln(S^3)\ln(1.2^2)$$

$$v = V + A$$

$$R = w^{1/3}[6561 \times 0.87918^v] \quad \text{if } v \leq 15.4$$

$$R = w^{1/3}[23.42 + 2736.9 \times 0.92883^v] \quad \text{if } v > 15.4$$

$$t = 1 - \exp\left\{-R^2/2/[C^2/\ln(4) + 0.09R^2]\right\}$$

$$P = t \quad \text{if } t \leq 0.82$$

$$Q = 2.826t - 0.94t^2 - 0.866 \quad \text{if } t > 0.82$$

$$P = Q \quad \text{if } Q \leq 1$$

$$P = 1.0 \quad \text{if } Q > 1 .$$

9.4. PROGRAM NOTES

None.

9.5. USER INSTRUCTIONS

VN.K	Sto A	(K = K factor, 21Q7 = 21.7, 42P6 = 42.6)
Yield, KT	Sto B	
CEP, ft	Sto C	
PVN	Key A	Calculate { Adjusted Vulnerability Number -pause- Weapon radius -pause- Single-shot probability of damage (SSPD)
QVN	Key E	
		Compute time 10-12 sec for PVN, 12-22 sec for QVN
Display SSPD	Key B	Calculate PD for one more shot
Key CEP, ft	Key C	Calculate SSPD [After Key A or Key E; quick SSPDs using new CEP values. Compute time 3 sec.]

EXAMPLE

21.9	Sto A			
1000 KT	Sto B			
5000 ft	Sto C			
		Adj VN	Weapon Radius	SSPD
VN = 21P9	Key A	12.509	13575.496	0.973
VN = 21Q9	Key E	17.253	7891.371	0.732
Display SSPD	Key B	PD = 0.928 (21Q9, 2 shots)		
Key 6000	Key C	SSPD = 0.627 (21Q9, 1000 KT, 6000 ft CEP)		

AD-A054 955 RAND CORP SANTA MONICA CALIF F/G 9/2

HAND CALCULATOR PROGRAMS FOR STAFF OFFICERS.(U)

APR 78 E W PAXSON

UNCLASSIFIED RAND/R-2280-RC NL

2 OF 3

AD

A054 955

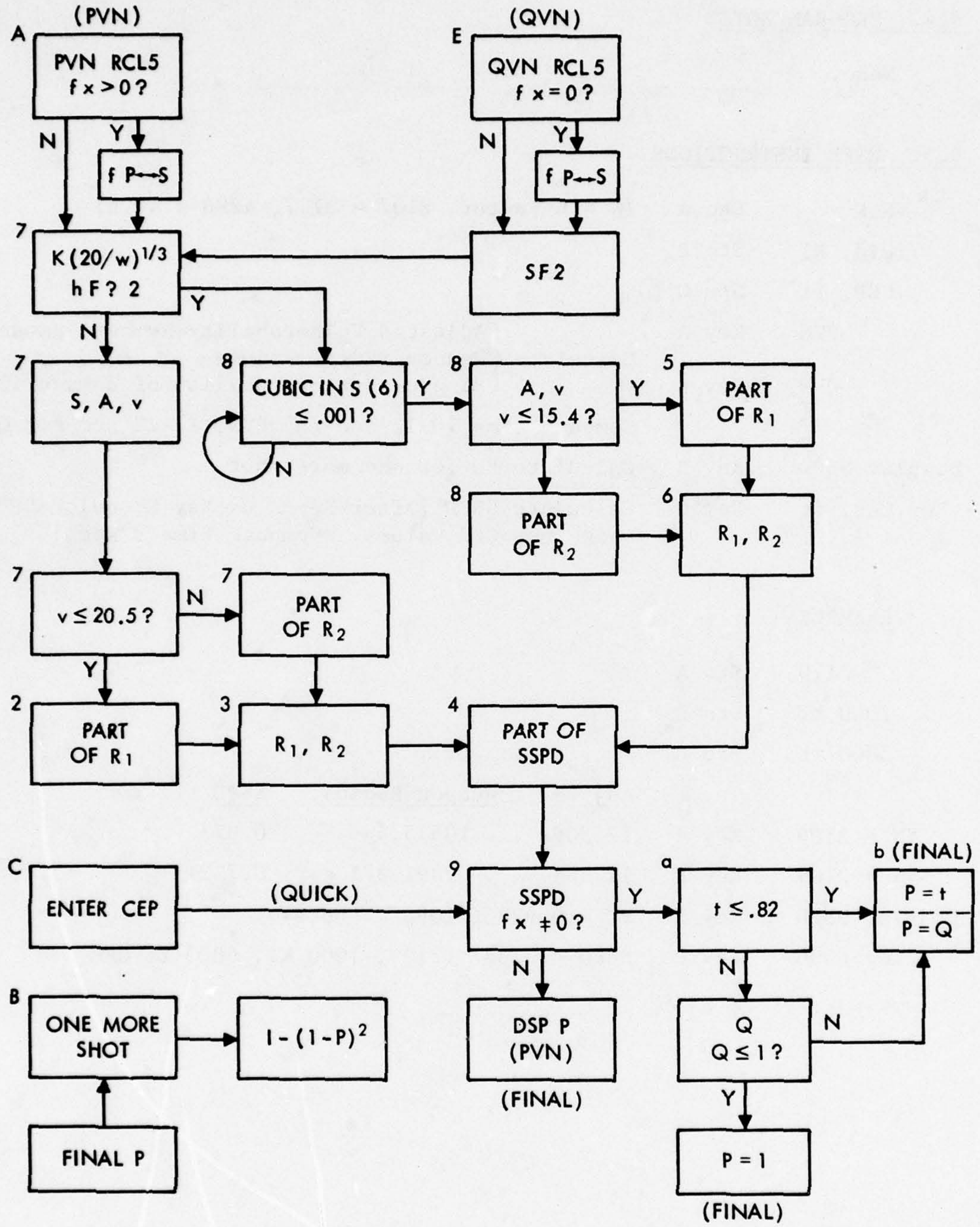


Fig. 9.1 Damage probability program logic skeleton

9.5 USER INSTRUCTIONS

1 9. DAMAGE PROBABILITIES, PVN AND QVN TARGETS 2

STEP	INSTRUCTIONS		INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	LOAD DATA AND PROGRAM					
2	VN.K STO A 21P9		21.9	STO	A	21.900
3	YIELD (KT) STO B		1000	STO	B	1000.000
4	CEP (FT) STO C		5000	STO	C	5000.000
5	PRESS A	ADJ VN			A	12.509
		WPN RAD.				13575.5
		SSPD				0.973
6	WITH ABOVE ENTRIES BUT					
	A 21 Q 9 TARGET	ADJ VN			E	17.253
		WPN. RAD.				7891.371
		SSPD				0.732
7	IF CEP IS 6000 FT AGAINST 21Q9		6000		C	0.627
8	FOR TWO SHOTS AGAINST 21Q9 (CEP = 6000)				B	0.861

9.6 DAMAGE PROBABILITIES

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	VN. K (PVN)
002	RCL5	36 05	IF NOT IN PRI,
003	X>0?	16-44	CHANGE TO PRI.
004	PSE	16-51	- - -
005	*LBL7	21 07	PRI
006	2	02	
007	0	00	
008	RCLB	36 12	
009	÷	-24	
010	RCLD	36 14	
011	Y^x	31	
012	RCLA	36 11	
013	FRC	16 44	
014	×	-35	
015	STO9	35 09	K $(20/w)^{1/3}$
016	F2?	16 23 02	IN SEC ?
017	GTO1	22 01	
018	2	02	
019	÷	-24	
020	X^2	53	
021	1	01	
022	+	-55	
023	RCLA	36 11	
024	FRC	16 44	
025	-	-45	
026	√X	54	
027	RCL9	36 09	
028	2	02	
029	÷	-24	
030	+	-55	(1)
031	LN	32	
032	2	02	
033	×	-35	$\ln S^2$
034	1	01	
035	.	-62	
036	2	02	
037	LN	32	
038	÷	-24	A (2)
039	RCLA	36 11	
040	INT	16 34	
041	+	-55	v (2)
042	PSE	16 51	DSP v (CAN USE
043	RCL0	36 00	f- x-)
044	X>Y?	16-34	BRANCH
045	GTO2	22 02	
046	R↓	-31	v
047	RCL4	36 04	
048	X⇄Y	-41	
049	Y^x	31	
050	RCL3	36 03	
051	×	-35	PART OF R_2 (4)
052	*LBL3	21 03	
053	RCLB	36 12	
054	RCLD	36 14	
055	Y^x	31	
056	×	-35	
057	PSE	16 51	DSP R_2, R_1
058	X^2	53	
059	STO9	35 09	R_2^2, R_1^2
060	.	-62	
061	0	00	
062	4	04	
063	*LBL4	21 04	
064	×	-35	
065	STO7	35 07	$.04\ R^2$, $.09\ R^2$
066	RCLC	36 13	
067	*LBL9	21 09	
068	X^2	53	
069	4	04	
070	LN	32	
071	÷	-24	
072	RCL7	36 07	
073	+	-55	
074	1/X	52	
075	2	02	
076	÷	-24	
077	RCL9	36 09	
078	×	-35	
079	CHS	-22	
080	e^x	33	
081	CHS	-22	
082	1	01	
083	+	-55	
084	STO8	35 08	P (5)
085	RCL5	36 05	
086	X≠0?	16-42	IN SEC ? (QVN)
087	GTOa	22 16 11	
088	RCL8	36 08	DSP P (PVN)
089	RTN	24	
090	*LBLa	21 16 11	
091	RCL8	36 08	
092	.	-62	
093	8	08	
094	2	02	
095	X>Y?	16-34	BRANCH
096	GTOb	22 16 12	
097	RCLE	36 15	
098	RCL8	36 08	
099	×	-35	
100	.	-62	
101	9	09	
102	4	04	
103	RCL8	36 08	
104	X^2	53	
105	×	-35	
106	-	-45	
107	RCLI	36 46	
108	-	-45	
109	STO8	35 08	Q (12)
110	1	01	
111	X≤Y?	16-35	I ≤ Q ? (13)
112	RTN	24	

REGISTERS

0	1	2	3	4	5	6	7	8	9
20.5	6383.35	0.8836	1900.05	0.9368			$.04 R_2^2$	P	$K(20/w)^{1/3}$

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
15.4	6561	0.8792	23.42	2736.9	0.9288	0.001	1, S_n	1- K, t, Q	$K(20/w)^{1/3}$

A	B	C	D	E	I
VN. K	KT	CEP (FT)	0.3333	2.826	0.866

9.6 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	*LBLb	21 16 12	DSP P (QVN) (11)
114	RCL8	36 08	
115	RTN	24	
116	*LBL2	21 02	
117	R↓	-31	v
118	RCL2	36 02	
119	X⇄Y	-41	
120	Y^x	31	
121	RCL1	36 01	
122	×	-35	
123	GTO3	22 03	PART OF R_1 (3)
124	*LBLE	21 15	VN, K (QVN)
125	RCL5	36 05	IF NEC, CHANGE
126	X=0?	16-43	TO SEC AND
127	P⇄S	16-51	SHOW BY F2
128	SF2	16 21 02	
129	GTO7	22 07	
130	*LBL1	21 01	(FROM 016)
131	1	01	
132	RCLA	36 11	
133	FRC	16 44	
134	-	-45	1-K
135	STO8	35 08	
136	1	01	
137	STO7	35 07	1
138	*LBL8	21 08	
139	RCL7	36 07	
140	RCL9	36 09	
141	×	-35	
142	RCL8	36 08	
143	+	-55	
144	RCLD	36 14	
145	Y^x	31	
146	RCL7	36 07	S_n
147	X⇄Y	-41	
148	STO7	35 07	S_{n+1}
149	X⇄Y	-41	
150	÷	-24	
151	1	01	
152	-	-45	
153	ABS	16 31	$\lvert 1 - S_{n+1}/S_n \rvert$
154	RCL6	36 06	
155	X≤Y?	16-35	TEST FOR ACC
156	GTO8	22 08	LOOP
157	RCL7	36 07	
158	LN	32	
159	1	01	
160	.	-62	
161	5	05	
162	×	-35	
163	1	01	
164	.	-62	
165	2	02	
166	LN	32	
167	÷	-24	A (7)
168	RCLA	36 11	

STEP	KEY ENTRY	KEY CODE	COMMENTS
169	INT	16 34	
170	+	-55	v (7)
171	PSE	16 51	DSP v (CAN USE f-x-)
172	RCL0	36 00	
173	X>Y?	16-34	v < 15.4 ?
174	GTO5	22 05	
175	R↓	-31	
176	RCL5	36 05	
177	X⇄Y	-41	
178	Y^x	31	
179	RCL4	36 04	
180	×	-35	
181	RCL3	36 03	
182	+	-55	PART OF R_2 (9)
183	*LBL6	21 06	
184	RCLB	36 12	
185	RCLD	36 14	
186	Y^x	31	$w^{1/2}$
187	×	-35	
188	PSE	16 51	DSP R_1, DSP R_2
189	X^2	53	(CAN USE
190	STO9	35 09	f-x-)
191	.	-62	
192	0	00	
193	9	09	
194	GTO4	22 04	
195	*LBL5	21 05	
196	R↓	-31	v
197	RCL2	36 02	
198	X⇄Y	-41	
199	Y^x	31	
200	RCL1	36 01	
201	×	-35	PART OF R_1 (8)
202	GTO6	22 06	
203	R/S	51	
204	*LBLC	21 13	(NEW CEP)
205	GTO9	22 09	
206	*LBLB	21 12	(ONE MORE SHOT)
207	CHS	-22	-P
208	1	01	
209	+	-55	
210	X^2	53	
211	CHS	-22	
212	1	01	
213	+	-55	$1-(1-P)^2$
214	R/S	51	

LABELS

DATA CARD ENTRIES ARE SHOWN BY ▭

FLAGS: 0 1 2 3

SET STATUS

FLAGS (ON OFF): 0 □ □ 1 □ □ 2 □ □ 3 □ □

TRIG: DEG □ GRAD □ RAD □

DISP: FIX □ SCI □ ENG □ n____

10. FOUR DEUCES (PRECISION 4.2-INCH MORTAR FIRE)

10.1. REFERENCES

a. FT 4.2-F-1, *Firing Tables, Mortar, 4.2-Inch, M30*, Department of the Army, December 1954.

b. FM 23-92, *4.2-Inch Mortar, M30*, Department of the Army, February 1961.

10.2. DISCUSSION

The 4.2-inch mortar, M30, is a rifled, muzzle-loaded weapon, known affectionately to the Army as the "four deuce." The tube, elevated at angles of 45° to 60° (800 mils to 1065 mils) delivers indirect fire to almost 5.5 kilometers, depending on the propellant charge, elevation, and round selected.

Indirect fire units use a meteorological message (a coded weather report) from division artillery in conjunction with unabridged firing tables to prepare firing data. As of 1961 (Ref. b) a rather time-consuming and largely manual procedure was used, which even sacrificed almost all tabular interpolation to save time and avoid errors. This section shows how the procedure of that era would have been simplified had a programmable hand-calculator been available then.

The primary task is to reduce the tables, which were based on range firings conducted at the Aberdeen Proving Grounds, to a set of formulas. This is accomplished by data fitting, a task for which the HP-67 is admirably suited if one has at hand Program 3 of the Standard Pac and Program 14 (Polynomial Approximation) of Stat Pac 1. Mark that this data fitting is purely empirical. It is based in no way on the physics and mathematics of exterior ballistics.

The only problem posed by the tailoring of formulas to number streams is the choice of the formula type to be used. Initial guidance is provided by plotting families of curves and staring at them. (See Sec. 21.)

The resulting formulas are simple and so is the required programming. You pay for this double simplicity but can exploit it. The large number of constants generated by fitting exceeds available storage space. But because the program is short, there is space to put constants, sometimes rounded, in the program itself. This is program/storage tradeoff.

The next subsection gives formulas that correct the fire for nonstandard conditions, in the order in which they will be programmed. This is also the approximate order of the manual calculations in the examples of Ref. a. The program significantly modifies the methods of that reference in respect to automatic interpolations, allowance of difference in altitude of mortar and target, ballistic winds, and elevation corrections.

10.3. EQUATIONS

General

(1) This program is restricted to the M30 firing the HE shell M329 with extension (long-range fire).

(2) The range of charges is 25.5 to 41 and increments of 1/8 are permitted.

(3) Elevations are restricted to 800 to 900 mils (₥), since the tables are so restricted for charges above 32.

(4) Hence charge is selected so that 850 ₥ (47.81°) elevation gives the *approximate* desired range (1 ₥ = 360/6400° = 0.05625 = 1/17.778).

(5) Meters rather than yards are used (1 yd = 0.9144 m).

Notation

H_o Altitude of mortar position in meters

R_o Chart range in meters

R_1 Range corrected for difference in altitude of mortar and target

R_2 R_1 corrected for metro and ballistic factors

R(800) Range for E = 800 ₥, a function of charge m

A_o Azimuth of fire (mortar to target) in mils, CW from N

E_0 850 ₥, the initial elevation

E_2 Corrected elevation (₥)

C_0 The initial charge selected

C_2 Corrected charge

ω The angle of fall in mils (impact angle)

T Powder temperature in F°

VE Muzzle velocity error in ft/sec for lot used (0 if not known. This can be obtained only from trial firings.)

r Deviation in shell weight from ⊡ ⊡ (-1, 0, +1). (Weight is measured in squares and stamped on the shell.)

H_T Altitude of target in meters

H_M Altitude of meteorological data plane (MDP) in *feet*

A_W Azimuth of ballistic wind in ₥ CW from N. (This is the direction *from* which the wind blows. It is an average of winds up to the maximum ordinate of the trajectory.)

α Angle between A_0 and A_W - 3200 in degrees

δ_0 Density of the air as percent of standard for the MDP altitude

δ_1 Corrected density for mortar altitude relative to MDP

W Ballistic wind in *miles per hour*

d Drift deflection because of shell rotation (always to the right)

D Deflection correction, final (₥)

ρ_1 Correction for range wind (tail or head), meters

ρ_2 Correction for round weight, meters

ρ_3 Correction for powder temperature, meters

ρ_4 Correction for actual air density, meters

ρ_5 Correction for VE, meters

$\rho = -(\rho_1 + \rho_2 + \rho_3 + \rho_4 + \rho_5)$, meters

P Change in muzzle velocity due to change in powder temperature (ft/sec)

Metro Msg

H_M, A_W, W, δ are known when the Metro Message is "solved." The message also gives air temperature, which is not relevant for mortar calculations. The message has 12 lines in addition to the heading. The initial digit is the standard altitude number. The correspondence is

Alt. No.	0	1	2	3	4	5
Height (ft)	0	600	1500	3000	4500	6000
Alt. No.	6	7	8	9	0	1
Height (ft)	9000	12000	15000	18000	24000	30000

The lines of the message give A_W, W, δ (and air temperature) appropriate for the maximum ordinate of the trajectory resulting from any particular combination of range, charge, and elevation.

For this program (800 ₥ $\le$ E $\le$ 900 ₥, R $\ge$ 3250 m), always use line 4 *except*

C = 25.5			Line 3
C = 41			Line 5
C $\ge$ 36	and	E $\ge$ 850	Line 5 .

Formulas

$$E_o = 850 \text{ ₥} \tag{1}$$

$$C_o = 11.932 \exp(0.000231\, R_o) \tag{2}$$

$$\omega = (867 + 4C_o)(360/6400) \tag{3}$$

$$R_1 = R_o + (H_T - H_o)/\tan\omega \tag{4}$$

($R_1 > R_o$ if target is above mortar)

$$\delta_1 = \delta_0 - 0.003\ (H_o/0.3048 - H_M) \tag{5}$$

(H_o is changed from meters to feet)

$$\alpha = 0.05625\ (A_W - 3200 - A_o) \tag{6}$$

$$d = 0.064\ E_o - (C_o + 1)/2 \tag{7}$$

$$D = -[d + 0.8\ W \sin \alpha] \tag{8}$$

(The tube is pointed for firing at $A_o + D$ ṁ)

$$\rho_1 = (0.325\ C_o - 4.682)\ W \cos \alpha \tag{9}$$

(The term in parentheses is the unit effect for a tail wind of 1 mph, Col. 15 of tables)

$$\rho_2 = (17.879 \ln C_o - 72.914)\ r \tag{10}$$

(Unit effect in parentheses Col. 17)

$$P = -23 + 0.68\ T - 0.005\ T^2, \quad T \leq 70^\circ$$
$$= 15.29 - 0.48\ T + 0.0038\ T^2, \quad T > 70^\circ \tag{11}$$

(Table of App. A of Ref. a, fitted)

$$\rho_3 = (16.3 - 2.5 \ln C_o)\ P \tag{12}$$

(Unit MV effect in parentheses Col. 18)

$$\rho_4 = (11.037 - 0.7293\ C_o)(\delta_1 - 100) \tag{13}$$

(Unit 1 percent effect in parentheses Col. 16)

$$\rho_5 = (16.3 - 2.5 \ln C_o) \cdot VE \tag{14}$$

$$\rho = -(\rho_1 + \rho_2 + \rho_3 + \rho_4 + \rho_5) \tag{15}$$

(Note the - sign)

$$R_2 = R_1 + \rho \tag{16}$$

This is the corrected final range.

$$C_2 = 11.932 \exp (0.000231\ R_2) \tag{17}$$

(This is the corrected charge for E = 350 ₥. It is *to be rounded* to the nearest 1/2 charge.)

$$R(800) = 52.51 + 81.73\ C_2 + 2.96 C_2^2 - 0.043\ C_2^3 \tag{18}$$

(Fitted from bottom line data of tables)

$$E_2 = 800 + [6.416 - 0.093\ C_2][R(800) - R_2]^{0.712}\ . \tag{19}$$

This is the final elevation at which to lay the tube. Note the correction of C_o using E = 850 ₥ and the final R_2. But then E = 850 ₥ is forgotten and the procedure is to go to *new* formulas to get E_2 with respect to the baseline elevation E = 800 ₥. This procedure yields a small correction to the initial 850 ₥.

Note: Because of lack of program/storage space, formula (11) is rewritten as

$$P = -(0.02\ \Delta T + 0.005\ \overline{\Delta T}^2) \qquad \Delta T \le 0 \tag{20}$$

$$= 0.05\ \Delta T + 0.0038\ \overline{\Delta T}^2 \qquad T > 0\ ,$$

and the user enters $\Delta T = T - 70°$, the deviation from the standard powder temperature with the proper sign.

Checks

To provide program verification, a step-by-step example is calculated manually. These results are then compared with the Army's field procedure, which does not require interpolation and rounds off various values. *Yards* instead of meters are employed in these checks.

R_o = 5320 (nearest 10 yd) , H_o = 1505 ft , H_T = 1210 ft ,

A_o = 4825 ₥ , wgt = ⊡ (r = -1) , T = 55°F ,

VE = - 12 ft/sec .

Solving the Metro Message (heading and line 5),

M 1 F 1 2 0 8 3 0 3

5 2 6 2 5 9 5 7 8 5 ,

we get: message from Station 1F, MDP = 1200 ft, as of 0830 hr; this is msg type 3 (for mortars); standard altitude for line 5 is 6000 ft; ballistic wind blows from 2600 ₥, strength 25 mph; air density is 95.7 percent, and air temperature is 85°F.

The above conditions were taken from Ref. a. They could be realized, for example, at Hunter Liggett Military Reservation in California on a day in November with Santa Ana winds blowing. Use Map Series V895S, Sheet 1755 1 NW, 1:25000. Put the mortar position on Hill 1516 (66960 73900) and the target close to the junction of two dirt roads and almost in the bed of Fria Creek (62095 74000).

Note: Yards rather than meters are used below.

Quantity	Field Method	Formulas
R_o	5300	5320
H_o	1500	1505
E_o	847	850
C_o	36.5	36.71
ω	1009	1014
R_1	5255*	5256
δ_1	94.8	94.8
d	38	35.6
D	-57	-52
ρ_1	113	114
ρ_2	9	9
ρ_3	-15	-11
ρ_4	88	90
ρ_5	-91	-96
ρ	104	106
R_2	5151	5150

The field method now selects a new charge for which the adjusted range R_2 is bracketed by a 50-yd tabular interval in the tables. Interpolation is now used in general. Here we can read off

$$C_2 = 35.5 \qquad E_2 = 854 \not{m} .$$

Formula (17) yields $C_2 = 35.41$. Round this to $C_2 = 35.5$ and obtain, via (18) and (19), $E_2 = 853 \not{m}$.

10.4. PROGRAM NOTES

"When I know more of gunnery
Than a novice in a nunnery,
I'll be the very model
Of a modern Major-General."

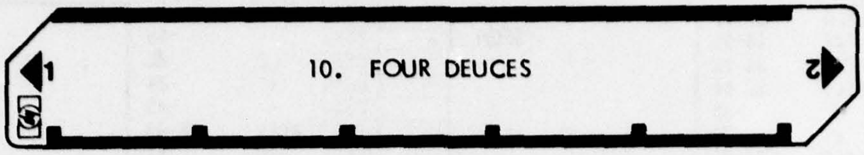
- W. S. Gilbert, *The Pirates of Penzance*, 1879

*The field method corrects 5320 by 1/2 the altitude difference in yards or by -50 yd. We have used the more accurate value -65 yd.

10.5. DATA CARD

```
 .3048 STO 7,    9/160 h ST I
          f P <-> S
11.932 STO 0,   .000231 STO 1,
 4.682 STO 2,   17.879  STO 3,
72.914 STO 4,   .0038   STO 5,
11.037 STO 6,   .7293   STO 7,
 52.51 STO 8,   81.73   STO 9.
          f P <-> S
RUN.  f W/DATA.  RECORD BOTH SIDES.
```

10.5 USER INSTRUCTIONS



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	LOAD DATA AND PROGRAM CARDS	EXAMPLE		
2	H_o(m) STO A	459		
3	R_o(m) STO B	4864.6		
4	A_o(ḿ) STO C	4825		
5	H_T (m) STO D	369		
6	r STO 0	-1		
7	T (°F) STO 1	-15		
8	VE (fps) STO 2	-12		
9	H_M(ft) STO 3	1200		
10	A_W(ḿ) STO 4	2600		
11	W (mph) STO 5	25		
	δ_o STO 6	95.7		
13	PRESS A/7 SECS, SEE CORRECTED AZ		A	4773
14	PRESS B/7 SECS, SEE CORRECTED CHARGE		B	35.41
15	KEY IN, ROUNDED TO NEAREST 1/2	35.5		
16	PRESS R/S (5 SECS), SEE CORRECTED EL			853
	<u>WARNING.</u> IF NEW RUN IS TO BE MADE WITHOUT RELOADING THEN 0 STO 8. ALSO REGISTERS B AND 6 MUST BE CHECKED.			

10.6 FOUR DEUCES

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	001 *LBLA	21 11	
	002 P⇄S	16-51	(SEC)
	003 RCLB	36 12	
	004 RCL1	36 01	
	005 ×	-35	
	006 e^x	33	
	007 RCL0	36 00	
	008 ×	-35	
	009 STOE	35 15	C_o (PRI)
010	010 P⇄S	16-51	
	011 RCLD	36 14	
	012 RCLA	36 11	
	013 -	-45	
	014 RCLE	36 15	
	015 4	04	
	016 ×	-35	
	017 8	08	
	018 6	06	
	019 7	07	
020	020 +	-55	
	021 RCLI	36 46	
	022 ×	-35	
	023 TAN	43	
	024 ÷	-24	
	025 RCLB	36 12	
	026 +	-55	
	027 STOB	35 12	R_1
	028 RCL6	36 06	
	029 RCLA	36 11	
030	030 RCL7	36 07	
	031 ÷	-24	
	032 RCL3	36 03	
	033 -	-45	
	034 3	03	
	035 ×	-35	
	036 EEX	-23	
	037 3	03	
	038 ÷	-24	
	039 -	-45	
040	040 STO6	35 06	δ_1
	041 RCL4	36 04	
	042 3	03	
	043 2	02	
	044 0	00	
	045 0	00	
	046 -	-45	
	047 RCLC	36 13	
	048 -	-45	
	049 RCLI	36 46	
050	050 ×	-35	
	051 STO9	35 09	α
	052 5	05	
	053 4	04	
	054 .	-62	
	055 4	04	
	056 RCLE	36 15	

STEP	KEY ENTRY	KEY CODE	COMMENTS
	057 1	01	
	058 +	-55	
	059 2	02	
060	060 ÷	-24	
	061 -	-45	
	062 RCL9	36 09	
	063 SIN	41	
	064 RCL5	36 05	
	065 ×	-35	
	066 .	-62	
	067 8	08	
	068 ×	-35	
	069 +	-55	
070	070 CHS	-22	
	071 RCLC	36 13	A_o + D RECORD
	072 +	-55	
	073 RTN	24	
	074 *LBLB	21 12	
	075 RCL9	36 09	
	076 COS	42	
	077 RCL5	36 05	
	078 ×	-35	
	079 RCLE	36 15	
080	080 .	-62	
	081 3	03	
	082 2	02	
	083 5	05	
	084 ×	-35	
	085 P⇄S	16-51	(SEC)
	086 RCL2	36 02	
	087 -	-45	
	088 ×	-35	
	089 P⇄S	16-51	(PRI)
090	090 STO8	35 08	ρ_1
	091 RCLE	36 15	
	092 LN	32	
	093 P⇄S	16-51	(SEC)
	094 RCL3	36 03	
	095 ×	-35	
	096 RCL4	36 04	
	097 -	-45	
	098 P⇄S	16-51	(PRI)
	099 RCL0	36 00	
100	100 ×	-35	ρ_2
	101 ST+8	35-55 08	
	102 0	00	
	103 RCL1	36 01	
	104 X≤Y?	16-35	
	105 GTO1	22 01	
	106 GTO2	22 02	
	107 *LBLC	21 13	
	108 1	01	
	109 6	06	
110	110 .	-62	
	111 3	03	
	112 RCLE	36 15	

REGISTERS

0	1	2	3	4	5	6	7	8	9
r	T-70	VE	H_M	A_W	W	δ_o/δ_1	.3048	$\Sigma\rho_i$	α
S0 11.932	S1 .000231	S2 4.682	S3 17.879	S4 72.914	S5 .0038	S6 11.037	S7 .7293	S8 52.51	S9 81.73

A	B	C	D	E	I
H_o	$R_o/R_1/R_2$	A_o	H_T	C_o/C_2	9/160 $^{m}_{o}$

DATA CARD ENTRIES ARE SHOWN AS ☐

10.6 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	LN	32	
114	.	-62	
115	4	04	
116	÷	-24	
117	-	-45	
118	STOD	35 14	MV. UNIT EFFECT
119	×	-35	
120	ST+8	35-55 08	ρ_3
121	RCL6	36 06	
122	EEX	-23	
123	2	02	
124	-	-45	
125	P⇄S	16-51	(SEC)
126	RCL6	36 06	
127	RCL7	36 07	
128	RCLE	36 15	
129	×	-35	
130	-	-45	
131	×	-35	ρ_4
132	P⇄S	16-51	(PRI)
133	ST+8	35-55 08	
134	RCL2	36 02	
135	RCLD	36 14	
136	×	-35	ρ_{5_ρ}
137	ST+8	35-55 08	
138	RCL8	36 08	
139	CHS	-22	
140	RCLB	36 12	
141	+	-55	
142	STOB	35 12	R_2
143	P⇄S	16-51	(SEC)
144	RCL1	36 01	
145	×	-35	
146	e^x	33	
147	RCL0	36 00	
148	×	-35	
149	R/S	51	ROUND R/S
150	STOE	35 15	C_2
151	3	03	
152	y^x	31	
153	.	-62	
154	0	00	
155	4	04	
156	3	03	
157	×	-35	
158	CHS	-22	
159	RCLE	36 15	
160	x^2	53	
161	2	02	
162	.	-62	
163	9	09	
164	6	06	
165	×	-35	
166	+	-55	
167	RCLE	36 15	
168	RCL9	36 09	
169	×	-35	
170	+	-55	
171	RCL8	36 08	
172	+	-55	
173	RCLB	36 12	
174	-	-45	
175	.	-62	
176	7	07	
177	1	01	
178	2	02	
179	y^x	31	
180	6	06	
181	.	-62	
182	4	04	
183	2	02	
184	RCLE	36 15	
185	.	-62	
186	0	00	
187	9	09	
188	×	-35	
189	-	-45	
190	×	-35	
191	8	08	
192	0	00	
193	0	00	
194	+	-55	E_2 RECORD (PRI)
195	P⇄S	16-51	
196	RTN	24	
197	*LBL1	21 01	
198	RCL1	36 01	
199	x^2	53	
200	2	02	
201	0	00	
202	0	00	
203	÷	-24	
204	CHS	-22	
205	RCL1	36 01	
206	5	05	
207	0	00	
208	÷	-24	
209	+	-55	
210	GTOC	22 13	
211	*LBL2	21 02	
212	RCL1	36 01	
213	2	02	
214	0	00	
215	÷	-24	
216	RCL1	36 01	
217	x^2	53	
218	P⇄S	16-51	(SEC)
219	RCL5	36 05	
220	×	-35	
221	+	-55	
222	P⇄S	16-51	(PRI)
223	GTOC	22 13	

LABELS

A AZ	B USED	C EL	D	E
a	b	c	d	e
0	1	2	3	4
5	6	7	8	9

FLAGS: 0, 1, 2, 3

SET STATUS

FLAGS (ON / OFF)	TRIG	DISP
0 ☐ ☒	DEG ☒	FIX ☒
1 ☐ ☒	GRAD ☐	SCI ☐
2 ☐ ☒	RAD ☐	ENG ☐
3 ☐ ☒		n____

11. A LASER EQUATION

11.1. REFERENCE

a. L. N. Peckham and R. W. Davis, *A Simplified Propagation Model for Laser System Studies*, Air Force Weapons Laboratory, Technical Report AFWL-TR-72-95, August 1972, ASTIA No. AD902736L.

11.2. DISCUSSION

The laser equation programmed in this section was brought to my attention by Lieutenant Colonel R. S. DeLaney, USAF. The equation applies to propagation in the atmosphere. A listing of the variables and parameters used in the equation shows the factors considered in it.

Symbol	Meaning	Units
P	power	watts
R	range	km
I	average intensity	watts/cm^2
b	blockage factor	--
K	thermal blooming factor	--
α	atmospheric extinction	1/km
k_1	power reduction factor	--
k_2	beamspread factor	--
λ	wavelength	microns
D	diameter of primary output mirror	meters
σ_{TR}	one sigma jitter/tracker	microradians
σ_{PL}	one sigma jitter/platform	microradians
σ_{BL}	one sigma jitter/boundary layer	microradians
σ_{AT}	one sigma jitter/atmosphere	microradians
θ	angle between beam and target normal	deg

11.3. EQUATIONS

$$I = \frac{100\ b \cdot K \cdot p \cdot \exp(-\alpha R) \cdot \cos\theta}{k_1 \cdot \pi R^2 [(0.9\ k_2\ \lambda/D)^2 + 4(\sigma_{TR}^2 + \sigma_{PL}^2 + \sigma_{BL}^2 + \sigma_{AT}^2)]} . \tag{1}$$

The number 100 is needed to get the intensity at the target in watts/cm^2 when R is in kilometers.

We will program three problems:

- Given P and R, find I
- Given I and R, find P
- Given I and P, find R.

In effect, the program is a digitized nomogram.

We will use Newton's method to get R given I and P. We have

$$I = LPe^{-\alpha R}/R^2 , \tag{2}$$

where

$$L = \frac{100 \cdot b \cdot K \cos\theta}{k_1 \cdot \pi[(0.9\ k_2\ \lambda/D)^2 + 4\sigma^2]} ,$$

$$\sigma^2 = \sigma_{TR}^2 + \sigma_{PL}^2 + \sigma_{BL}^2 + \sigma_{AT}^2 . \tag{3}$$

We want the root of

$$f(R) = LPe^{-\alpha R}/R^2 - I = 0 . \tag{4}$$

then

$$f'(R) = -(\alpha + 2/R)\ LPe^{-\alpha R}/R^2 , \tag{5}$$

$$R_{i+1} = R_i + \frac{LPe^{-\alpha R_i} - IR_i^2}{LPe^{-\alpha R_i}(\alpha + 2/R_i)} . \tag{6}$$

The only point of interest is the determination of a good starting value R_o. The function $e^{-\alpha R}/R^2$ is concave upward and extremely flat for even moderately large R. If an initial R_o is picked that is greater than R, it may well happen that the flat tangent projected backward will generate negative values for the successive R_i.

For the root of (4),

$$R^2 = Ae^{-\alpha R} , \qquad A = LP/I .$$

A first approximation is $\sqrt{A}$, but this is clearly too large. But taking

$$R_o = \sqrt{A}\, e^{-\frac{\alpha\sqrt{A}}{2}} , \qquad (7)$$

a value smaller than the actual root is found.

The program also computes the illuminated area by

$$0.01\ \pi R^2 [(0.9\ K_2\ \lambda/D)^2 + 4\sigma^2]\ \sec\theta . \qquad (8)$$

11.4. PROGRAM NOTES

(1) The program is an example of the use of the flag F3 to find the solution of any one of three problems. The program structure is:

P	R	I
*fLBLA	*fLBLB	*fLBLC
hF?3	hF?3	hF?3
GTO1	GTO2	GTO3
Calculate P	Calculate R	Calculate I
*fLBL1	*fLBL2	*fLBL3
STO A	STO B	STO C
hRTN	hRTN	hRTN

Flag F3 is set by data entry, and cleared by test. If you key P and press A, F3 is set, P is stored in A, and F3 is cleared. If you then key I and press C, I is stored in C. Now press B. Since F3 is not set, the step "GTO 2" is skipped and R is calculated.

(2) For the preceding problem, an "h PAUSE" shows the successive steps in the convergence to two decimal places. When P is large and/or I is small, the convergence is slow because the value of R_0 from (7) is small. You can speed up convergence by choosing an R_0 that is large but still smaller than the expected final root. Then execute the sequence: Key R_0, STO B, GTO 0, R/S.

Example. Let the parameters be:

$b = 0.9$, $K = 1$, $\alpha = 0.1$, $k_1 = 1.5$, $k_2 = 20/9$,

$\lambda = 3.6$, $D = 1$, each of the four sigmas = 4 (so that

σ^2 is 64), $\theta = 60°$. Then PRESS E to initialize.

(1) $P = 10^6$ (EEX 6), PRESS A, $R = 4$, PRESS B.

Now PRESS C to get $I = 1299.60$ watts/cm^2.

PRESS D to get 309.47 cm^2 for the illuminated area at this range.

(2) 4, PRESS B, 1299.60, PRESS C. PRESS A to get

$P = 1\,000\,002.64$ watts (because of roundoff in I).

(3) 10^6, PRESS A, 1299.60, PRESS C. PRESS B to get

$R = 4$ after two iterations (3.99, 4.00).

11.5 USER INSTRUCTIONS

◀1	11. A LASER EQUATION			2▶
P	R	I	AREA	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	STO THE 9 PARAMETERS IN REGISTERS 0-8. (SEE REGISTER STORAGE AT 11.6.)			
2	PRESS E			
3	KEY P, PRESS A, KEY R, PRESS B. PRESS C TO GET I.			
4	KEY P, PRESS A, KEY I, PRESS C. PRESS A TO GET P.			
5	KEY P, PRESS A, KEY I, PRESS C. PRESS B TO GET R			
6	PRESS D TO GET ILLUMINATED AREA			

11.6 A LASER EQUATION

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	001 *LBLE	21 15	
	002 RCL5	36 05	
	003 RCL4	36 04	
	004 ×	-35	
	005 RCL6	36 06	
	006 ÷	-24	
	007 .	-62	
	008 9	09	
	009 ×	-35	
010	010 X²	53	
	011 RCL7	36 07	
	012 4	04	
	013 ×	-35	
	014 +	-55	
	015 Pi	16-24	π
	016 ×	-35	
	017 EEX	-23	
	018 2	02	
	019 CHS	-22	10^{-2}
020	020 ×	-35	
	021 STOE	35 15	SPOT AREA, R = 1
	022 RCL0	36 00	
	023 RCL1	36 01	
	024 ×	-35	bK
	025 RCL8	36 08	
	026 COS	42	cos θ
	027 ×	-35	
	028 RCL3	36 03	k_1
	029 ÷	-24	
030	030 RCLE	36 15	
	031 ÷	-24	
	032 STO9	35 09	L (3)
	033 RTN	24	
	034 *LBLA	21 11	
	035 F3?	16 23 03	
	036 GTO1	22 01	
	037 RCLB	36 12	R
	038 RCL2	36 02	α
	039 ×	-35	
040	040 e^x	33	exp (αR)
	041 RCLB	36 12	
	042 X²	53	
	043 ×	-35	
	044 RCLC	36 13	I
	045 ×	-35	
	046 RCL9	36 09	L
	047 ÷	-24	P (1)
	048 RTN	24	
	049 *LBLC	21 13	
050	050 F3?	16 23 03	
	051 GTO3	22 03	
	052 RCLB	36 12	
	053 RCL2	36 02	
	054 ×	-35	
	055 CHS	-22	
	056 e^x	33	exp (-αR)
	057 RCLA	36 11	
	058 ×	-35	
	059 RCL9	36 09	L
060	060 ×	-35	
	061 RCLB	36 12	
	062 X²	53	R^2
	063 ÷	-24	I (1)
	064 RTN	24	
	065 *LBLB	21 12	
	066 F3?	16 23 03	
	067 GTO2	22 02	
	068 RCL9	36 09	L
	069 RCLA	36 11	
070	070 ×	-35	
	071 RCLC	36 13	
	072 ÷	-24	
	073 √X	54	
	074 STOI	35 46	h STI $\sqrt{A}$
	075 RCL2	36 02	
	076 ×	-35	
	077 2	02	
	078 ÷	-24	
	079 CHS	-22	
080	080 e^x	33	
	081 RCLI	36 46	
	082 ×	-35	
	083 STOB	35 12	R_o (7)
	084 *LBL0	21 00	
	085 RCL9	36 09	
	086 RCLA	36 11	
	087 ×	-35	
	088 RCLB	36 12	
	089 X²	53	
090	090 ÷	-24	
	091 RCLB	36 12	
	092 RCL2	36 02	
	093 ×	-35	
	094 e^x	33	
	095 ÷	-24	
	096 STOI	35 46	LP exp $(-\alpha R_i)/R_i^2$
	097 RCLC	36 13	
	098 −	-45	
	099 RCLI	36 46	
100	100 ÷	-24	
	101 RCL2	36 02	
	102 2	02	
	103 RCLB	36 12	
	104 ÷	-24	
	105 +	-55	
	106 ÷	-24	
	107 RCLB	36 12	
	108 +	-55	
	109 STOD	35 14	R_{i+1} (6)
110	110 RCLB	36 12	
	111 −	-45	
	112 DSP2	-63 02	$R_{i+1}-R_i$

REGISTERS

0	1	2	3	4	5	6	7	8	9
b	K	α	k_1	k_2	λ	D	σ^2	θ	L
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I
P	R/R_i	I	R_{i+1}	SPOT AREA	$\sqrt{A}$

11.6 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
	113 RND	16 24	
	114 X=0?	16-43	CHECK FOR 2 PLACE ACCURACY
	115 GTO4	22 04	
	116 RCLD	36 14	
	117 STOB	35 12	MOVE R_{i+1}
	118 PSE	16 51	SHOW CURRENT VALUE
	119 GTO0	22 00	LOOP
120	120 *LBL1	21 01	
	121 STOA	35 11	P
	122 RTN	24	
	123 *LBL2	21 02	
	124 STOB	35 12	R
	125 RTN	24	
	126 *LBL3	21 03	
	127 STOC	35 13	I
	128 RTN	24	
	129 *LBLD	21 14	
130	130 RCLE	36 15	
	131 RCL8	36 08	
	132 COS	42	cos θ
	133 ÷	-24	
	134 RCLB	36 12	R
	135 X^2	53	
	136 ×	-35	ILLUM. AREA (8)
	137 RTN	24	
	138 *LBL4	21 04	
	139 RCLB	36 12	FINAL R
140	140 RTN	24	
150			
160			
170			
180			
190			
200			
210			
220			

LABELS

A	B	C	D	E
P	R	I	AREA	L (3)
a	b	c	d	e
0	1 STO P	2 STO R	3 STO I	4
5	6	7	8	9

FLAGS: 0, 1, 2, 3

SET STATUS

FLAGS (ON OFF)	TRIG	DISP
0 ☐ ☐	DEG ☐	FIX ☐
1 ☐ ☐	GRAD ☐	SCI ☐
2 ☐ ☐	RAD ☐	ENG ☐
3 ☐ ☐		n____

12. SHAKING THE DICE (A WAR GAMING EXAMPLE)

12.1. REFERENCE

a. E. W. Paxson, *Partially Discriminatory Mortar Fire*, The Rand Corporation, P-5807, February 1977.

12.2. DISCUSSION

In manual war gaming, people instead of computers make the tactical decisions depending on the situation. Yet there are recurring events, such as firefights or air intercepts, whose outcomes must be assessed systematically on the basis of agreed-upon rules and planning factors. The assessment consists of taking a sample from a probability distribution function for outcomes, since the game cannot move on without a definite result. War gamers call this "shaking the dice." It will not do to say, "The probability is 0.4 that at least three tanks of a company of 10 will be destroyed." The game demands a statement such as, "In this firefight 3 tanks were immobilized and one was set afire."

In any game, there are usually large numbers of recurring events of the same type. One expects that the results for this set of events will average out--giving the mean behavior of the model underlying the event type. To clarify this statement, the simple fundamental principle of the Monte Carlo method (sampling from a probability distribution) is invoked. For example, suppose an event can have four outcomes O_1, O_2, O_3, O_4 with respective probabilities p_1, p_2, p_3, p_4, where $p_1 + p_2 + p_3 + p_4 = 1$. Put $P_1 = p_1$, $P_2 = p_1 + p_2$, $P_3 = p_1 + p_2 + p_3$, $P_4 = p_1 + p_2 + p_3 + p_4 = 1$. Take a large number M of random numbers uniformly distributed over the interval (0, 1). By the Law of Large Numbers, one expects, then, that of the M random numbers,

p_1M will be in the interval 0 to P_1,

p_2M will be in the interval P_1 to P_2,

p_3M will be in the interval P_2 to P_3,

p_4M will be in the interval P_3 to 1.

This section shows by an example how hand calculator programs can provide assessments of this nature to speed up the play of manual war games.

A game (or series of games) is set up to test the behavior in a full tactical environment of an innovative weapon system, which is to destroy enemy armor as part of a combined arms force.

The proposed system envisages a new type of mortar round that has a heat-seeking sensor head controlling the maneuver of the round to a target during the steep terminal phase of the trajectory. These rounds are ripple-fired at a set of armored targets, picking targets at random. The sensor head will reject targets previously set afire or exploded (K-kills) to avoid the moth-and-flame effect. But previously immobilized vehicles (M-kills) may wastefully absorb rounds that home on the still-warm engines. Such overkill is a common battlefield event.

If at any time t during the engagement there are j vehicles still moving and k vehicles either still moving or immobilized by previous fire, then the probability P(j, k, t) of the state (j, k) at time t can be determined analytically (see Ref. a). But the analysis is complex, as are the resulting formulas for P(j, k, t). For use in war gaming, it would still be necessary to sample with respect to P(j, k, t). It is much more economical to adopt the procedure of the following subsection.

12.3. EQUATIONS

Let r be the probability that a round gets an M-kill, immobilizing the target. Let s be the probability of a K-kill, with the target exploded or set afire. Initially, there are A moving targets, against which N rounds can be fired during the target exposure interval.

Then the state changes per round, with their associated probabilities, are shown in the following table, remembering that of k targets one is picked at random:

State Change	Probability	Reason
$(j, k) \rightarrow (j, k)$	$p_1 = 1 - s - rj/k$	No K-kill *and* no M-kill of a moving target
$(j, k) \rightarrow (j - 1, k)$	$p_2 = rj/k$	M-kill of a moving target, k does not change
$(j, k) \rightarrow (j - 1, k - 1)$	$p_3 = sj/k$	K-kill of a moving target, k reduces by 1
$(j, k) \rightarrow (j, k - 1)$	$p_4 = s(1 - j/k)$	K-kill of an immobilized target

The corresponding bounds in the interval (0, 1) are $P_1 = 1 - s - rj/k$, $P_2 = 1 - s$, $P_3 = 1 - s + sj/k$, $P_4 = 1$.

Pseudorandom numbers over the interval (0, 1) will be generated by the multiplicative linear congruential method. Let R_o be the initial "seed," a seven-digit decimal fraction. Let m be the multiplier, and let R_i be the *ith* pseudorandom number generated. Then

$$R_{i+1} = \textit{fractional part of } (mR_i) .$$

In the Hewlett-Packard Stat Pac 1 (section 04), the values $R_o = 0.5284163$ and $m = 997$ are used. With these choices, the number of *different* random numbers before returning to R_o (the period of the sequence) is 500,000. The sequence passes the standard tests for randomness.*

12.4. PROGRAM NOTES

Figure 12.1 provides the logical flow of the programming.

It is essential to save the last random number, to be used as the "seed" for the next evaluation.

---

*Other "good" pairs are: $R_o = 0.1111111$, $m = 291$; $R_o = 0.7742713$, $m = 997$.

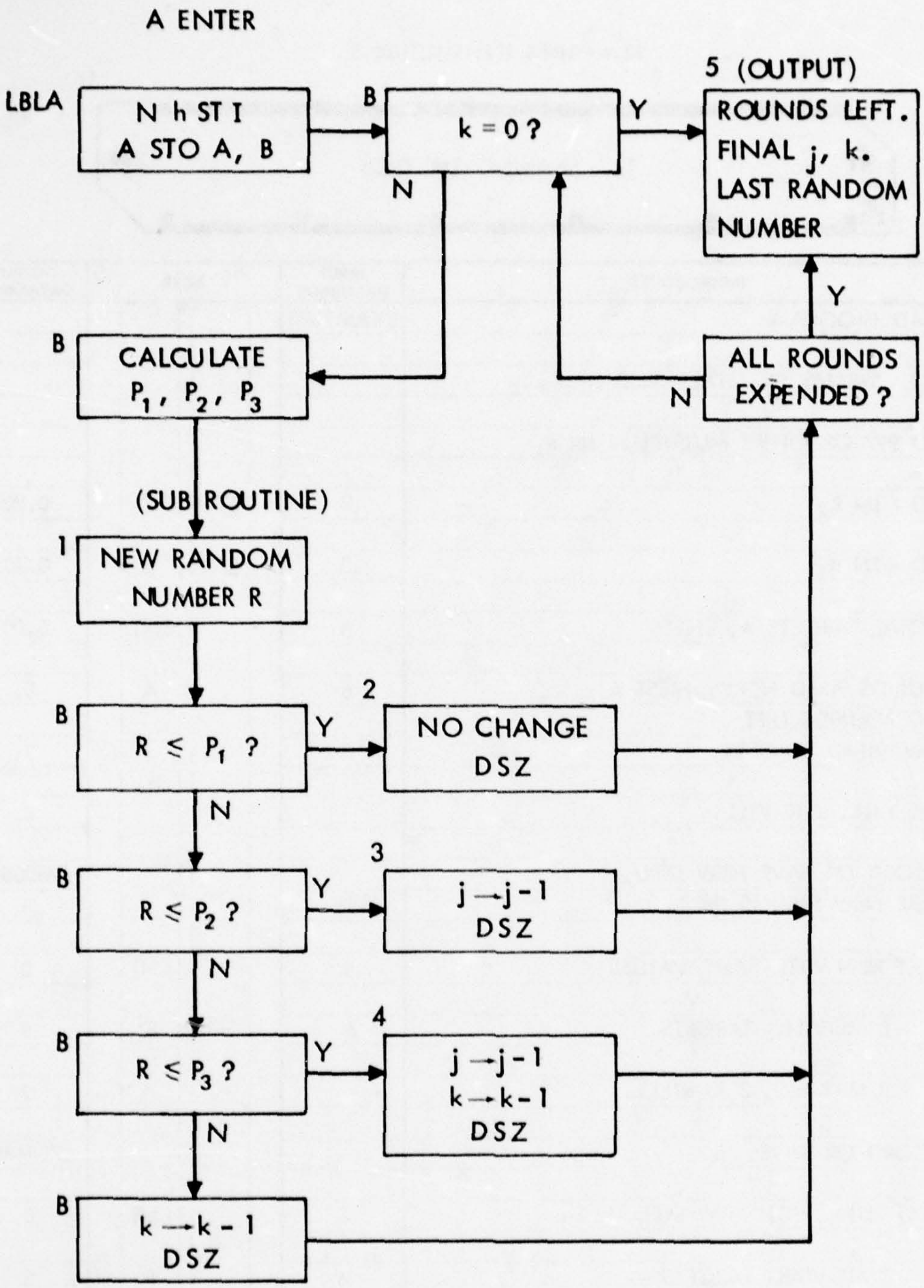


Fig. 12.1. — Program flowchart

12.6 USER INSTRUCTIONS

12. SHAKING THE DICE

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	LOAD PROGRAM	EXAMPLES			
2	STO .5284163 OR OTHER SEED IN R_0				
3	STO 997 OR OTHER MULTIPLIER IN R_1				
4	STO r IN R_2	.2	STO	2	0.20
5	STO s IN R_3	.3	STO	3	0.30
6	INITIAL TARGETS A, ENTER	5		ENT	5.00
7	ROUNDS FIRED N, KEY, PRESS A	6		A	0.
	(NO ROUNDS LEFT .)				
	2 MOVING TARGETS,				2.
	1 M-KILL, 2 K-KILLS				3.
	RECORD OR SAVE NEW SEED.				.7500827
	(HERE NEW SEED IS IN R_0.)				
	NEXT RUN WITH SAME VALUES	5		ENT	0
	0 MOVING TARGETS	6		A	0
	2 M-KILLS, 3 K-KILLS				2
	RECORD OR SAVE.				.7336883
	NEXT RUN, WITH NEW SEED IN R_0	5		ENT	0
	3 MOVING TARGETS	6		A	3
	NO M-KILLS, 2 K-KILLS.				3
	RECORD.				.8173707

12.6 SHAKING THE DICE

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	INITIALIZE
002	STOI	35 46	
003	X⇄Y	-41	
004	STOA	35 11	
005	STOB	35 12	
006	*LBLB	21 12	
007	RCLB	36 12	
008	X=0?	16-43	
009	GTO5	22 05	
010	1	01	
011	RCLA	36 11	
012	RCLB	36 12	
013	÷	-24	
014	STOC	35 13	
015	RCL2	36 02	
016	×	-35	
017	-	-45	
018	RCL3	36 03	
019	-	-45	
020	STO4	35 04	P_1
021	1	01	
022	RCL3	36 03	
023	-	-45	
024	STO5	35 05	P_2
025	RCLC	36 13	
026	RCL3	36 03	
027	×	-35	
028	+	-55	
029	STO6	35 06	P_3
030	RCL4	36 04	
031	GSB1	23 01	NEXT RANDOM NR.
032	X≤Y?	16-35	
033	GTO2	22 02	TESTS FOR
034	RCL5	36 05	
035	RCL0	36 00	STATE CHANGES
036	X≤Y?	16-35	
037	GTO3	22 03	
038	RCL6	36 06	
039	RCL0	36 00	
040	X≤Y?	16-35	
041	GTO4	22 04	
042	RCLB	36 12	
043	1	01	
044	-	-45	
045	STOB	35 12	
046	DSZI	16 25 46	
047	GTOB	22 12	SKIP ON ZERO
048	GTO5	22 05	
049	*LBL1	21 01	
050	RCL0	36 00	
051	RCL1	36 01	
052	×	-35	
053	FRC	16 44	
054	STO0	35 00	NEXT R IN R_o
055	RTN	24	
056	*LBL2	21 02	
057	DSZI	16 25 46	NO CHANGE
058	GTOB	22 12	
059	GTO5	22 05	
060	*LBL3	21 03	
061	RCLA	36 11	
062	1	01	
063	-	-45	
064	STOA	35 11	j - 1
065	DSZI	16 25 46	
066	GTOB	22 12	
067	GTO5	22 05	
068	*LBL4	21 04	
069	RCLA	36 11	
070	1	01	
071	-	-45	
072	STOA	35 11	j - 1
073	RCLB	36 12	
074	1	01	
075	-	-45	
076	STOB	35 12	k - 1
077	DSZI	16 25 46	
078	GTOB	22 12	
079	*LBL5	21 05	
080	DSP0	-63 00	
081	RCLI	36 46	
082	PRTX	-14	ROUNDS LEFT
083	RCLA	36 11	
084	PRTX	-14	FINAL j
085	RCLB	36 12	
086	PRTX	-14	FINAL k
087	DSP7	-63 07	
088	RCL0	36 00	NEW SEED
089	RTN	24	

REGISTERS

0	1	2	3	4	5	6	7	8	9
.5284163	997	r	s	P_1	P_2	P_3			
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I
j	k	j/k			

13. OPTIMUM ALLOCATION OF RESOURCES

13.1. REFERENCES

a. Hanan Luss and Shiv K. Gupta, "Allocation of Effort Resources Among Competing Activities," *Operations Research*, Vol. 23, No. 2, March-April 1975.

b. A. Charnes and W. W. Cooper, "The Theory of Search: Optimum Distributions of Search Effort," *Management Science*, Vol. 5, 1958.

13.2. DISCUSSION

This is a topic in nonlinear (convex) programming. The bibliography of Ref. b indicates optimum search as one military motivation for the study of programming of this nature. Another military application asks for the optimum allocation of weapons to a target system organized into classes of targets of given number and value for which the weapons' kill probabilities differ. Civilian applications arise in allocating advertising budgets, portfolio selection, and budgeting (Ref. a).

The problem is formulated as follows. Let B be the available resources to be allocated to N activities in the amounts $x_1, x_2, \ldots, x_N$, where

$$x_1 + x_2 + \ldots + x_N = B , \qquad x_i \geq a_i \geq 0 .^* \tag{1}$$

If x_i is applied to activity i, the return on the investment is $Q_i(x_i)$, which is a differentiable and strictly concave increasing function. For example, in the weapon allocation application, if there are T_i targets in class i all of value V_i, and if the SSPK is P_i, then

$$Q_i(x_i) = T_i V_i \left[1 - (1 - P_i)^{x_i/T_i}\right] , \tag{2}$$

---

*Higher authority or other considerations may dictate that some activities be assigned minimum (nonzero) resources.

since on the average x_i/T_i weapons are applied to each target and the square brackets contain the kill probability per target. Figure 13.1 shows $[1 - (1 - 0.5)^{x/20}]$. Note the rapid decrease in marginal returns for the larger values of x. One "standard" form for $Q_i(x_i)$ is

$$Q_i(x_i) = S_i[1 - \exp(-m_i x_i)] , \tag{3}$$

which yields (2) if $S_i = T_i V_i$ and $m_i = -(1/T_i)\ell n\,(1 - P_i)$.

We now want to maximize

$$R = Q_1(x_1) + Q_2(x_2) + \dots + Q_N(x_N) \tag{4}$$

subject to (1).

13.3. EQUATIONS

From (3),

$$\partial Q_i(x_i)/\partial x_i = S_i m_i \exp(-m_i x_i) . \tag{5}$$

Reindex the activities so that $S_i m_i \geq S_{i+1}\, m_{i+1}$, which are the marginal returns at the origin. Introduce the Lagrangian multipliers M_ℓ and maximize

$$\sum_{i=1}^{\ell} \left\{ Q_i(x_i) - M_\ell \cdot \left(\sum_{i=1}^{\ell} x_i - B \right) \right\} .$$

To this end

$$\frac{\partial Q_i(x_i)}{\partial x_i} = M_\ell = \frac{\partial Q_j(x_j)}{\partial x_j} , \qquad i, j = 1, \dots, \ell$$

express the marginal equilibrium. We find that the successive M_ℓ are connected by

$$M_{\ell+1} ** \sum_{1}^{\ell+1} 1/m_i = (S_{\ell+1}\ m_{\ell+1})**(1/m_{\ell+1}) \cdot M_\ell ** \sum_{1}^{\ell} 1/m_i \ , \tag{6}$$

where "**" means "to the power" and $M_1 = S_1 m_1 \exp(-Bm_i)$.

The solution algorithm is to stop at that ℓ for which

$$S_{\ell+1}\ m_{\ell+1} \leq M_{\ell+1} \ . \tag{7}$$

Then

$$x_i^* = \frac{1}{m_i} \ln \frac{S_i m_i}{M_\ell} \ , \qquad i = 1, \ldots, \ell \tag{8}$$

and the maximum return is

$$R_\ell^* = \sum_{i=1}^{\ell} \frac{S_i m_i - M_\ell}{m_i} \ . \tag{9}$$

Figure 13.2 shows the physical realization of this algorithm. The Q that gives the maximum return is used first, then the Q that yields the next-highest return, and so on. Find the x_i^* where the slopes of the Q-curves are all equal, since taking a unit of resources from one pocket and putting it into another (moving away from the marginal equilibrium) decreases overall returns. Clearly, stop when even the marginal return at the origin of an activity cannot contribute as much as its more lucrative fellows.

It is characteristic of such solutions that some activities receive no effort. For example, in some air defense problems, targets of very low value get no defense allocation.

A considered or experienced guess can frequently come within 5 percent of the calculated optimum. This is again typical of these problems. But (1) you never know how close you are to the optimum, (2) the improvement being interest on an investment, making many investments adds up, and (3) if, for reasons external to the model, certain allocations to certain categories are specified, you will know what penalty is paid.

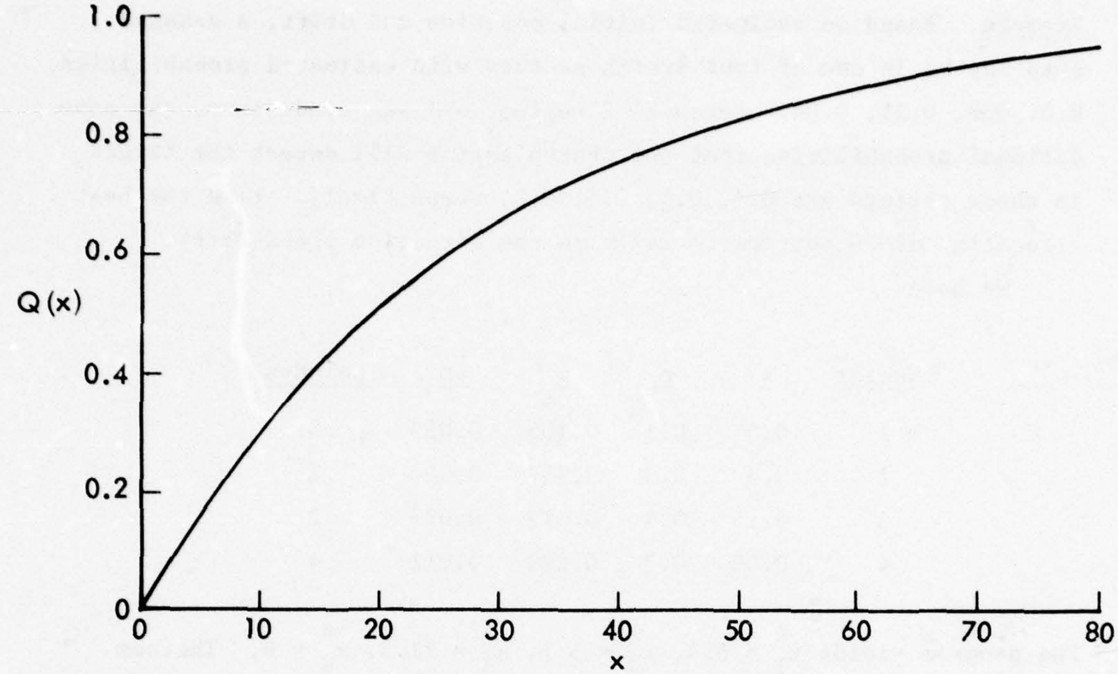


Fig. 13.1 — A return function

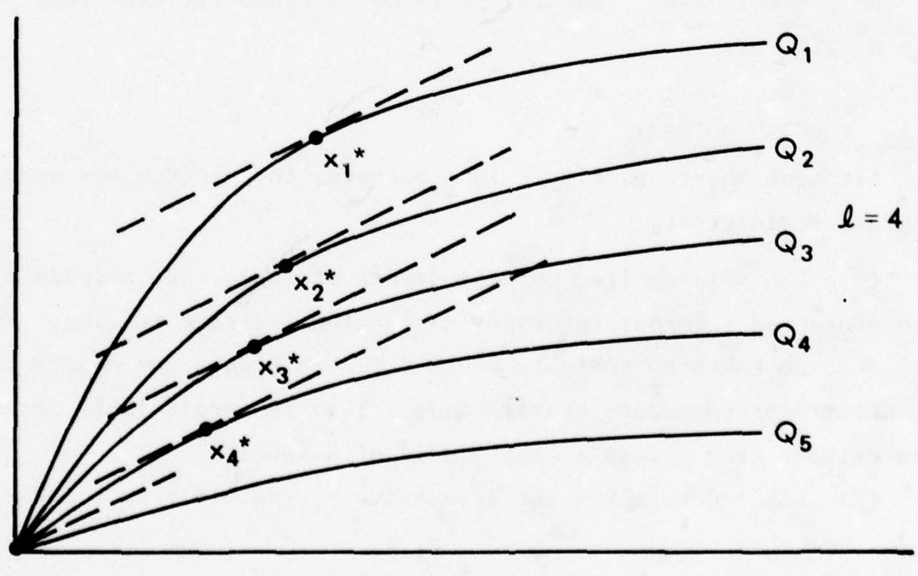


Fig. 13.2 — Solution algorithm

Example. Based on estimated initial position and drift, a damaged ship may be in one of four search sectors with estimated probabilities 0.5, 0.3, 0.15, 0.05. Because of weather and sea conditions, the conditional probabilities that one search sortie will detect the target in these sectors are 0.1, 0.3, 0.4, 0.2, respectively. Find the best allocation of 20 sorties to maximize the detection probability.

We have

Sector	s	p	m	sm	Reindex
1	0.5	0.1	0.105	0.053	3
2	0.3	0.3	0.357	0.107	1
3	0.15	0.4	0.511	0.077	2
4	0.05	0.2	0.223	0.011	4

The program yields $x_1^* = 5.4$, $x_2^* = 3.1$, $x_3^* = 11.5$, $x_4^* = 0$. The sum is 20 as it must be. The detection probability is 0.73. Figure 13.3 shows the Q functions and the solution points and illustrates the algorithm. But it indicates also that the schematic of Fig. 13.2 is not the general case. That is, it is not necessarily true that $x_1^* \geq x_2^* \geq x_3^* \ldots$.

13.4. PROGRAM NOTES

Although short and simple in structure, the program has several elements of interest.

(1) LBL A loads $1/m_i$ using indirect storing. LBL B loads $S_i m_i$ into protected secondary storage, also using indirect storing. But care must be taken to indicate whether f P ↔ S puts the memory in the primary or secondary storage mode. This is particularly important when exiting from a loop in the middle of a label.

(2) LBL 1 determines the successive x_i^* (Eq. (8)) by f ISZ up to ℓ.

(3) LBL 2 then determines R^* by forming the sum (Eq. (9)) in reverse order from ℓ down to 1. f DSZ provides the exit by skipping on zero.

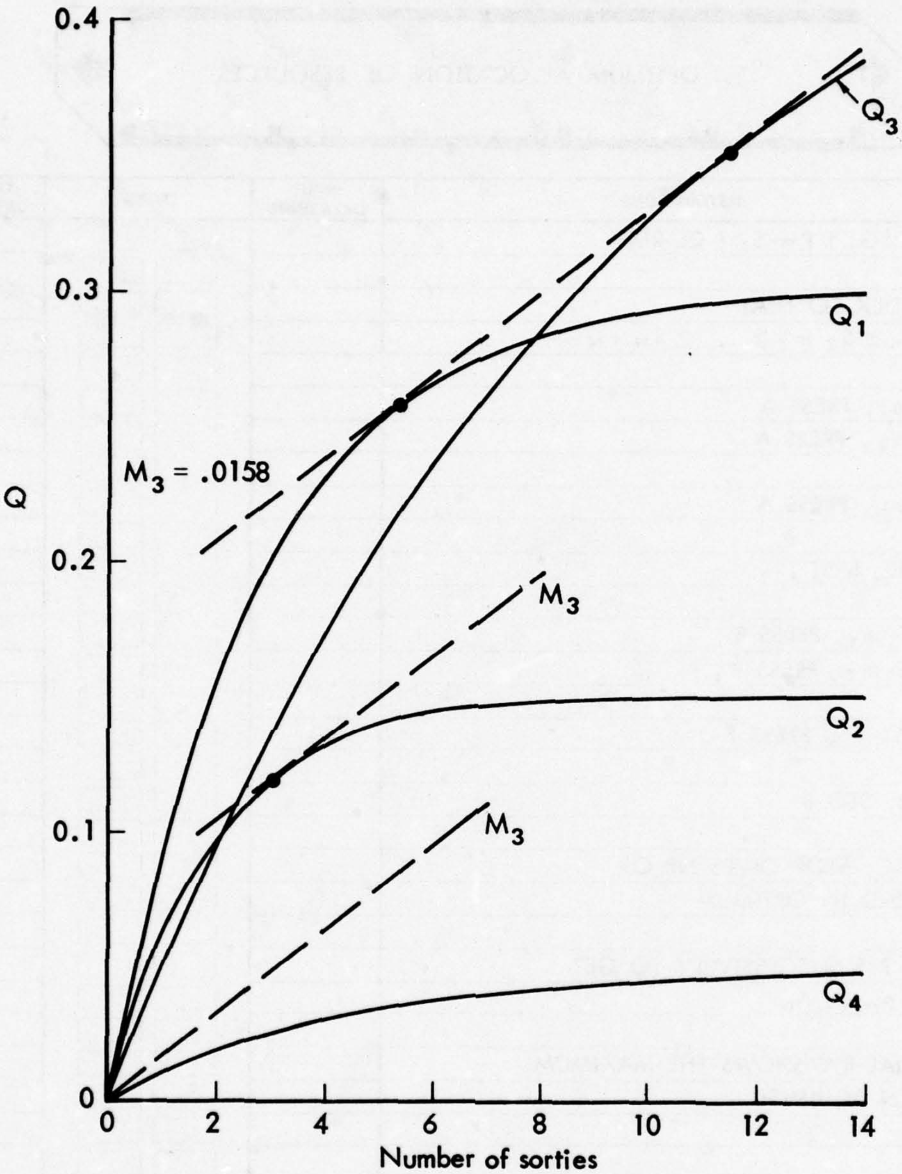


Fig. 13.3—A search problem

13.5 USER INSTRUCTIONS

13. OPTIMUM ALLOCATION OF RESOURCES

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	f CL REG, f P↔S, f CL REG			
2	RE-INDEX SO THAT $S_1 m_1 \geq S_2 m_2 \geq \ldots \geq S_N m_N$			
3	KEY m_1, PRESS A KEY m_2, PRESS A . . . KEY m_N, PRESS A			
4	KEY 10, h ST I			
5	KEY $S_1 m_1$, PRESS B KEY $S_2 m_2$, PRESS B . . . KEY $S_N m_N$ PRESS B			
6	KEY B, STO B			
7	PRESS C, STOP GIVES NR OR X_i USED IN OPTIMUM			
8	PRESS R/S SUCCESSIVELY TO GET $X_1, X_2, \ldots, X_N$			
9	A FINAL R/S SHOWS THE MAXIMUM RETURN ACHIEVED			
	(A SECOND EXAMPLE IS ON NEXT PAGE)			

13.6 OPTIMUM ALLOCATION OF RESOURCES

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	INDIRECT STORAGE
002	1/X	52	OF $1/m_i$
003	ISZI	16 26 46	
004	STOi	35 45	STO (i)
005	RTN	24	
006	*LBLB	21 12	SINCE 10 IN R_I,
007	ISZI	16 26 46	INDIRECT STORAGE
008	STOi	35 45	OF $S_i m_i$ IN 11, 12,...
009	RTN	24	
010	*LBLC	21 13	
011	1	01	
012	STOI	35 46	h ST I (RE-INDEX)
013	RCL1	36 01	
014	STOD	35 14	$1/m_1$
015	1/X	52	
016	RCLB	36 12	
017	×	-35	
018	CHS	-22	$-Bm_1$
019	e^x	33	
020	P⇄S	16-51	SEC
021	RCL1	36 01	$S_1 M_1$
022	×	-35	
023	STOA	35 11	M_1 (6)
024	GTOD	22 14	
025	*LBLD	21 14	
026	ISZI	16 26 46	
027	RCLA	36 11	
028	RCLi	36 45	
029	X≤Y?	16-35	$S_{\ell+1} m_{\ell+1} \leq M_{\ell+1}$ (7)
030	GTOE	22 15	
031	P⇄S	16-51	PRI
032	RCLi	36 45	$\sum_1^i$
033	RCLD	36 14	
034	+	-55	
035	STOE	35 15	NEW PARTIAL SUM
036	RCLA	36 11	
037	RCLD	36 14	
038	RCLE	36 15	
039	÷	-24	
040	y^x	31	$M_i ** \sum_1^i / \sum_1^{i+1}$
041	RCLi	36 45	
042	RCLE	36 15	
043	÷	-24	$(1/m_{i+1}) / \sum_1^{i+1}$
044	P⇄S	16-51	SEC
045	RCLi	36 45	
046	X⇄Y	-41	
047	y^x	31	
048	×	-35	
049	STOA	35 11	M_{i+1} (6)
050	RCLE	36 15	SWITCH
051	STOD	35 14	
052	GTOD	22 14	LOOP
053	*LBLE	21 15	
054	RCLI	36 46	$\ell + 1$
055	1	01	
056	−	-45	ℓ
057	STOC	35 13	
058	R/S	51	
059	0	00	
060	STOI	35 46	
061	GTO1	22 01	
062	*LBL1	21 01	LEFT 030 IN SEC
063	ISZI	16 26 46	
064	RCLi	36 45	$S_i m_i$
065	RCLA	36 11	
066	÷	-24	
067	LN	32	
068	P⇄S	16-51	PRI
069	RCLi	36 45	RCL(i), $1/m_i$
070	×	-35	X_i^* (8)
071	R/S	51	SEC
072	P⇄S	16-51	
073	RCLI	36 46	
074	RCLC	36 13	ℓ
075	X=Y?	16-33	FINISHED ?
076	GTO2	22 02	
077	GTO1	22 01	LOOP
078	*LBL2	21 02	LEFT 076 IN SEC
079	RCLi	36 45	RCL(i), $S_i m_i$
080	RCLA	36 11	
081	−	-45	
082	P⇄S	16-51	PRI
083	RCLi	36 45	$1/m_i$
084	×	-35	
085	ST+0	35-55 00	(9)
086	P⇄S	16-51	SEC
087	DSZI	16 25 46	
088	GTO2	22 02	LOOP FOR SUM
089	P⇄S	16-51	PRI
090	RCL0	36 00	
091	RTN	24	

LABELS

A	B	C	D	E
$1/m_i$	$S_i m_i$	M_1	(6), TEST	ℓ
a	b	c	d	e
0	1 X^*_i	2 R^*_ℓ	3	4
5	6	7	8	9

REGISTERS

0	1	2	3	4	5	6	7	8	9
R^*_ℓ	$1/m_1$	—	—	—	—	—	—	—	$1/m_9$

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
	$S_1 m_1$	—	—	—	—	—	—	—	$S_9 m$

A	B	C	D	E	I
M_ℓ	B	ℓ	$\sum_1^{\ell} 1/m_i$	$\sum_1^{\ell+1} 1/m_i$	

<u>EXAMPLE.</u> B = 200 weapons are to be allocated to 150 targets in 5 value classes to get maximum return.

Class	T_i Number	v_i Value	P_i SSPK	S	m	Sm	Re-index
1	10	5	.05	50	.00513	.25647	5
2	20	4	.2	80	.01116	.89257	3
3	30	3	.3	90	.01189	1.07002	1
4	40	2	.4	80	.01277	1.02165	2
5	50	1	.5	50	.01386	.69315	4

The formulas are: $Q_i = S_i\left[1-(1-P_i)^{x_i/T_i}\right]$,

$S_i = T_i\ v_i,\ m_i = -\ln(1-P_i)/T_i$.

3	KEY IN m, PRESS A	.01189		A	84.10
		.01277		A	78.31
		.01116		A	89.61
		.01386		A	72.15
		.00513		A	194.93
4	10 h STI	10	h	STI	10.00
5	KEY IN S, PRESS B	1.07002		B	1.07
		1.02165		B	1.02
		.89257		B	0.89
		.69315		B	0.69
		.25647		B	0.26
6	200 STO B	200	STO	B	200.00
	DSP 0				200.
7	PRESS C			C	4.
	(NO WEAPONS APPLIED TO CLASS 1)				
8	R/S			R/S	65.
	R/S			R/S	57.
	R/S			R/S	53.
	R/S			R/S	25
	(SUM IS 200)				
9	R/S			R/S	140.
	(THE VALUE ACHIEVED IS 140)				

PART III

COST PROGRAMS

14. LOG-LINEAR CUMULATIVE AVERAGE AND UNIT COSTING

14.1. REFERENCES

a. H. E. Boren, Jr. and H. G. Campbell, *Learning Curve Tables*, The Rand Corporation, RM-6191-PR, April 1970 (3 vols.).

b. R. W. Hamming, *Numerical Methods for Scientists and Engineers*, McGraw-Hill, New York, 1962.

14.2. DISCUSSION

Learning curve theory assumes that each time the total quantity of items produced doubles, the cost per item is reduced to a constant percentage of the previous cost. The relationship is given by the power or log-linear relation

$$y = ax^b ,$$

where x is the cumulative production quantity. If y is the average cost of the first x units, we have the cumulative average learning curve. If y is the cost of the *xth* unit, we have the unit learning curve.

For plotting purposes, the midpoint x_m, corresponding to the lot average cost, is to be determined.

14.3. EQUATIONS

If S is the fraction to which cost decreases (the learning curve percentage) when the quantity is doubled, the slope of the learning curve is

$$b = \ell n\, S/\ell n\, 2 . \quad (1)$$

Then, with a the cost of the first unit,

$$y_c = ax^b \quad (2)$$

is the average cost of the first x units, and the total cost for the first x units is

$$T = xy_c = ax^{b+1} . \tag{3}$$

The unit cost at the *xth* unit is

$$y_u = a\left[x^{b+1} - (x - 1)^{b+1}\right] . \tag{4}$$

Let (x_m, y_m) be the midpoint for the first lot of n units. Then $y_m = y_c = an^b$, and using (4), x_m is the solution of the equation

$$x_m^{b+1} - (x_m - 1)^{b+1} - n^b = 0 . \tag{5}$$

The above equations apply to the log-linear cumulative case.

For the log-linear unit curve, the unit cost at the *xth* unit is

$$y_u = ax^b \tag{6}$$

and the cumulative average cost for the first n units is

$$y_c = \frac{a}{n} \sum_{x=1}^{n} x^b , \tag{7}$$

since the total cost is

$$T = a \sum_{x=1}^{n} x^b . \tag{8}$$

The midpoint (x_m, y_m) is determined by

$$y_m = y_c , \qquad x_m = (y_m/a)^{1/b} . \tag{9}$$

The only programming problems are posed by Eqs. (5) and (8).

In applying Newton's method to (5), a good first estimate is needed for x_m. We have

$$f(x) = x^{b+1} - (x - 1)^{b+1} - n^b$$

$$= x^{b+1} - x^{b+1} (1 - 1/x)^{b+1} - n^b ,$$

and using the first two terms in the expansion of $(1 - 1/x)^{b+1}$, the first estimate for x is

$$x_0 = n/(b + 1)^{1/b} , \qquad (10)$$

which is very close for large n.

To get an excellent approximation for the sum in Eqs. (7) and (8), use the Gregory formula (Ref. a, p. 158):

$$\int_0^n f(x)\, dx = \frac{1}{2} (f_0 + f_n) + \sum_{i=1}^{n-1} f_i + \frac{1}{12} (\Delta f_0 - \Delta f_{n-1})$$

$$- \frac{1}{24} (\Delta^2 f_0 + \Delta^2 f_{n-2}) + \ldots .$$

Because x^b is integrable, the formula can be applied in the backward direction to get the sum. Since x^b is steep for small x, we start the sum at x = 4. (Hamming calls this "low cunning.") We have

$$\sum_{x=1}^{n} x^b \doteq 1 + 2^b + 3^b + \int_3^{n+1} x^b\, dx - \frac{1}{2} \left[3^b + (n + 1)^b \right]$$

$$- \frac{1}{12} \left[4^b - 3^b - (n + 1)^b + n^b \right]$$

$$+ \frac{1}{24} \left[5^b - 2 \cdot 4^b + 3^b + (n + 1)^b - 2 \cdot n^b + (n - 1)^b \right] ,$$

yielding the rather inelegant result

$$\sum_{x=1}^{n} x^b \doteq 1 + 2^b + \left[\frac{5}{8} - \frac{3}{b+1}\right] 3^b - \frac{1}{6} 4^b + \frac{1}{24} 5^b$$
$$+ \frac{1}{24} (n-1)^b - \frac{1}{6} n^b + \left[\frac{n+1}{b+1} - \frac{3}{8}\right] (n+1)^b . \qquad (11)$$

14.4. PROGRAM NOTES

1. In finding a first approximation to the root of Eq. (5) by expression (10), a value less than 1 arises for small n (<5). This would lead to an Error signal because of $(x - 1)^{b+1}$. LBL 2 provides a starting value of 1.01 to avoid this.

2. For small values of n (<5), get $\sum x^b$ by manual calculation (not programmed).

3. The programming done here involved experimenting with the relatively uncommon second-order Newton method for the root of f(x) = 0. The formula* is

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left[1 + \frac{f(x_n)\, f''(x_n)}{2(f'(x_n))^2}\right],$$

compared with the first-order method's

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} .$$

Convergence to a given accuracy is somewhat faster (20 percent in running time) but at additional programming cost to get f″(x). The first-order method is used, employing DSP 2, f RND, to get two-decimal-place accuracy.

4. Obviously, a = 1 in programming, since a is simply a multiplier.

---

*There is a typographical error in this formula as given on p. 82 of Ref. a.

14.5 USER INSTRUCTIONS

114. LOG-LINEAR CUMULATIVE AVERAGE AND UNIT COSTING 2

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	KEY IN % AS .XX, PRESS A	.69	A	0.690
2	KEY IN n, PRESS R/S	81	R/S	0.465
3	FOR CUM. AVG., PRESS B. SEE CUM. AVG. COST		B	0.095
4	PRESS R/S. SEE CUM. TOT. COST.		R/S	7.706
5	PRESS R/S. SEE COST OF nTH UNIT		R/S	0.044
6	PRESS C. SEE MIDPOINT X_m		C	19.85
	FOR UNIT COSTING, REPEAT 1 AND 2 ABOVE, DSP 3.			
1		.69		0.690
2		81		0.465
7	PRESS D. SEE COST OF nTH UNIT.		D	0.095
8	PRESS R/S. SEE TOTAL COST.		R/S	15.021
9	PRESS R/S. SEE AVG COST.		R/S	0.185
10	PRESS R/S. SEE MIDPOINT X_m.		R/S	23.281

14.6 LOG-LINEAR COSTING

STEP	KEY ENTRY		KEY CODE	COMMENTS
001	001	*LBLA	21 11	
	002	STOA	35 11	S
	003	R/S	51	
	004	STO2	35 02	n
	005	RCLA	36 11	
	006	LN	32	
	007	2	02	
	008	LN	32	
	009	÷	-24	
010	010	STO0	35 00	b (1)
	011	1	01	
	012	+	-55	
	013	STO1	35 01	b + 1
	014	RTN	24	
	015	*LBLB	21 12	CUM AVG
	016	RCL2	36 02	
	017	RCL0	36 00	
	018	Y^x	31	y_c (2)
	019	R/S	51	
020	020	RCL2	36 02	
	021	RCL1	36 01	
	022	Y^x	31	T (3)
	023	R/S	51	
	024	RCL2	36 02	
	025	1	01	
	026	−	-45	
	027	RCL1	36 01	
	028	Y^x	31	
	029	−	-45	y_u (4)
030	030	RTN	24	
	031	*LBLC	21 13	CUM AVG MID-PT.
	032	DSP2	-63 02	
	033	RCL1	36 01	
	034	RCL0	36 00	
	035	1/X	52	
	036	Y^x	31	
	037	RCL2	36 02	
	038	÷	-24	
	039	1/X	52	x_o (10)
040	040	1	01	
	041	X⇄Y	-41	
	042	X≤Y?	16-35	SEE PRGM NOTE 1
	043	GTO2	22 02	
	044	STOA	35 11	x_o
	045	STOB	35 12	
	046	GTO3	22 03	
	047	*LBL2	21 02	
	048	1	01	
	049	.	-62	
050	050	0	00	
	051	1	01	
	052	STOA	35 11	x_o
	053	STOB	35 12	
	054	GTO3	22 03	
	055	*LBL3	21 03	
	056	RCLB	36 12	$x_i + 1$
	057	STOA	35 11	
	058	GSB0	23 00	
	059	GSB1	23 01	
060	060	÷	-24	
	061	CHS	-22	
	062	RCLA	36 11	
	063	+	-55	
	064	STOB	35 12	x_{i+1}, SEE NOTE 3
	065	RCLA	36 11	
	066	−	-45	
	067	RND	16 24	2 PLACE ACC.
	068	X≠0?	16-42	OK ?
	069	GTO3	22 03	LOOP
070	070	RCLB	36 12	
	071	RTN	24	x_m
	072	*LBL0	21 00	
	073	RCLA	36 11	x_i
	074	RCL1	36 01	
	075	Y^x	31	
	076	RCLA	36 11	
	077	1	01	
	078	−	-45	
	079	RCL1	36 01	
080	080	Y^x	31	
	081	−	-45	
	082	RCL2	36 02	
	083	RCL0	36 00	
	084	Y^x	31	
	085	−	-45	RTNS $f(x_i)$ (5)
	086	RTN	24	
	087	*LBL1	21 01	
	088	RCLA	36 11	x_i
	089	RCL0	36 00	
090	090	Y^x	31	
	091	RCLA	36 11	
	092	1	01	
	093	−	-45	
	094	RCL0	36 00	
	095	Y^x	31	
	096	−	-45	
	097	RCL1	36 01	
	098	×	-35	
	099	RTN	24	RTNS $f'(x_i)$
100	100	*LBLD	21 14	UNIT CURVE
	101	RCL2	36 02	
	102	RCL0	36 00	
	103	Y^x	31	
	104	R/S	51	y_u (6)
	105	RCL2	36 02	
	106	1	01	
	107	+	-55	
	108	RCL1	36 01	
	109	÷	-24	
110	110	3	03	
	111	ENT↑	-21	
	112	8	08	

REGISTERS

0	1	2	3	4	5	6	7	8	9
b	b + 1	n							
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I
S, x_i	x_{i+1}				

14.6 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	÷	-24	3/8
114	−	-45	
115	RCL2	36 02	
116	1	01	
117	+	-55	
118	RCL0	36 00	
119	Y^x	31	
120	×	-35	LAST TERM IN (11)
121	RCL2	36 02	
122	RCL0	36 00	
123	Y^x	31	
124	6	06	
125	÷	-24	$-n^b/6$
126	−	-45	
127	RCL2	36 02	
128	1	01	
129	−	-45	
130	RCL0	36 00	
131	Y^x	31	
132	5	05	
133	RCL0	36 00	
134	Y^x	31	
135	+	-55	
136	2	02	
137	4	04	
138	÷	-24	
139	+	-55	
140	4	04	
141	RCL0	36 00	
142	Y^x	31	
143	6	06	
144	÷	-24	
145	−	-45	
146	5	05	
147	ENT↑	-21	
148	8	08	
149	÷	-24	
150	3	03	
151	RCL1	36 01	
152	÷	-24	
153	−	-45	
154	3	03	
155	RCL0	36 00	
156	Y^x	31	
157	×	-35	
158	+	-55	
159	2	02	
160	RCL0	36 00	
161	Y^x	31	
162	+	-55	
163	1	01	
164	+	-55	
165	R/S	51	$\sum_{x=1}^{n} x^b$ (11)
166	RCL2	36 02	
167	÷	-24	
168	R/S	51	y_c (7)
169	RCL0	36 00	
170	1/X	52	
171	Y^x	31	
172	RTN	24	x_m (9)

LABELS

A b	B CUM AVG	C MID POINT	D UNIT CURVE	E
a	b	c	d	e
0 $f(x_i)$	1 $f'(x_i)$	2 NOTE 1	3 x_m(5)	4
5	6	7	8	9

FLAGS: 0, 1, 2, 3

SET STATUS

FLAGS (ON OFF)	TRIG	DISP
0 ☐ ☐	DEG ☐	FIX ☐
1 ☐ ☐	GRAD ☐	SCI ☐
2 ☐ ☐	RAD ☐	ENG ☐
3 ☐ ☐		n_____

15. TIME-PHASED PROCUREMENT COSTING

15.1. REFERENCES

None.

15.2. DISCUSSION

The costing approach of this section is a hitherto undocumented model owing to H. G. Massey of The Rand Corporation. For definiteness, the model will be explained in terms of aircraft procurement, although it applies equally to the procurement of any system with any number of components. The initial batch of test articles is not priced, it being assumed that they are funded from another account. But the average prices of the batch's components provide the costs of initial articles for learning curve calculations. Overruns and inflation are not included. The model determines, by fiscal year, New Obligational Authority (NOA) required to meet the production/delivery schedule.

15.3. EQUATIONS

Specify that two required sequences of possessed aircraft (S_n, T_n), n = 1, ..., N, are to be in the fleet at the end of year n, where S_n are squadron or UE aircraft and T_n are training aircraft.

If A and B are flying hours per year per aircraft (FH/Y), for squadron and training aircraft, the *cumulative* fleet flying hours through year n are approximately

$$H_n = \sum_{i=1}^{n} [AS_i + BT_i] . \qquad (1)$$

The *cumulative* fleet attrition is given by

$$a_n = C \cdot H_n^d . \qquad (2)$$

A fraction λ of the fleet is assigned to command support (pipeline). With allowance for these two factors, the *cumulative* number of aircraft to be procured through year n is

$$Q_n = (1 + \lambda) P_n + a_n . \quad (3)$$

If the fleet is to be kept in a steady-state condition (S_N, T_N) for M more years, aircraft will be delivered in year N + 1 to meet the attrition requirements of these M years. Hence H_{N+1} and a_{N+1} will be calculated to determine this final buy.

An aircraft has three major components: engines, airframe, and avionics. There is a cumulative buy program for each component, allowing for lead times and learning curve effects. Cumulative average costing is most convenient (see Sec. 14).

Take engines first. The procurement parameters can be written as the string of numbers

$$(a_1, p_{11}, x_{11}, p_{12}, x_{12}, p_{13}, t_1, \mu_1) ,$$

where the first subscript refers to component 1 (engines) and a_1 is the cost of the first article, p_{11} is the learning curve percentage (or rather fraction) up to article number x_{11}, p_{12} is the learning curve percentage for articles $x_{11} + 1$ up to x_{12}, and p_{13} is the subsequent percentage. Of course, a learning curve may have only one or two segments. The lead time in months is t_1, so that an engine takes t_1 months from start of fabrication to delivery to the fleet as part of a complete aircraft. (Mating of engines and avionics with the airframe is taken to be part of the airframe lead time and cost.) The number of engines required per aircraft is μ_1, allowing for multi-engine models and spares. With these production parameters:

Up to x_{11} the cumulative average total cost for x articles is

$$a_1 x^{b_{11}+1} , \quad (4)$$

where $b_{11} = \ln p_{11} / \ln 2$.

From $x_{11} + 1$ to x_{12}, the cumulative total cost is

$$a_1 \cdot x_{11}^{b_{11}-b_{12}} \cdot x^{b_{12}+1} , \tag{5}$$

and above $x_{12} + 1$,

$$a_1 \cdot x_{11}^{b_{11}-b_{12}} \cdot x_{12}^{b_{12}-b_{13}} \cdot x^{b_{13}+1} . \tag{6}$$

These expressions are verified by putting $x = x_{11}$ in (5) and $x = x_{12}$ in (6). These expressions define a *cumulative* cost function $c_1(x)$ for all articles numbered 1 through x.

Next express t_1 in years and write

$$t_1 = \mathrm{INT}(t_1) + \mathrm{FRAC}(t_1) .$$

Thus a 26-month lead time is 2.167 = 2 + 0.167. Then fabrication starts at $-t_1$, but during the fiscal year $-\mathrm{INT}(t_1)$, only $\mu Q_1\ \mathrm{FRAC}(t_1)$ articles will be started, requiring NOA of

$$C_1(\mu Q_1\ \mathrm{FRAC}(t_1)) .$$

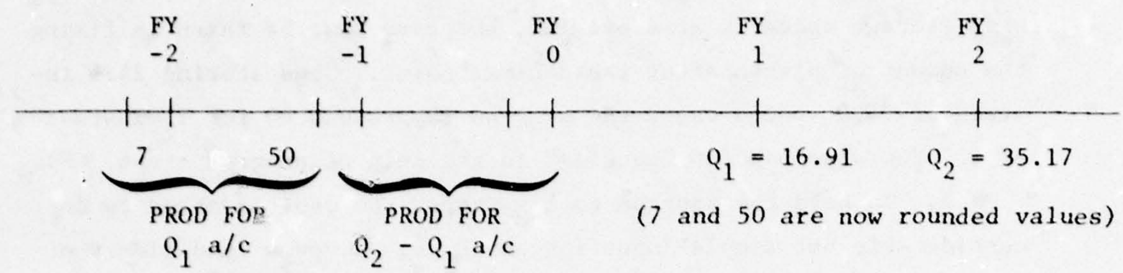
To the end of the next FY, $-\mathrm{INT}(t_1) + 1$, the *cumulative* number of engines started is

$$\mu Q_1 + \mu(Q_2 - Q_1)\ \mathrm{FRAC}(t_1)) , \tag{7}$$

and NOA for that FY is

$$C_1(\mu Q_1 + \mu(Q_2 - Q_1)\ \mathrm{FRAC}(t_1)) . \tag{8}$$

By example, if t_1 = 26 months, Q_1 = 16.91, Q_2 = 35.17,



Note that deliveries to the fleet are made at a uniform rate during a year. So that formula (7) will apply for the first year and the last year, store $Q_0 = 0$ and store Q_{N+1} as Q_{N+2}. Then, for the last FY, we will have the correct cumulative total Q_{N+1}.

The above procedure applies equally to the remaining two components. The program then generates the *cumulative* NOA component by component. The user then adds vertically by FY, and takes differences horizontally to get the final NOA by FY that the procurement program would demand.

The model well illustrates the simplicity resulting when cumulative average costing is used rather than unit costing.

15.4. PROGRAM NOTES

1. LBL A uses indirect addressing and simple looping to produce the cumulative flying hours H_i.

2. LBL B computes required cumulative aircraft Q_i and puts these in the primary storage originally occupied by the squadron and training aircraft of the original schedule.

3. The GTO E of line 074 will lead to the storage of Q_{N+1} in Q_{N+2} also (see above for reason).

4. LBL C calculates the coefficients required for a segmented learning curve.

5. LBL 6 calculates the successive cumulative number of articles produced and NOA required as the fiscal years are incremented.

6. "Packed" storage is used extensively. Thus, $S_2 = 24$ and $T_2 = 6$ are stored in Register 2 as 24.06. Then S_2 is retrieved by 'f INT' and T_2 by 'g FRAC, EEX 2, X'. This storage device is useful

when storage space is at a premium, but care must be taken in fixing the number of places after the decimal point. Thus storing 24.6 instead of 24.06 would cause the program to produce 60 for T_2 instead of 6. You also pay for "packing" in the coin of program steps.

7. To hold the program to 224 steps, the user is asked to do considerable but simple inputting, such as: Given a lead time t of 26 months, key in 26 and then 'ENTER, 12, ÷, STO 6'.

8. For lack of storage space, the user records output as produced and does some final additions and subtractions to get the NOA by fiscal year.

Example. The required buildup schedule is:

End of FY	1	2	3	4	5	6
Squadron acft	12	24	48	84	108	108
Training acft	3	6	12	21	27	27

The *steady-state fleet* is to be reached at the end of year 5. This will be kept constant through year 10. FH/Y = 240 for squadron acft and 720 for training acft. The cumulative attrition coefficient is 0.00015 and the exponent is 1.05. The command support factor is 5 percent. The engine learning curve has three segments, and the parameters for this component are

$$a_1 = 10,\ p_{11} = 0.9,\ x_{11} = 60,\ p_{12} = 0.8,\ x_{12} = 110,$$
$$p_{13} = 0.6,\ t_1 = 26,\ \mu_1 = 2.5\ .$$

For the airframe (two segments):

$$a_2 = 25,\ p_{21} = 0.85,\ x_{21} = 30,\ p_{22} = 0.75,\ t_2 = 20,\ \mu_2 = 1\ .$$

For avionics (one segment):

$$a_3 = 30,\ p_{31} = 0.75,\ t_3 = 14,\ \mu_3 = 1\ .$$

Calculation

FY	1	2	3	4	5	6
Accum. FH	5040	15120	35280	70560	115920	342720
Req. acft	16.91	35.17	71.93	128.75	172.90	238.97

The sixth year allows for five years of flying and attrition.

FY	-2	-1	0	1	2	3	4	5
Cum. engines	7	50	103	204	340	460	597	597
Cum. NOA	52.08	275.88	464.52	571.37	653.54	707.63	757.85	757.85
Cum. airframes	0	11	29	60	110	158	217	239
Cum. NOA	0	156.73	329.20	506.78	722.45	892.89	1075.00	1137.47
Cum. avionics	0	3	20	41	81	136	184	239
Cum. NOA	0	57.05	173.05	263.35	392.21	531.08	633.80	738.58
Total cum. NOA	52.08	489.66	966.77	1341.50	1768.20	2131.60	2466.65	2633.90
FY NOA	52.08	437.58	477.11	374.73	426.70	363.40	335.05	167.25

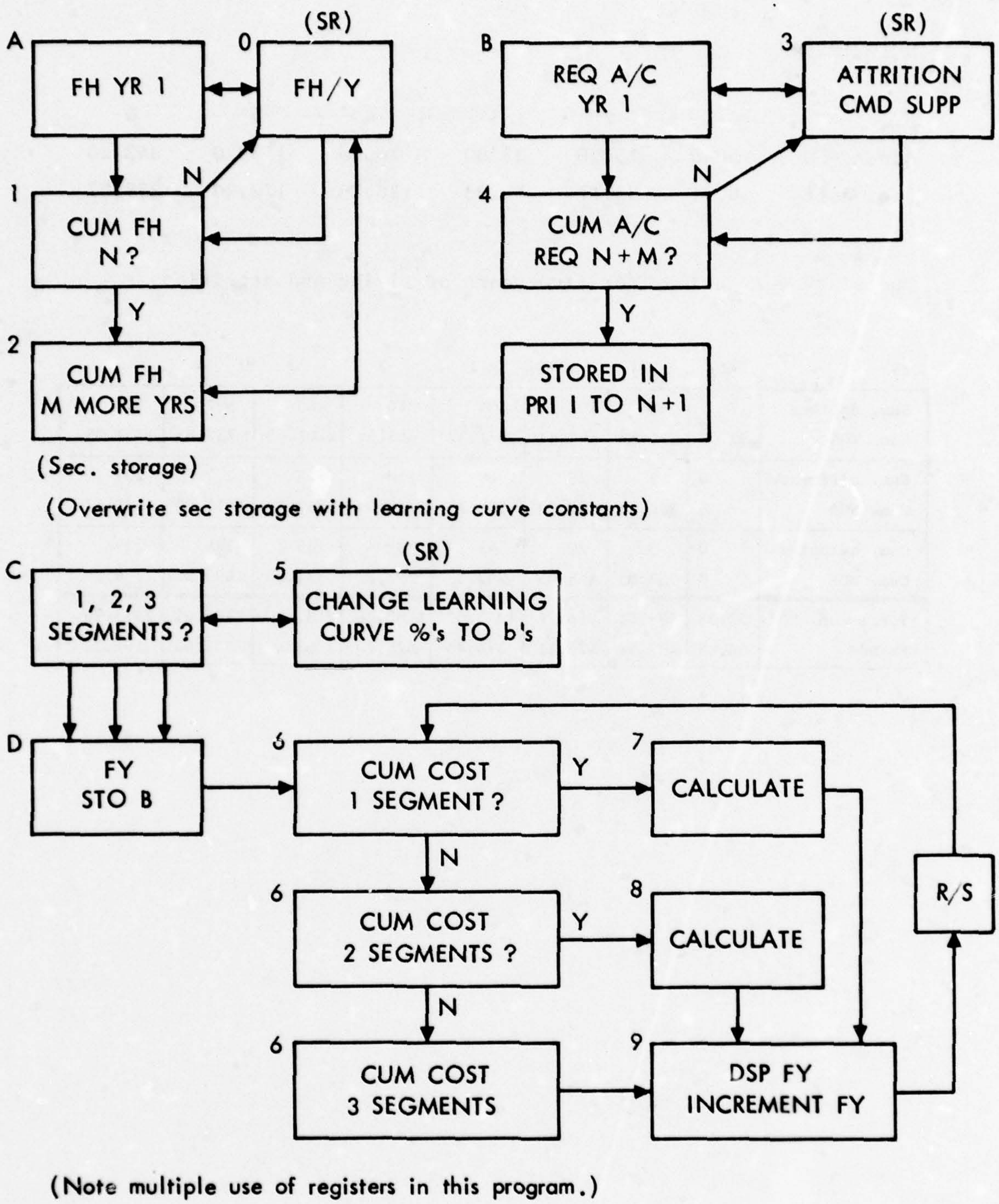


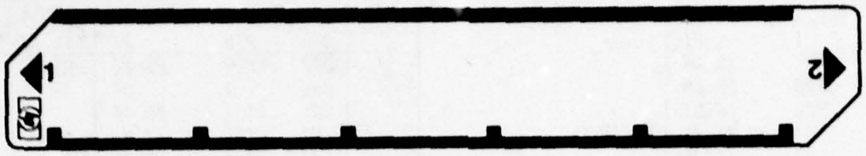
Fig. 15.1—Time phased procurement flowchart

15.5 USER INSTRUCTIONS

15. TIME PHASED PROCUREMENT COSTING

STEP	INSTRUCTIONS EXAMPLE OF 14.4	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	STO N.M IN A. N YEARS TO FINAL				
	BUILDUP, M ADDITIONAL YEARS OF				
	STEADY STATE FLEET OPERATION				
	($N \leq 7$, NOTE 5.05 AND <u>NOT</u> 5.5):	5.05	STO	A	5.05
2	STO $1+\lambda$ IN B, WHERE λ IS THE				
	COMMAND SUPPORT FACTOR	1.05	STO	B	1.05
	STO C IN C AND d IND.	.00015	STO	C	1.5 -04
	ATTRITION FACTORS, EQ (2).	1.05	STO	D	1.05
4	STO A. B IN 0. A IS FH/Y PER	240.720	STO	0	240.72
	SQ. A/C, B IS FH/Y PER TRG A/C.				
	(B HAS 3 DIGITS, INITIAL ZERO IF NEEDED)				
5	STO $S_n \cdot T_n$ IN R_n, $n = 1, \ldots, n$,	12.03	STO	1	12.03
	AND STO $S_N \cdot T_N$ IN R_{N+}, ALSO.	24.06	STO	2	24.06
	S_n IS REQD SQ A/C AT END OF YR n.	48.12	STO	3	48.12
	T_n IS REQD TRG A/C AT END OF YR n.	84.21	STO	4	84.21
		108.27	STO	5	108.27
			STO	6	108.27
6	PRESS A.			A	45360
	TO SEE CUM FH		f	P↔S	45360
			RCL	1	5040
			RCL	2	15120
			RCL	3	35280
			RCL	4	70560
			RCL	5	115920
			RCL	6	342720
7	f P↔S (IMPORTANT)		f	P↔S	342720
8	PRESS B			B	238.97
	TO SEE CUM. A/C REQ (UNROUNDED)		RCL	1	16.91
			RCL	2	35.17
			RCL	3	71.93
			RCL	4	128.75
			RCL	5	172.90
			RCL	6	238.97
9	f P↔S (IMPORTANT). FOR ENGINES:		f	P↔S	238.97
	a_1 STO 0	10	STO	0	10.00
	P_{11} STO 1	.9	STO	1	0.90
	x_{11} STO 2	60	STO	2	60.00
	P_{12} STO 3	.8	STO	3	0.80

15.5 USER INSTRUCTIONS



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
	x_{12} STO 4	110	STO 4	110.00
	P_{13} STO 5	.6	STO 5	0.60
	t_1 (IN MOS) DIVIDE BY 12, STO 6	2.17	STO 6	2.17
	n_1 (NR OF ENGINES/A/C) STO 9	2.5	STO 9	2.50
10	PRESS C. SEE FY.		C	-2.00
	TO GET CUM NR OF ENGINES PRODUCED		RCL D	7.00
	TO GET CUM NOA		RCL E	52.08
	R/S FOR NEXT FY		R/S	-1.00
			RCL D	50.00
			RCL E	275.88
	CONTINUE UNTIL CUM NR DECREASES			
11	RETURN TO 9 FOR NEXT COMPONENT			
	BUT <u>DO NOT USE</u> f P↔S.			
	IF COMPONENT LEARNING CURVE			
	HAS 2 SEGMENTS:			
	EEX 9, STO 4; t/12, STO 6;			
	μ STO 9. NO OTHER STORAGE NEEDED			
	IF ONLY ONE SEGMENT:			
	EEX 9, STO 2; t/12, STO 6;			
	μ STO 9. NO OTHER STORAGE NEEDED.			
12	RECORD NOA FOR EACH COMPONENT			
	BY FY. ADD VERTICALLY. TAKE			
	DIFFERENCES TO GET INCREMENTAL			
	NOA BY FY.			

15.6 TIME PHASED PROCUREMENT

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	001 *LBLA	21 11	
	002 1	01	
	003 STOI	35 46	
	004 GSB0	23 00	
	005 P⇄S	16-51	
	006 STOi	35 45	TEMP
	007 P⇄S	16-51	
	008 GSB1	23 01	(OR GTO 1)
	009 *LBL0	21 00	FH/Y
010	010 RCLi	36 45	INDIRECT
	011 INT	16 34	S_i
	012 RCL0	36 00	
	013 INT	16 34	A
	014 ×	-35	
	015 RCLi	36 45	
	016 FRC	16 44	
	017 EEX	-23	
	018 2	02	100
	019 ×	-35	T_i
020	020 RCL0	36 00	
	021 FRC	16 44	
	022 EEX	-23	
	023 3	03	1000
	024 ×	-35	B
	025 ×	-35	
	026 +	-55	$AS_i + BT_i$
	027 RTN	24	
	028 *LBL1	21 01	
	029 ISZI	16 26 46	
030	030 GSB0	23 00	
	031 DSZI	16 25 46	
	032 P⇄S	16-51	SEC
	033 RCLi	36 45	
	034 +	-55	
	035 ISZI	16 26 46	
	036 STOi	35 45	PARTIAL SUM
	037 P⇄S	16-51	PRI
	038 RCLA	36 11	
	039 INT	16 34	N
040	040 RCLI	36 46	
	041 X=Y?	16-33	FINISHED FOR N ?
	042 GTO2	22 02	
	043 GTO1	22 01	LOOP
	044 *LBL2	21 02	
	045 GSB0	23 00	
	046 RCLA	36 11	
	047 FRC	16 44	
	048 EEX	-23	
	049 2	02	
050	050 ×	-35	$M.\ (AS_N + BT_N)$
	051 ×	-35	
	052 P⇄S	16-51	SEC
	053 RCLi	36 45	H_N (1)
	054 +	-55	
	055 ISZI	16 26 46	
	056 STOi	35 45	TOTAL FH
	057 P⇄S	16-51	BECAUSE GTO 2 IN
	058 RTN	24	042, NO RTN TO 009
	059 *LBLB	21 12	PRI
060	060 1	01	
	061 STOI	35 46	
	062 GSB3	23 03	
	063 STOi	35 45	Q_1
	064 *LBL4	21 04	
	065 ISZI	16 26 46	
	066 GSB3	23 03	
	067 STOi	35 45	Q_n
	068 RCLA	36 11	
	069 INT	16 34	
070	070 1	01	
	071 +	-55	N + 1
	072 RCLI	36 46	
	073 X=Y?	16-33	FINISHED ?
	074 GTOE	22 15	
	075 GTO4	22 04	LOOP
	076 *LBL3	21 03	
	077 RCLi	36 45	
	078 INT	16 34	S_N
	079 RCLi	36 45	
080	080 FRC	16 44	
	081 EEX	-23	
	082 2	02	
	083 ×	-35	T_n
	084 +	-55	
	085 RCLB	36 12	
	086 ×	-35	$(1+\lambda)\ (S_n+T_n)$
	087 P⇄S	16-51	SEC
	088 RCLi	36 45	H_n
	089 RCLD	36 14	
090	090 Y^x	31	
	091 RCLC	36 13	
	092 ×	-35	a_n (2)
	093 +	-55	Q_n (3)
	094 P⇄S	16-51	PRI
	095 RTN	24	RTN TO 063/067
	096 *LBLC	21 13	SEC (USER INST 9)
	097 0	00	
	098 STOI	35 46	
	099 GSB5	23 05	
100	100 EEX	-23	
	101 9	09	
	102 RCL2	36 02	
	103 X=Y?	16-33	1 SEGMENT ?
	104 GTOD	22 14	
	105 RCL1	36 01	
	106 RCL3	36 03	
	107 −	-45	
	108 Y^x	31	
	109 RCL0	36 00	
110	110 ×	-35	
	111 STO7	35 07	
	112 EEX	-23	

REGISTERS

0	1	2	3	4	5	6	7	8	9
A.B	$S_1 . T_1$	$S_2 . T_2$	-	-	-	-	$S_N . T_N$	$S_N . T_N$	

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
a_1	P_{11}	x_{11}	P_{12}	x_{12}	P_{13}	$t_1/12$			μ_1

A	B	C	D	E	I
N.M	$1+\lambda$	C	d		

15.6 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	9	09	
114	RCL4	36 04	
115	X=Y?	16-33	2 SEGMENTS ?
116	GTOD	22 14	
117	RCL3	36 03	
118	RCL5	36 05	
119	−	-45	
120	Y^x	31	
121	RCL7	36 07	
122	x	-35	
123	STO8	35 08	
124	GTOD	22 14	
125	*LBL5	21 05	
126	ISZI	16 26 46	
127	RCLI	36 46	
128	7	07	
129	X=Y?	16-33	ALL CONSTANTS ?
130	RTN	24	
131	RCLi	36 45	P_{1i}
132	LN	32	
133	2	02	
134	LN	32	
135	÷	-24	
136	STOi	35 45	$b_{1i} = \ln P_{1i} / \ln$
137	ISZI	16 26 46	
138	RCLi	36 45	x_{1i}
139	EEX	-23	SEE USER INST. 11
140	9	09	
141	X=Y?	16-33	
142	RTN	24	
143	GTO5	22 05	LOOP FOR SEGMENTS
144	*LBLD	21 14	
145	0	00	
146	STOD	35 14	
147	STOE	35 15	CLEAR
148	STOI	35 46	
149	RCL6	36 06	t/12
150	INT	16 34	
151	CHS	-22	
152	STOB	35 12	FY
153	RCL6	36 06	
154	FRC	16 44	
155	STOC	35 13	
156	P⇄S	16-51	PRI
157	0	00	
158	STO0	35 00	
159	*LBL6	21 06	
160	RCLi	36 45	Q_n
161	RCLi	36 45	(DELETE 161)
162	ISZI	16 26 46	
163	RCLi	36 45	Q_{n+1}
164	−	-45	
165	CHS	-22	
166	RCLC	36 13	FRAC
167	x	-35	
168	+	-55	
169	P⇄S	16-51	
170	RCL9	36 09	μ
171	x	-35	
172	DSP0	-63 00	
173	RND	16 24	
174	STOD	35 14	(8)
175	DSP2	-63 02	
176	RCL2	36 02	x_{11}
177	X⇄Y	-41	
178	X≤Y?	16-35	
179	GTO7	22 07	1st SEGMENT
180	RCL4	36 04	
181	X⇄Y	-41	
182	X≤Y?	16-35	
183	GTO8	22 08	2nd SEGMENT
184	RCL5	36 05	
185	1	01	$b_{13}+1$
186	+	-55	
187	Y^x	31	
188	RCL8	36 08	
189	x	-35	3rd SEGMENT
190	STOE	35 15	
191	GTO9	22 09	
192	*LBL9	21 09	
193	RCLB	36 12	
194	R/S	51	DISPLAY FY (152)
195	RCLB	36 12	
196	1	01	
197	+	-55	
198	STOB	35 12	
199	P⇄S	16-51	
200	GTO6	22 06	
201	*LBL7	21 07	1st SEGMENT
202	RCL1	36 01	
203	1	01	
204	+	-55	
205	Y^x	31	
206	RCL0	36 00	
207	x	-35	
208	STOE	35 15	
209	GTO9	22 09	
210	*LBL8	21 08	2nd SEGMENT
211	RCL3	36 03	
212	1	01	
213	+	-55	
214	Y^x	31	
215	RCL7	36 07	
216	x	-35	
217	STOE	35 15	
218	GTO9	22 09	
219	*LBLE	21 15	
220	R↓	-31	
221	R↓	-31	TO GET Q_{N+1}
222	ISZI	16 26 46	
223	STOi	35 45	
224	RTN	24	Q_{N+1}

LABELS					FLAGS
A	B	C	D	E	0
a	b	c	d	e	1
0	1	2	3	4	2
5	6	7	8	9	3

SET STATUS

FLAGS (ON OFF)	TRIG	DISP
0 ☐ ☐	DEG ☐	FIX ☐
1 ☐ ☐	GRAD ☐	SCI ☐
2 ☐ ☐	RAD ☐	ENG ☐
3 ☐ ☐		n____

16. COST/BENEFIT STREAMS

16.1. REFERENCES

None.

16.2. DISCUSSION

The model of this section is also due to H. G. Massey of The Rand Corporation. It deals with decisions to spend money now as opposed to later during the life cycle of a weapon system. That is, should engineering development monies be spent now with the expectation that future operating and support costs will be lower? The planner must decide, for example, whether to install engine diagnostic equipment now, assuming that future maintenance will otherwise be less efficient and therefore more costly. The model quantifies such decision problems, using as a yardstick the present value of a discounted stream of expenditures and benefits, both of which are expressed in constant dollars (no allowance for inflation).

A simple example will illustrate. Suppose we assume that $10M spent now will lead to operating and support (O&S) costs of $20M 8 years from now; if no money is spent now, these future costs will be $40M. Let the discount rate be 10 percent, as currently mandated by the Department of Defense. If the $10M were invested at 10 percent compounded interest for 8 years, it would yield $21.4M. Consequently, we would save $1.4M by *not* improving the system now. But if the rate were 5 percent, we would *lose* $5.2M by not improving the system now. It follows that the rate mandated or assumed has a controlling impact on the decision.

With respect to estimating the future costs ($20M and $40M above), one other crucial point must be made. Suppose the system in question is a fleet of aircraft of a given type. Then we must keep the operational capability constant in the two cases. That is, the O&S costs must be assessed in both cases to produce the same in-commission rate or other measure of operational capability for the fleet.

16.3. EQUATIONS

Set a time horizon of N future fiscal years, the expected life of the weapon system. Let C_i be the costs of the proposed near-term improvements for years 1, ..., N. Let B_i be the assessed *incremented* O&S *savings* given the stream C_i. Then the discounted present value of the benefit/cost stream at a rate r is

$$P(r) = \sum_{i=1}^{N} (B_i - C_i)(1 + r)^{-i+1} , \tag{1}$$

since by convention present values are stated for fiscal year 1.

The "breakeven" year j is defined as the first j for which

$$\sum_{i=1}^{j} (B_i - C_i)(1 + r)^{-i+1} > 0 , \tag{2}$$

if $P(r) > 0$.

The "internal rate of return" is that r* for which

$$P(r^*) = 0 . \tag{3}$$

If the actual rate r is greater than r*, then $P(r) < 0$, because $(1 + r)^{-i+1} < (1 + r^*)^{-i+1}$. In this case, the near-term improvement investment would not be made.

The value r* is found by Newton's method. We have, using $D_i = B_i - C_i$,

$$\begin{aligned} P'(r) &= \sum_{i=1}^{N} (1 - i) \cdot D_i \cdot (1 + r)^{-i} \\ &= \frac{1}{1 + r}\left[P(r) - \sum_{i=1}^{N} i \cdot D_i \cdot (1 + r)^{-i+1}\right] , \end{aligned} \tag{4}$$

which is written in this form to simplify the programming. Finally

$$1 + r_n = 1 + r_{n-1} - \frac{(1 + r_{n-1}) \cdot P(r_{n-1})}{P(r_{n-1}) - \sum_{i=1}^{N} i \cdot D_i \cdot (1 + r)^{-i+1}} . \quad (5)$$

16.4. PROGRAM NOTES

(1) This program uses indirect addressing in a natural and straightforward fashion. The successive D_i's are stored in registers 1, ..., 19 (19 years is the maximum for this program). Looping by f ISZ is then simple.

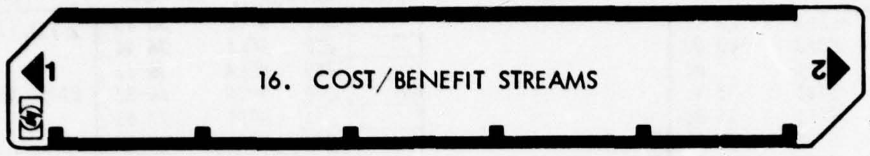
(2) To get the breakeven year, we want to test for a change of sign in

$$P(j, r) = \sum_{i=1}^{j} D_i (1 + r)^{-i+1}$$

as this partial sum crosses the *time* axis. A simple procedure is to test the ratios $P(j + 1, r)/P(j, r)$ to see when, if at all, the ratio is *negative*.

(3) In using Newton's method, a simple first-guess at the root is $r = 0$. But $1 + r$ or 1 is then stored in B. If any iteration produces $1 + r < 1$, stop. In this case there is no internal rate of return.

16.5 USER INSTRUCTIONS



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	STO N IN A	5	STO	A	5.00
2	STO $D_1 = B_1 - C_1$ IN 1	-5	STO	1	-5.00
	D_2,	-3	STO	2	-3.00
	D_3	1	STO	3	1.00
	D_4	4	STO	4	4.00
	D_5	7	STO	5	7.00
3	IF $N > 9$, f P↔S		f	P↔S	
	STO D_{10} IN 0				
	STO D_{11} IN 1, UP TO D				
	f P↔S (IMPORTANT)		f	P↔S	
4	STO 1 + r IN B (r = 10%)	1.1	STO	B	1.10
5	PRESS A. SEE BREAKEVEN YEAR			A	5.000
6	PRESS R/S. SEE P(r)			R/S	0.886
7	TO FIND r*, THE INTERNAL RATE OF RETURN,				
	STORE A FIRST GUESS 1 + r IN B	1.1	STO	B	1.100
8	PRESS B. SEE f -x-.			B	1.136
9	PRESS R/S WHEN DISPLAY REPEATS.				1.139
	(SHOWS CONVERGENCE.)				1.139
	(THE INTERNAL RATE OF RETURN IS 13.9%,				
	WHICH IS GREATER THAN THE MANDATED 10%.				
	MAKE THE IMPROVEMENT.)				

16.6 COST BENEFIT STREAMS

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	
002	DSP3	-63 03	
003	1	01	
004	STOI	35 46	
005	RCL1	36 01	
006	STOE	35 15	D_1
007	STO0	35 00	D_1
008	*LBLC	21 13	
009	ISZI	16 26 46	
010	RCLB	36 12	
011	1	01	
012	RCLI	36 46	1 - i
013	−	-45	
014	Y^x	31	
015	RCLi	36 45	
016	×	-35	$D_i(1+r)^{1-i}$
017	ST+0	35-55 00	PARTIAL SUM
018	0	00	
019	RCL0	36 00	TEST FOR SIGN
020	RCLE	36 15	CHANGE
021	÷	-24	
022	X<0?	16-45	
023	GSB0	23 00	
024	RCLI	36 46	
025	RCLA	36 11	
026	X=Y?	16-33	END OF LOOP ?
027	GTO2	22 02	
028	RCL0	36 00	
029	STOE	35 15	
030	GTOC	22 13	LOOP
031	*LBL0	21 00	
032	RCLI	36 46	DSP BREAKEVEN YR
033	R/S	51	
034	*LBL2	21 02	
035	RCL0	36 00	
036	RTN	24	DSP P(r)
037	*LBLB	21 12	
038	1	01	
039	STOI	35 46	
040	RCL1	36 01	
041	STO0	35 00	D_1
042	STOE	35 15	D_1
043	*LBLD	21 14	
044	ISZI	16 26 46	
045	RCLB	36 12	$1 + r_n$
046	1	01	
047	RCLI	36 46	
048	−	-45	1 - i
049	Y^x	31	
050	RCLi	36 45	
051	×	-35	$D_i(1+r_n)^{1-i}$
052	ST+0	35-55 00	$\Sigma D_i(1+r)^{1-i}$
053	RCLI	36 46	
054	×	-35	$i D_i(1+r_n)^{1-i}$
055	RCLE	36 15	
056	+	-55	
057	STOE	35 15	$\Sigma i D_i(1+r_n)^{1-i}$
058	RCLI	36 46	
059	RCLA	36 11	
060	X=Y?	16-33	END OF LOOP ?
061	GTO1	22 01	
062	GTOD	22 14	
063	*LBL1	21 01	
064	RCL0	36 00	
065	RCLE	36 15	
066	−	-45	
067	RCL0	36 00	
068	÷	-24	(5)
069	1/X	52	
070	RCLB	36 12	
071	×	-35	
072	CHS	-22	
073	RCLB	36 12	
074	+	-55	
075	STOB	35 12	$(1+r_{n+1})$
076	PRTX	-14	
077	GTOB	22 12	LOOP

REGISTERS

0	1	2	3	4	5	6	7	8	9
P(r)	D_1	D_2	-	-	-	-			-

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
-	-	-	-	-	-	-	-	-	D_{19}

A	B	C	D	E	I
N	(1+r)			USED	

PART IV

MATHEMATICAL FUNCTIONS AND ALGORITHMS

17. THE NORMAL FUNCTION AND ITS INVERSE

17.1. REFERENCES

None.

17.2. DISCUSSION

This program is frequently used in conjunction with others, such as the Q function (Program 18). Extensive application of the error function is made in *Hewlett Packard HP-65 Programs for Evaluating Effectiveness of Field Artillery Weapons*, prepared for the Joint Technical Coordinating Group for Munitions Effectiveness (Surface-to-Surface) by Booz-Allen Applied Research, Shalimar, Florida, September 1975.

17.3. EQUATIONS

The normal distribution, or function, is

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\,(-t^2/2)\,dt\ , \qquad (1)$$

where x is in units of σ. An early approximation, due to J. D. Williams, is

$$F_0(x) = \frac{1 + [1 - \exp\,\{-2x^2/\pi\}]^{1/2}}{2}\ , \qquad x \geq 0\ , \qquad (2)$$

which has a maximum error of about 0.002. R. N. Snow replaces the curly brackets in (2) by

$$-2x^2\left(\frac{1}{\pi} - \frac{x^2}{x^4 + 230}\right)$$

and obtains an accuracy better than 0.0001.

PRECEDING PAGE BLANK

A first approximation to the inverse is obtained by solving (2) for x,

$$x_0 = \left[- \frac{\pi \, \ell n \{1 - (2F_0 - 1)^2\}}{2} \right]^{1/2} \quad (3)$$

For greater accuracy, x_0 is used in solving the equation $F(x) - y = 0$ by Newton's method, which is appropriate since the derivative is simple,

$$F'(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2) .$$

The error function is

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt , \quad \text{and} \quad (4)$$

$$\text{erf}(x) = 2 \cdot F(\sqrt{2} \cdot x) - 1 . \quad (5)$$

Given a value E to find the corresponding x, use this program with $F = 1/2\,[1 + E]$ and divide the resulting value by $\sqrt{2}$.

Example 1. Find erf(0.5).

Key in $1/\sqrt{2}$, PRESS A, then 2, x, 1 - .
Answer is 0.5206.

Example 2. Find x for erf(x) = 0.5206.

Key in F = (1 + 0.5206)/2, PRESS B, PRESS C, $\sqrt{2}$, ÷ .
Answer is 0.50.

17.4. PROGRAM NOTES

None.

17.5 USER INSTRUCTIONS

17. THE NORMAL FUNCTION AND ITS INVERSE

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	KEY x	.233		
2	PRESS A		A	.5921
	OUTPUT IS F(x)			
3	KEY F	.5921		
4	PRESS B		B	
	OUTPUT IS x			.2329
5	NOW PRESS C FOR A REFINED VALUE OF x, AND CONTINUE IF DESIRED		C	.2330
6	(PRESS A AGAIN TO RETRIEVE F)			

17.6 THE NORMAL FUNCTION

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	
002	STOA	35 11	x
003	GSB0	23 00	
004	RCLA	36 11	
005	X<0?	16-45	x<0
006	GTO1	22 01	
007	RCL0	36 00	
008	DSP4	-63 04	
009	RTN	24	
010	*LBL1	21 01	CORRECTS FOR x<0
011	1	01	
012	RCL0	36 00	
013	-	-45	
014	DSP4	-63 04	
015	RTN	24	
016	*LBL0	21 00	
017	RCLA	36 11	
018	X^2	53	
019	STO1	35 01	
020	X^2	53	
021	2	02	
022	3	03	
023	0	00	
024	+	-55	
025	1/X	52	
026	RCL1	36 01	
027	×	-35	
028	CHS	-22	
029	Pi	16-24	
030	1/X	52	
031	+	-55	
032	RCL1	36 01	
033	×	-35	
034	2	02	
035	×	-35	
036	CHS	-22	
037	e^x	33	
038	CHS	-22	
039	1	01	
040	+	-55	
041	√X	54	
042	1	01	
043	+	-55	
044	2	02	
045	÷	-24	
046	STO0	35 00	F(x)
047	RTN	24	
048	*LBLB	21 12	
049	STOB	35 12	F
050	2	02	
051	×	-35	
052	1	01	
053	-	-45	
054	X^2	53	
055	CHS	-22	
056	1	01	
057	+	-55	
058	LN	32	
059	Pi	16-24	
060	×	-35	
061	2	02	
062	÷	-24	
063	CHS	-22	
064	√X	54	
065	STO2	35 02	x_o
066	DSP4	-63 04	
067	RCLB	36 12	
068	.	-62	
069	5	05	
070	X>Y?	16-34	.5 > F
071	GTO2	22 02	
072	RCL2	36 02	
073	RTN	24	
074	*LBL2	21 02	
075	RCL2	36 02	
076	CHS	-22	CHANGE TO $-x_o$
077	STO2	35 02	
078	RTN	24	
079	*LBLC	21 13	
080	RCL2	36 02	
081	X^2	53	
082	2	02	
083	÷	-24	
084	CHS	-22	
085	e^x	33	
086	Pi	16-24	
087	2	02	
088	×	-35	
089	√X	54	
090	÷	-24	
091	STO5	35 05	$F'(x_o)$
092	RCL2	36 02	
093	GSBA	23 11	
094	CHS	-22	
095	RCLB	36 12	
096	+	-55	
097	RCL5	36 05	
098	÷	-24	
099	RCL2	36 02	x_1
100	+	-55	
101	STO2	35 02	
102	DSP3	-63 03	
103	RTN	24	

REGISTERS

0	1	2	3	4	5	6	7	8	9
F(x)	x^2	x_o			$F'(x_o)$				

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I
x	F				

18. THE Q FUNCTION (OFFSET COVERAGE FUNCTION)

18.1. REFERENCES

a. J. I. Marcum and P. Swerling, "Studies of Target Detection by Pulsed Radar," *IRE Transactions on Information Theory, Vol. IT-6, No. 2,* April 1960. (Reprints of Rand RM-754, December 1947, and RM-1217, March 1954.)

b. J. I. Marcum, *Tables of Q Functions*, The Rand Corporation, RM-339 (ASTIA No. AD 116551), January 1950.

c. D. P. Meyer and H. A. Mayer, *Radar Detection*, Academic Press, New York and London, 1973.

d. L. A. Wainstein and V. D. Zubakov, *Extraction of Signals from Noise,* Prentice-Hall, Englewood Cliffs, N.J., 1962.

e. L. E. Brennan and I. S. Reed, "An Iterative Method of Computing the Q Function," *IRE Transactions on Information Theory, Vol. IT-11, No. 2,* April 1965.

18.2. DISCUSSION

The Q function

$$Q(r,R) = \int_R^\infty u \exp\left\{-\frac{r^2 + u^2}{2}\right\} \cdot I_0(ru)\, du \qquad (1)$$

is basic in radar detection theory. It is expressible in Lommel functions of the first kind (Ref. a), but is not integrable in closed form.

The Q function's more common application is perhaps in offset bombing calculations. For a circular normal distribution $(0,\sigma)$, a weapon radius R (in units of σ), and a point target at a distance r (in units of σ) from the origin (the aiming point), the probability of coverage is simply $P(R,r) = 1 - Q(r,R)$. If CEP in feet is used and r', R' are in feet,

$$r = r' \frac{\sqrt{2 \ln 2}}{CEP}, \qquad R = R' \frac{\sqrt{2 \ln 2}}{CEP},$$

$$\sqrt{2 \ln 2} = 1.17741 .$$

The damage probability program (Sec. 9) can be used to get weapon radius, and this program is then employed to find collateral damage to other point targets.

18.3. EQUATIONS

The Bessel function $I_0(x)$ is given by

$$I_0(x) = \sum_{n=0}^{\infty} (\tfrac{x}{2})^{2n}/(n!)^2 \ . \qquad (2)$$

Put (2) in (1) and interchange the order of summation and integration to get

$$P(R,r) = 1 - Q(r,R) = \sum_{n=0}^{\infty} k_n(r^2/2)\ K_n(R^2/2) \ , \qquad (3)$$

where

$$k_n(y) = x^n e^{-y}/n!$$

$$K_n(x) = \int_0^x \frac{u^n e^{-u}}{n!}\, du = \frac{\Gamma_{n+1}(x)}{n!} \ , \qquad (4)$$

and Γ_{n+1} is the incomplete gamma function.

The recursion relations for k_n and K_n are

$$k_0(y) = e^{-y}$$

$$k_n(y) = \frac{y}{n} k_{n-1}(y) \ , \qquad n > 0$$

$$K_0(x) = 1 - e^{-x} \qquad (5)$$

$$K_n(x) = K_{n-1}(x) - k_n(x) \ , \qquad n > 0 \ .$$

After N iterations,

$$P(R,r) = \sum_{n=1}^{N-1} k_n K_n + R(N) ,$$

where the remainder

$$R(N) = \sum_{n=N}^{\infty} k_n K_n . \tag{6}$$

Reference e shows, for $N > rR/\sqrt{2}$, that

$$R(N + 1) \leq k_N(r^2/2)\, K_N(R^2/2) . \tag{7}$$

Let $N_0 = rR/\sqrt{2}$ and let Δ be the desired accuracy. Then the iteration can be terminated at n = N, if

$$N > N_0 , \quad k_N(r^2/2)\, K_N(R^2/2) \leq \Delta . \tag{8}$$

If r and R are small, the convergence is rapid. But as r and R increase, the number of terms required--and thus computation time--become excessive.

However, for r, R ≥ 5, Ref. d provides an excellent approximation in terms of the cumulative Gaussian function

$$P(R,r) = \frac{1}{\sqrt{2\pi}} \int_{Z_0}^{\infty} e^{-u^2/2}\, du + E(Z_0) , \tag{9}$$

where

$$Z_0 = (R - r)(1 - 1/4r^2) - 1/2r . \tag{10}$$

The approximation is improved by adding two more terms to (10), taking

$$Z_1 = (R - r)(1 - 1/4r^2) - 1/2r - 1/48r^3 - (R - r)^2/10r^3 \quad (11)$$

Then the error $E(Z_1) \leq 0.0005$ for $r \geq 4.25$. It can also be shown that for

$$R \geq 2.45 + \frac{1}{r - 1.7} ,$$

$$E(Z_1) \leq 0.005 . \quad (12)$$

The regions of interest for the P(R,r) calculation are then shown in Fig. 18.1. For region A, use the iteration method, Eqs. (3) and (8). The maximum number of iterations required to obtain an error $\Delta < 0.0005$ is 10. For region B, use the approximation (11) in (9), with an execution time of 7 sec. In general, the regions above $R - r = -2.8$ and below $R - r = 2.8$ are without interest. However, even in these regions, the approximation method may be used.

18.4. PROGRAM NOTES

1. For an accuracy $\Delta \leq 0.0005$, enter R, r, and Δ and press B. The program will choose the better method to compute P(R,r). "Better" is defined as entailing the least computer time to obtain at least accuracy Δ.

2. For an accuracy Δ better than 0.005, again enter R, r, and Δ but press A.

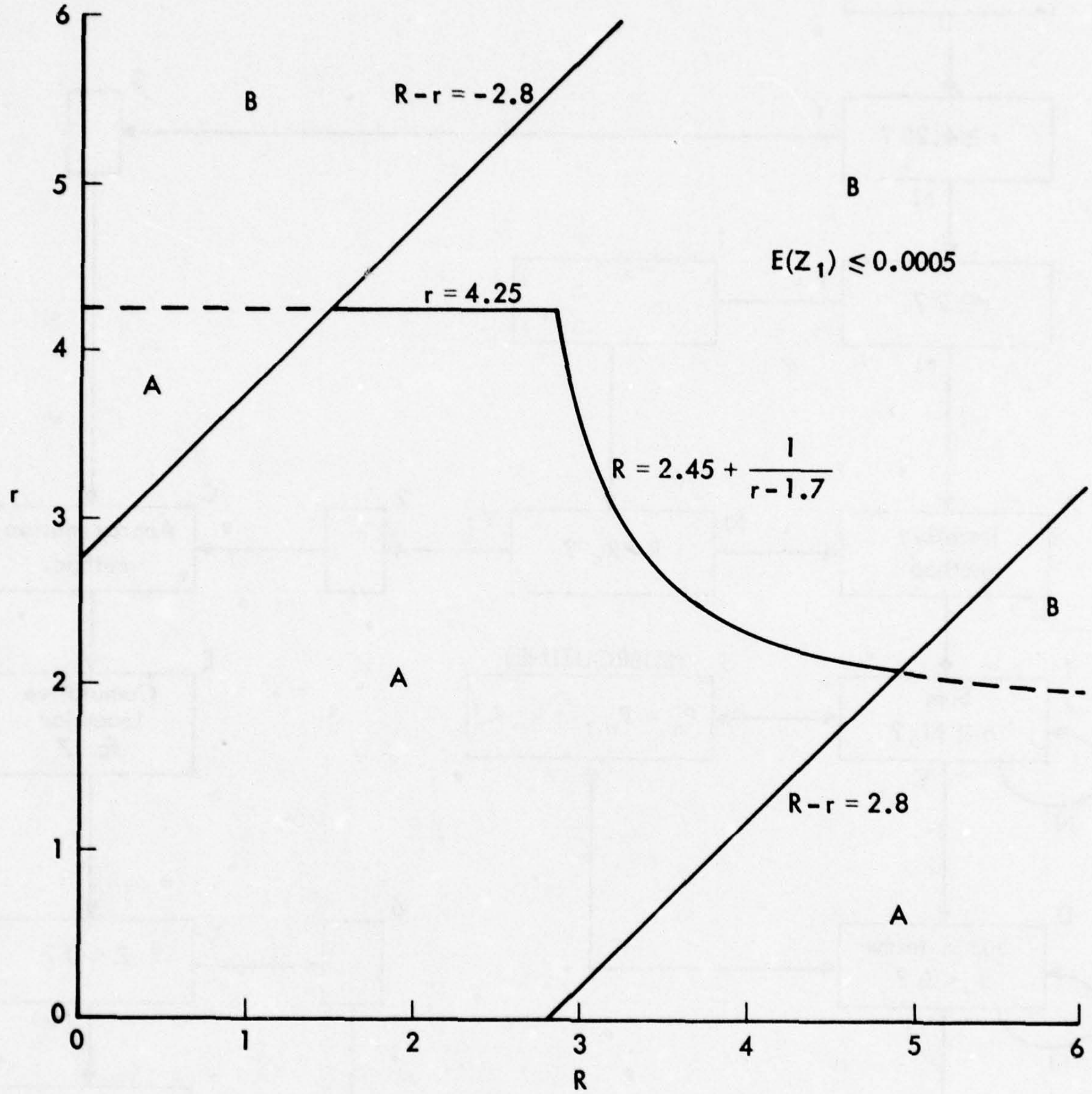


Fig. 18.1. — Iteration and approximation regions

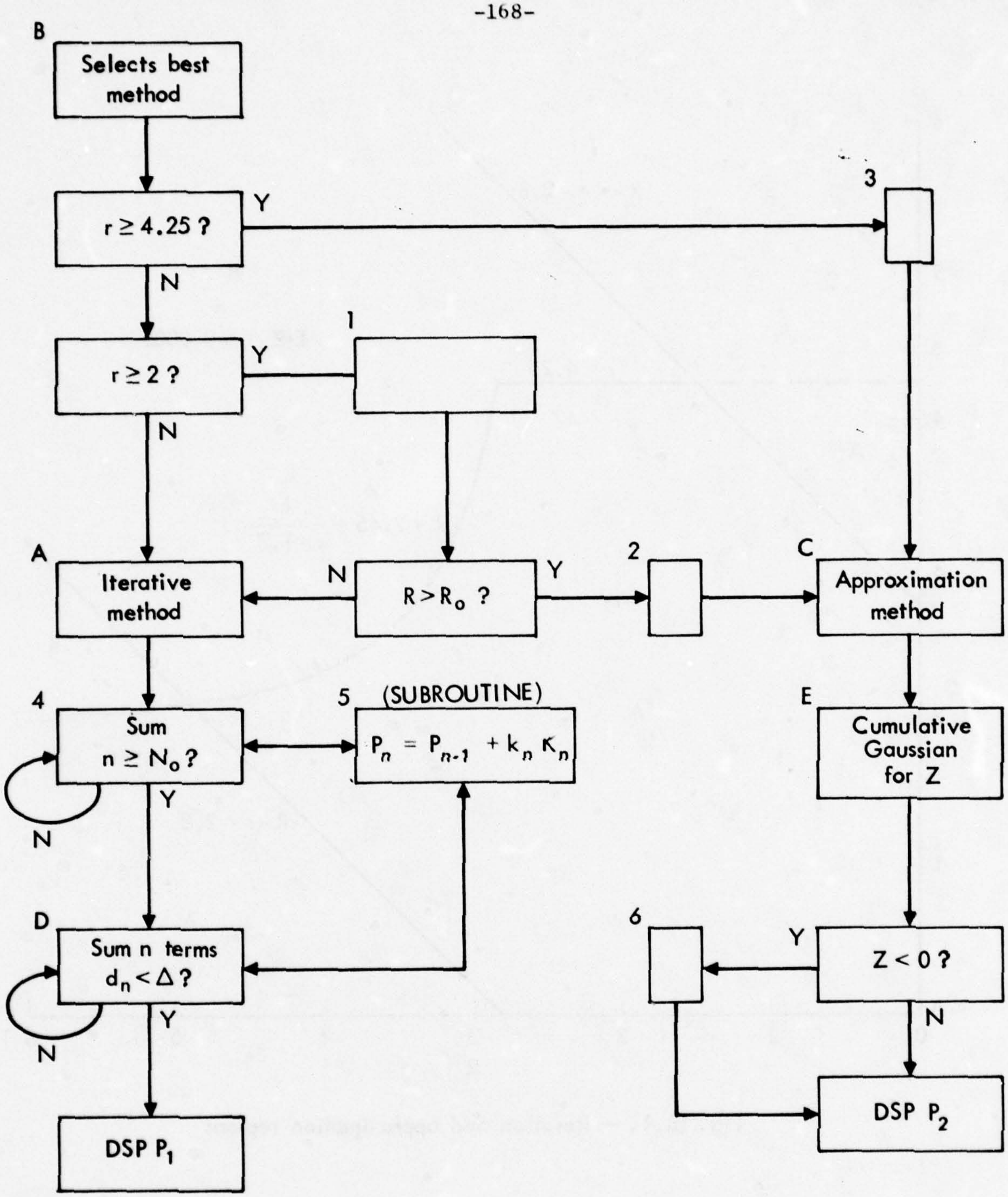
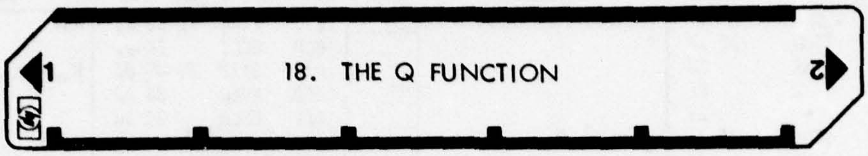


Fig. 18.2 — P(Q) function program flowchart

18.5 USER INSTRUCTIONS



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	STO Δ (ACCURACY) IN E	.0005	STO	E	0.0005
2	KEY R, ENTER	2.6		ENT	2.6000
3	KEY r, PRESS B	2.5		B	0.4217
	(BEST METHOD IS CHOSEN FOR				
	4 PLACE ACCURACY BY THE PRGM)				
	TO SELECT APPROX. METHOD				
4	STO Δ IN E (DSP 5)	.00005	STO	E	.00005
5	KEY R, ENTER	2.6		ENT	2.6000
6	KEY r, PRESS A (DSP 5)	2.5		A	0.42172
	TO SELECT APPROX. METHOD FOR				
	THIS CASE DO STEPS 4, 5, 6				
	PRESS C				
				C	0.4604
	(WRONG REGION FOR APPROX. METHOD)				

18.6 THE Q FUNCTION

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
001	001 *LBLA	21 11	ITERATIVE SOLUTION		057 STx1	35-35 01	kn
	002 STOD	35 14	r		058 RCL1	36 01	
	003 X²	53			059 ST-2	35-45 02	$K_n = K_{n-1} - k_n$
	004 2	02		060	060 RCLB	36 12	
	005 ÷	-24			061 RCL0	36 00	
	006 STOB	35 12	$y = r^2/2$		062 ÷	-24	y/n
	007 CHS	-22			063 RCL3	36 03	
	008 eˣ	33			064 ×	-35	J_n
	009 STO3	35 03	$J = e^{-y}$		065 STO3	35 03	
010	010 STO4	35 04			066 RCL2	36 02	
	011 R↓	-31			067 ×	-35	
	012 STOC	35 13	R		068 ST+4	35-55 04	$P_n = P_{n-1} + J_n K_n$
	013 X²	53			069 RTN	24	
	014 2	02		070	070 *LBLB	21 12	SELECTS BEST METHOD
	015 ÷	-24			071 STOD	35 14	r
	016 STOA	35 11	$x = R^2/2$		072 4	04	
	017 CHS	-22			073 .	-62	
	018 eˣ	33			074 2	02	
	019 STO1	35 01	$k_o = e^{-x}$		075 5	05	
020	020 CHS	-22			076 X≤Y?	16-35	$4.25 \leq r$?
	021 1	01			077 GTO3	22 03	$r \geq 4.25$
	022 +	-55			078 R↓	-31	r
	023 STO2	35 02	$K_o = 1 - k_o$		079 2	02	
	024 STx4	35-35 04	$J_o K_o$	080	080 X≤Y?	16-35	$2 \leq r$?
	025 RCLC	36 13			081 GTO1	22 01	$2 \leq r \leq 4.25$
	026 RCLD	36 14			082 R↓	-31	r
	027 ×	-35			083 GTOA	22 11	TO A IF r<2
	028 2	02			084 *LBL1	21 01	
	029 √X	54			085 R↓	-31	
030	030 ÷	-24	$N_o' = Rr/\sqrt{2}$		086 1	01	
	031 1	01			087 .	-62	
	032 +	-55	$N_o = 1 + Rr/\sqrt{2}$		088 7	07	
	033 STOI	35 46			089 −	-45	
	034 0	00		090	090 1/X	52	
	035 STO0	35 00	n = 0		091 2	02	
	036 *LBL4	21 04	LOOP UNTIL $N \geq N_o$		092 .	-62	
	037 GSB5	23 05			093 4	04	
	038 DSZI	16 25 46	$N_o = 0$		094 5	05	
	039 GTO4	22 04			095 +	-55	$R_o = 2.45 + 1/(r - 1.7)$
040	040 *LBLD	21 14			096 X≤Y?	16-35	$R_o \geq R$?
	041 GSB5	23 05			097 GTO2	22 02	
	042 RCLE	36 15			098 R↓	-31	R
	043 X≤Y?	16-35			099 RCLD	36 14	r
	044 GTOD	22 14		100	100 GTOA	22 11	TO A IF $R > R_o$
	045 RCL0	36 00			101 *LBL2	21 02	
	046 DSP0	-63 00			102 R↓	-31	R
	047 PSE	16 51	FLASH n		103 RCLD	36 14	r C(approx)
	048 RCL4	36 04			104 GTOC	22 13	IF $R \geq R_o$
	049 DSP4	-63 04	DISPLAY P_1		105 *LBL3	21 03	
050	050 RTN	24			106 R↓	-31	r
	051 *LBL5	21 05			107 *LBLC	21 13	APPROX P_2
	052 1	01			108 STOD	35 14	r
	053 ST+0	35-55 00	n = n + 1		109 X⇄Y	-41	R
	054 RCLA	36 11		110	110 STOC	35 13	
	055 RCL0	36 00			111 X⇄Y	-41	
	056 ÷	-24	x/n		112 −	-45	R − r

REGISTERS

0	1	2	3	4	5	6	7	8	9
n	k_n	K_n	J_n	P_n					
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I
$x = R^2/2$	$y = r^2/2$	R	r	Δ	N_o

18.6 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	STO8	35 08	
114	RCLC	36 13	
115	3	03	
116	RCLD	36 14	
117	×	-35	
118	−	-45	
119	RCLD	36 14	
120	X²	53	
121	STO9	35 09	
122	4	04	
123	×	-35	
124	÷	-24	
125	+	-55	$z' = R - r + (R-3r)/4r^2$
126	4	04	
127	8	08	
128	RCL9	36 09	
129	×	-35	
130	1/X	52	
131	−	-45	$z'' = z' - 1/48r^2$
132	RCL8	36 08	
133	X²	53	
134	1	01	
135	8	08	
136	RCL9	36 09	
137	×	-35	
138	RCLD	36 14	
139	×	-35	
140	÷	-24	
141	−	-45	$z = z'' - (R-r)^2/12r^3$
142	*LBLE	21 15	CUMULATIVE GAUSSIAN (SEE PROGRAM 17)
143	STO6	35 06	
144	X²	53	
145	STO7	35 07	
146	X²	53	
147	2	02	
148	3	03	
149	8	08	
150	+	-55	
151	1/X	52	
152	RCL7	36 07	
153	×	-35	
154	Pi	16-24	
155	1/X	52	
156	−	-45	
157	RCL7	36 07	
158	×	-35	
159	2	02	
160	×	-35	
161	e^x	33	
162	CHS	-22	
163	1	01	
164	+	-55	
165	√X	54	
166	1	01	
167	+	-55	
168	2	02	
169	÷	-24	
170	STO5	35 05	
171	RCL6	36 06	
172	DSP4	-63 04	
173	X<0?	16-45	IF $z < 0$
174	GTO6	22 06	
175	X⇄Y	-41	
176	RTN	24	DISPLAY P_2
177	*LBL6	21 06	
178	1	01	P_2 IF $z < 0$
179	RCL5	36 05	
180	−	-45	
181	STO5	35 05	
182	RTN	24	DISPLAY P_2

LABELS

A	B	C	D	E
ITERATION	(A or C) BEST METHOD	APPROX.	LOOP, TEST△	CUM GAUSSIAN
a	b	c	d	e
0	1 TEST R VS R_o	2 (101)	3 (105)	4 LOOP, $n \leq N_o$
5 COMPUTE d_n	6 P_2 IF $z \leq 0$	7	8	9

FLAGS: 0, 1, 2, 3

SET STATUS

FLAGS	TRIG	DISP
ON OFF		
0 ☐ ☐	DEG ☐	FIX ☐
1 ☐ ☐	GRAD ☐	SCI ☐
2 ☐ ☐	RAD ☐	ENG ☐
3 ☐ ☐		n____

19. LINEAR PROGRAMMING AND 3 × 3 MATRIX GAMES

19.1. REFERENCES

a. G. B. Dantzig, *Linear Programming and Extensions*, The Rand Corporation, R-366-PR, August 1963 (published by Princeton University Press).

b. A. M. Glickman, *An Introduction to Linear Programming and the Theory of Games*, John Wiley and Sons, New York, 1963.

c. R. W. Metzger, *Elementary Mathematical Programming*, John Wiley and Sons, New York, 1958.

d. J. D. Williams, *The Compleat Strategyst*, McGraw-Hill, New York, revised edition, 1966.

e. M. Dresher, *Games of Strategy: Theory and Applications*, The Rand Corporation, R-360, May 1961 (published by Prentice-Hall).

19.2. DISCUSSION

The 3-activity linear programming problem may be formulated:

find $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$ satisfying

$$a_{11}\, x_1 + a_{12}\, x_2 + a_{13}\, x_3 \leq b_1$$

$$a_{21}\, x_1 + a_{22}\, x_2 + a_{23}\, x_3 \leq b_2$$

$$a_{31}\, x_1 + a_{32}\, x_2 + a_{33}\, x_3 \leq b_3$$

to maximize

$$M = c_1x_1 + c_2x_2 + c_3x_3 \; .$$

The recipe for solving this problem by the pivot method is very simple, although the result of deep and extensive analysis. Set up Tableau 1:

	x_1	x_2	x_3		
	a_{11}	a_{12}	a_{13}	b_1	u_1
	a_{21}	a_{22}	a_{23}	b_2	u_2
	a_{31}	a_{32} (circled)	a_{33}	b_3	u_3
	$-c_1$	$-c_2$	$-c_3$	0	M

TABLEAU 1

1. The pivot *column* corresponds to the most *negative* of $-c_1$, $-c_2$, $-c_3$. Say this is $-c_2$.

2. The pivot *row* is found by finding

$$b_1/a_{12}\ ,\quad b_2/a_{22}\ ,\quad b_3/a_{32}$$

for only those b's that are *positive*, and then selecting the minimum. Say this is b_3/a_{32}. Then the pivot is a_{32}.

3. Replace the pivot by its reciprocal, and divide all other entries in the pivot's column by the negative of the pivot.

4. For an entry *other* than those in the pivot's row and column, add to that entry the product of the entry in the same column to the left or right of the pivot in *its* row and the entry in the (new) pivot's column to the left or right of the entry in *its* row.

5. Now modify the pivot's row by dividing all entries other than the pivot by the pivot.

6. Interchange x_2 and u_3.

These operations produce the new Tableau 2:

x_1	u_3	x_3		
$a_{11} + a_{31}\ (-a_{12}/a_{32})$	$-a_{12}/a_{32}$	$a_{13} + a_{33}\ (-a_{12}/a_{32})$	$b_1 + b_3\ (-a_{12}/a_{32})$	u_1
$a_{21} + a_{31}\ (-a_{22}/a_{32})$	$-a_{22}/a_{32}$	$a_{23} + a_{33}\ (-a_{22}/a_{32})$	$b_2 + b_3\ (-a_{22}/a_{32})$	u_2
a_{21}/a_{32}	$1/a_{32}$	a_{33}/a_{32}	b_3/a_{32}	x_2
$-c_1 + a_{31}\ (c_2/a_{32})$	c_2/a_{32}	$-c_3 + a_{33}\ (c_2/a_{32})$	$0 + b_3\ (c_2/a_{32})$	M

TABLEAU 2

In the numerical example below, the initial pivot is a_{23} (*not* a_{32} as above).

<u>Example</u> (Ref. b, Sec. 5)

x_1	x_2	x_3		
1	1	1	100	u_1
3	2	(4)	210	u_2
3	2	0	150	u_3
-5	-4	-6	0	M

(Tableau 1)

x_1	x_2	u_2		
1/4	1/2	-1/4	47.5	u_1
3/4	1/2	1/4	52.5	x_3
3	(2)	0	150.0	u_3
-1/2	-1	3/2	315.0	M

(Tableau 2)

7. Repeat the above process until all entries in the bottom row are *nonnegative*. In the rightmost column read the optimal x's and maximum M. Any u left in that column indicates that the corresponding x is 0.

Continuing the example,

x_1	u_3	u_2		
-1/2	-1/4	-1/4	10	u_1
0	-1/4	1/4	15	x_3
3/2	1/2	0	75	x_2
1	1/2	3/2	390	M

$x_1 = 0,\ x_2 = 75,\ x_3 = 15,\ M = 390$ (Answer)

(Tableau 3)

The previous example in equation form is:

find $x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0$ subject to

$$x_1 + x_2 + x_3 \leq 100$$

$$3x_1 + 2x_2 + 4x_3 \leq 210$$

$$3x_1 + 2x_2 \leq 150$$

to *maximize* $M = 5x_1 + 4x_2 + 6x_3$.

Suppose the problem were:

find $x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0$ subject to

$$x_1 + x_2 + x_3 \geq 100$$

$$3x_1 + 2x_2 + 4x_3 \geq 210$$

$$3x_1 + 2x_2 \geq 150$$

to *minimize* $m = 5x_1 + 4x_2 + 6x_3$.

We set up the *dual* problem (the pattern should be clear):

find $y_1 \geq 0$, $y_2 \geq 0$, $y_3 \geq 0$ subject to

$$y_1 + 3y_2 + 3y_3 \leq 5$$

$$y_1 + 2y_2 + 2y_3 \leq 4$$

$$y_1 + 4y_2 \leq 6$$

to *maximize* $M = 100y_1 + 210y_2 + 150y_3$, and solve this problem by the previous method. The values for x variables are now read off on the bottom row under the interchanged u's with the same subscript.

The relation of the duals is clarified by using the program to solve 3 × 3 matrix games. The procedure is readily understood by following through a specific example.

BLUE is the maximizing player, RED his minimizing opponent. It is desirable to add a constant to all entries in the payoff matrix to make all entries positive if necessary.

This increases the value of the original game by that constant but does not change the proportions in which the strategies are played. The new example is

		RED		
		r_1	r_2	r_3
	b_1	2	6	0
BLUE	b_2	5	3	6
	b_3	5	4	3

where b_i (r_j) is the probability BLUE (RED) will choose course of action or strategy i (j). Then

$$1 \geq b_i \geq 0,\ 1 \geq r_j \geq 0,\ b_1 + b_2 + b_3 = 1,\ r_1 + r_2 + r_3 = 1\ .$$

Look at matters from RED's point of view. Against each of BLUE's *pure* strategies, RED must expect to pay BLUE

$$2r_1 + 6r_2,\ 5r_1 + 3r_2 + 6r_3,\ 5r_1 + 4r_2 + 3r_3\ .$$

Let μ (unknown) be the greatest of these three. Then putting $y_i = r_i/\mu$,

$$2y_i + 6y_2 \leq 1,\ 5y_1 + 3y_2 + 6y_3 \leq 1,\ 5y_1 + 4y_2 + 3y_3 \leq 1\ ,$$

and RED wants to *minimize* μ by *maximizing* $M = 1/\mu$,

$$M = y_1 + y_2 + y_3\ .$$

This is the standard linear programming problem with the right-upper border all +1's and the left lower border all -1's. The first tableau is:

y_1	y_2	y_3		
2	(6)	0	1	v_1
5	3	6	1	v_2
5	4	3	1	v_3
-1	-1	-1	0	M

Any column could be the first pivot column. We choose the circled number as the pivot because $1/6 < 1/5$ (a variation on step (2) above).

The second tableau (by the program) is:

y_1	v_1	y_3		
2	.17	0	.17	y_2
4	-.50	(6)	.50	v_2
3.67	-.67	3	.33	v_3
-.67	.17	-1	.17	M

The third tableau is:

y_1	v_1	v_2		
2	.17	0	.17(1/6)	y_2
.67	-.08	.17	.08(1/12)	y_3
1.67	-.42	-.50	.08(1/12)	v_3
0	.08	.17	.25(1/4)	M

[illegible]

it is given in the high-level language appropriate to our minds, but only with care and time do we avoid errors.

The HP-67 understands only a relatively low-level language even though it is quite advanced over early coding in machine language. So what we see without effort (the *Gestalt*) requires many steps of programming.

2. The programming problem is compounded in difficulty because the HP-67 cannot work with the double indices of matrix elements. We must "linearize" the matrix and devise an "address arithmetic" for this problem, and then exploit the indirect addressing capability of the I-register.

Consider item (4) of the recipe. Overlay the linear programming matrix with the matrix of register addresses of entries:

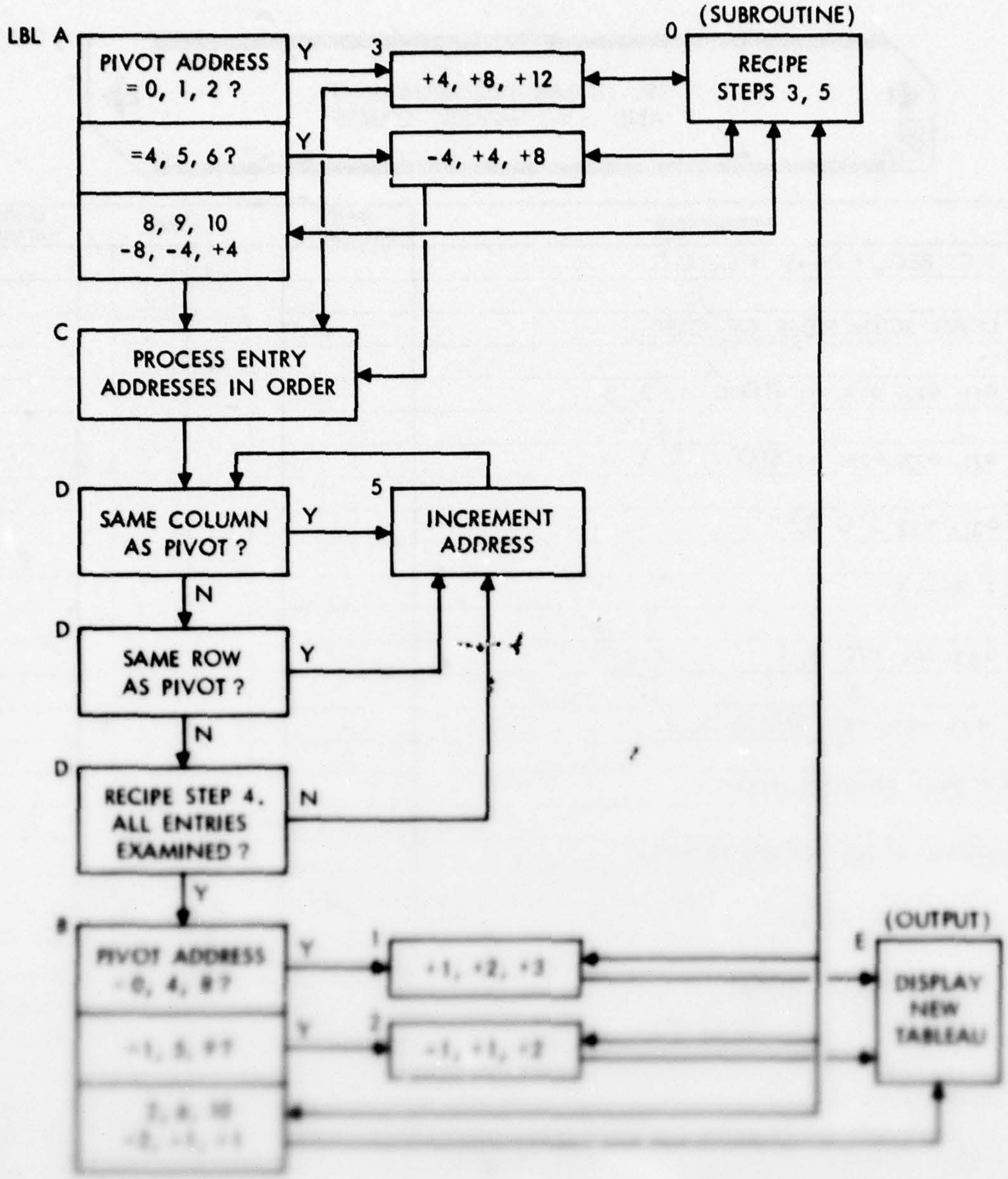
0	1	2(y)	3(e)
4	5	6	7
8	9	10(p)	11(x)
12	13	14	15

[illegible]

4. Items (1) and (2) of the recipe are not carried out by this program. They can be programmed, but only at the cost of an additional program card and thus of time. And in this case a man can find the pivot faster than the machine can if he has the matrix in front of him.

The philosophical argument is that the HP-67 and the user are engaged in a cooperative one-shot (or few-shot) enterprise, rather than bulk or production computing. Each partner should contribute what he can do better and quicker.

5. Running time for each iteration is 47 sec, exclusive of the final display (f -x-) of the new tableau for manual recording.


(SUBROUTINE)
0
RECIPE STEPS 3, 5
LBL A
PIVOT ADDRESS =0, 1, 2 ?
Y
3
+4, +8, +12
=4, 5, 6 ?
Y
-4, +4, +8
8, 9, 10
-8, -4, +4
C
PROCESS ENTRY ADDRESSES IN ORDER
D
SAME COLUMN AS PIVOT ?
Y
5
INCREMENT ADDRESS
N
D
SAME ROW AS PIVOT ?
Y
N
D
RECIPE STEP 4. ALL ENTRIES EXAMINED ?
N
Y
PIVOT ADDRESS
Y
Y
(OUTPUT)
DISPLAY NEW TABLEAU

19.5 USER INSTRUCTIONS

19. LINEAR PROGRAMMING AND 3 X 3 MATRIX GAMES

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	f CL REG, f P⟷S, f CL REG			
2	LOAD BOTH SIDES OF CARD			
3	a_{11}, a_{12}, a_{13}, b_1 STO 0, 1, 2, 3			
4	a_{21}, a_{22}, a_{23}, b_2 STO 4, 5, 6, 7			
5	a_{31}, a_{32} STO 8, 9			
6	f P⟷S			
7	a_{33}, b_3 STO 0, 1			
8	$-c_1$, $-c_2$, $-c_3$ STO 2, 3, 4			
9	f P⟷S (IMPORTANT)			
	(PRESS E TO REVIEW ENTRIES)			
10	DETERMINE PIVOT			
11	KEY IN PIVOT ADDRESS (0, 1, 2, 4, 5, 6, 8, 9, 10)			
12	PRESS A			
13	ON FLASHING STOP, RECORD NEW TABLEAU IN LEFT TO RIGHT ORDER			
14	PRESS E TO REVIEW TABLEAU AGAIN IF NEEDED, AS ON INITIAL INPUT			
15	GO TO STEP [illegible] ABOVE			
[illegible]	[illegible]			

19.5 USER INSTRUCTIONS

19. LINEAR PROGRAMMING (EXAMPLE)

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
	THE EXAMPLE IS IN THE TEXT,				
	TABLEAU 2 INTO TABLEAU 3	.25	STO	0	
		.5		1	
		-.25		2	
		47.5		3	
		.75		4	
		.5		5	
		.25		6	
		52.5		7	
		3		8	
	PIVOT	2		9	
	f P↔S	0	STO	0	
		150		1	
		-.5		2	
		-1		3	
		1.5		4	
		315		5	
			f	P↔S	
	PIVOT ADDRESS	9		A	-.50
					-.25
					-.25
					10.00
					0.00
					-.25
					.25
	x_3				15.00
					1.50
					.50
					0.00
	x_2				75.00
					1.00
					.50
					1.50
	M				[illegible]
	LAST NR. IN x_1				[illegible]

19.6 LINEAR PROGRAMMING

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	
002	STOI	35 46	PIVOT ADD(RESS)
003	STOB	35 12	
004	RCLi	36 45	
005	1/X	52	
006	CHS	-22	
007	STOA	35 11	−1/PIVOT VALUE
008	RCLB	36 12	
009	4	04	
010	÷	-24	
011	INT	16 34	
012	X=0?	16-43	PIVOT ADD = 0, 1, 2
013	GTO3	22 03	
014	1	01	
015	X=Y?	16-33	PIVOT ADD = 4, 5, 6
016	GTO4	22 04	
017	RCLB	36 12	
018	8	08	PIVOT ADD −8
019	−	-45	
020	STOI	35 46	
021	GSB0	23 00	
022	RCLB	36 12	
023	4	04	
024	−	-45	PIVOT ADD −4
025	STOI	35 46	
026	GSB0	23 00	
027	RCLB	36 12	
028	4	04	
029	+	-55	PIVOT ADD + 4
030	STOI	35 46	
031	GSB0	23 00	
032	GTOC	22 13	
033	*LBL0	21 00	(RECIPE STEPS 3, 5)
034	RCLi	36 45	
035	RCLA	36 11	
036	×	-35	COL. ENTRY
037	STOi	35 45	DIVIDED BY−PIVOT
038	RTN	24	
039	*LBL3	21 03	
040	RCLB	36 12	
041	4	04	
042	+	-55	
043	STOI	35 46	PIVOT ADD + 4
044	GSB0	23 00	(PIVOT COL)
045	RCLB	36 12	
046	8	08	
047	+	-55	PIVOT ADD + 8
048	STOI	35 46	(PIVOT COL)
049	GSB0	23 00	
050	RCLB	36 12	
051	1	01	
052	2	02	
053	+	-55	PIVOT ADD + 12
054	STOI	35 46	(PIVOT COL)
055	GSB0	23 00	
056	GTOC	22 13	

STEP	KEY ENTRY	KEY CODE	COMMENTS
057	*LBL4	21 04	
058	RCLB	36 12	
059	4	04	
060	−	-45	PIVOT ADD −4
061	STOI	35 46	
062	GSB0	23 00	
063	RCLB	36 12	
064	4	04	
065	+	-55	PIVOT ADD +4
066	STOI	35 46	
067	GSB0	23 00	
068	RCLB	36 12	
069	8	08	
070	+	-55	PIVOT ADD + 8
071	STOI	35 46	
072	GSB0	23 00	
073	GTOC	22 13	
074	*LBLC	21 13	
075	0	00	
076	STOC	35 13	FIRST ENTRY ADD.
077	GTOD	22 14	
078	*LBLD	21 14	
079	RCLB	36 12	PIVOT ADD.
080	4	04	
081	÷	-24	
082	INT	16 34	
083	RCLC	36 13	ENTRY ADD.
084	4	04	
085	÷	-24	
086	INT	16 34	
087	−	-45	
088	4	04	
089	×	-35	$\bar{z}$ (PRGM NOTE 2)
090	RCLC	36 13	
091	+	-55	
092	STOD	35 14	$\bar{x} = \bar{e} + \bar{z}$
093	RCLB	36 12	
094	X=Y?	16-33	TEST FOR ENTRY
095	GTO5	22 05	IN SAME COL. AS PIVOT
096	RCLD	36 14	
097	−	-45	
098	RCLC	36 13	
099	+	-55	$\bar{p} - \bar{z} + \bar{e} = \bar{p} - \bar{z}$
100	STOE	35 15	$= \bar{y}$
101	RCLB	36 12	
102	X=Y?	16-33	TEST FOR ENTRY
103	GTO5	22 05	IN SAME ROW AS PIVOT
104	RCLD	36 14	
105	STOI	35 46	
106	RCLi	36 45	$x = [\bar{x}]$
107	RCLE	36 15	
108	STOI	35 46	
109	X⇄Y	-41	INTERCHANGE
110	RCLi	36 45	$y = [\bar{y}]$
111	×	-35	xy
112	RCLC	36 13	$\bar{e}$

REGISTERS

a_{11}	a_{12}	a_{13}	b_1	a_{21}	a_{22}	a_{23}	b_2	a_{31}	a_{32}
a_{33}	b_3	$-c_1$	$-c_2$	$-c_3$	M				

−1/PIVOT	PIVOT ADDRESS	ENTRY ADDRESS	1 (OLD) ADDRESS	1 NEW ADDRESS	

19.6 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	STOI	35 46	
114	X⇄Y	-41	
115	RCLi	36 45	
116	+	-55	
117	RCLC	36 13	
118	STOI	35 46	
119	X⇄Y	-41	
120	STOi	35 45	(RECIPE STEP 4)
121	1	01	
122	5	05	
123	RCLC	36 13	
124	X≤Y?	16-35	ALL ENTRIES ?
125	GTO5	22 05	
126	GTOB	22 12	
127	*LBL5	21 05	
128	RCLC	36 13	
129	1	01	
130	+	-55	
131	STOC	35 13	INCREMENT
132	GTOD	22 14	ENTRY ADD.
133	*LBLB	21 12	
134	RCLA	36 11	
135	CHS	-22	
136	STOA	35 11	1/p
137	RCLB	36 12	
138	STOI	35 46	p
139	RCLA	36 11	
140	STOi	35 45	RECIP. OF PIVOT
141	RCLB	36 12	
142	4	04	
143	÷	-24	
144	FRC	16 44	
145	X=0?	16-43	PIVOT ADD = 0, 4, 8
146	GTO1	22 01	
147	4	04	
148	1/X	52	
149	X=Y?	16-33	PIVOT ADD = 1, 5, 9
150	GTO2	22 02	
151	RCLB	36 12	PIVOT ADD = 2, 6, 10
152	STOI	35 46	
153	DSZI	16 25 46	
154	DSZI	16 25 46	
155	GSB0	23 00	1 ST ROW ENTRY
156	ISZI	16 26 46	
157	GSB0	23 00	2ND ROW ENTRY
158	ISZI	16 26 46	
159	ISZI	16 26 46	
160	GSB0	23 00	ROW ENTRY TO RIGHT
161	GTOE	22 15	
162	*LBL1	21 01	
163	RCLB	36 12	
164	STOI	35 46	
165	ISZI	16 26 46	p̄ + 1
166	GSB0	23 00	ROW ENTRY/p
167	ISZI	16 26 46	
168	GSB0	23 00	NEXT ROW ENTRY
169	ISZI	16 26 46	
170	GSB0	23 00	LAST ROW ENTRY
171	GTOE	22 15	
172	*LBL2	21 02	
173	RCLB	36 12	
174	STOI	35 46	
175	DSZI	16 25 46	ROW ADD TO LEFT
176	GSB0	23 00	
177	ISZI	16 26 46	
178	ISZI	16 26 46	ROW ADD TO RIGHT
179	GSB0	23 00	
180	ISZI	16 26 46	
181	GSB0	23 00	LAST ROW ENTRY
182	GTOE	22 15	
183	*LBLE	21 15	REVIEW TABLEAU
184	0	00	
185	STOI	35 46	
186	GTO6	22 06	
187	*LBL6	21 06	
188	RCLi	36 45	
189	PRTX	-14	f - x -
190	ISZI	16 26 46	
191	1	01	
192	5	05	
193	RCLI	36 46	
194	X≤Y?	16-35	ALL ENTRIES ?
195	GTO6	22 06	
196	RTN	24	

LABELS	FLAGS	SET STATUS: FLAGS (ON / OFF)	TRIG	DISP
A, B, C, D, E; a, b, c, d, e; 0–9	0, 1, 2, 3	0 ☐ ☐ / 1 ☐ ☐ / 2 ☐ ☐ / 3 ☐ ☐	DEG ☐ GRAD ☐ RAD ☐	FIX ☐ SCI ☐ ENG ☐ n ___

20. FOURTH-ORDER DIFFERENTIAL EQUATIONS

20.1. REFERENCE

a. R. W. Hamming, *Numerical Methods for Scientists and Engineers*, McGraw-Hill, New York, 1962.

20.2. DISCUSSION

Systems of four first-order (frequently nonlinear) differential equations, sometimes in the guise of two second-order equations, occur more commonly than one would think: The basic beam deflection equation is of the fourth order; chemical kinetic systems with four (or more) equations are common; reentry trajectories are specified by two second-order equations (see Program 4); Lanchester equations with two force components for each side and with variable coefficients occur; problems in optimal control theory and in differential game theory lead to such systems.

Since the HP-67 has a limited number of storage registers (26) and of program steps (224), and since program space is needed to define the functions of the system, we seek an alternative to the rather complex Runge-Kutta "standard" formulas, an alternative that is miserly of program space and yet has good accuracy for relatively large time intervals--that is, *good relative stability*, defined as the rate of growth of the error relative to the growth of the solution.

Moreover, programming must fully exploit the indirect control afforded by the powerful I-register. That is, the number in the I-register can be the *address* of a storage register or the *name* of a label (subroutine). Then the instruction STO (i) or RCL (i) or GTO (i) moves X-register data to the right register or recalls data from the desired register or sends the program to the right place. Hence, in conjunction with incrementing and decrementing the I-register, serial treatment of all four equations can be accomplished with the same economical set of processing instructions.

The final programming problem is to move data around, like freight cars in a marshalling yard, so that storage spaces are freed just in time to make space for a new claimant.

20.3. EQUATIONS

Section 14.3 of Ref. a describes a simple predictor-corrector approach for first-order equations that seems to have promise and is readily extended to a system of equations.

Consider the equation

$$dx/dt = X(x,t), \; x(0) = x_0 \; .$$

Let the time interval be h. Suppose that at time (n - 1)h we are at x_{n-1}. Then a good *predicted* value for x_{n+1} appears to be

$$p_{n+1} = x_{n-1} + 2h \, x'_n$$

$$= x_{n-1} + 2h \, X(x_n,t) \; ,$$

since x'_n is the slope at the midpoint of the double interval. According to Hamming, the error term is $h^3 x'''(\theta)/3$.

The value p_{n+1} is now *corrected* by taking

$$c_{n+1} = x_n + h \, [x'_{n+1} + x'_n]/2 \; ,$$

where p_{n+1} is used to determine $p'_{n+1} = X(p_{n+1},t)$. The error term for c_{n+1} is $-h^3 x'''(0)/12$. If x''' is approximately constant in the interval, then

$$p_{n+1} - c_{n+1} = 5h^3 x'''/12$$

$$= (x_{n+1} - c_{n+1}) \cdot 5$$

or

$$x_{n+1} = (4c_{n+1} + p_{n+1})/5$$

$$= [4x_n + p_{n+1} + 2h(x'_n + p'_{n+1})]/5 \; ,$$

and

$$p'_{n+1} = X(p_{n+1}, \overline{n+1}\,h) .$$

To get started, we need x_1 in addition to x_0. This is done by expanding in a Taylor's series

$$x_1 = x_0 + hx'_0 + h^2x''_0/2 + \ldots .$$

Hamming recommends carrying the series to the h^3 or h^4 terms. In our applications we have stopped frequently at h^2 because of the labor in computing x'''_0 and x''''_0. But note that if h/2 is used, the error is multiplied by 1/8.

The preceding analysis is readily generalized to a system of four equations:

$$x' = X(x,y,u,v,t) \qquad y' = Y(x,y,u,v,t)$$

$$u' = U(x,y,u,v,t) \qquad v' = V(x,y,u,v,t) .$$

We have

$$p_{n+1} = x_{n-1} + 2hx'_n \qquad q_{n+1} = y_{n-1} + 2hy'_n$$

$$r_{n+1} = u_{n-1} + 2hu'_n \qquad s_{n+1} = v_{n-1} + 2hv'_n$$

$$p'_{n+1} = X(p_{n+1}, q_{n+1}, r_{n+1}, s_{n+1}, \overline{n+1}\,h)$$

etc.

$$x_{n+1} = [4x_n + p_{n+1} + 2h(x'_n + p'_{n+1})]/5$$

etc.

20.4. PROGRAM NOTES

1. For each equation we need temporary storage for x_{n-1}, x_n, x'_n, p_{n+1}. Reference to the register contents on the first page of

the program listing shows how the storage is laid out. Secondary (protected) storage must of course be used. Fortunately, in indirect control the registers S0 to S9 become 10 to 19, so that f P ↔ S switching of primary and secondary registers need not be programmed.

2. Label numbers are assigned to the subroutines that compute X, Y, U, V to agree with the register numbers where x_n, y_n, u_n, v_n are stored. That is, LBL 3 ~ 3, LBL 7 ~ 7, LBL B ~ 11, LBL A ~ 15 in the indirect control mode. Hence if we are working with the equation for V, say with 15 in the I-register, then f GSB(i) sends us to the correct routine for the fourth equation of the set.

3. In evaluating the functions X, Y, U, V, the locations of x, y, u, v are assumed by the programming to be 3, 7, S1, S5. Hence in evaluating p'_{n+1}, etc., we must be careful to move x_n from 3 to 2, etc., and *then* p_{n+1} from 5 into 3, etc.

4. Extensive use of ISZ and DSZ is made to drive indirect control to the right addresses at the right time.

5. Time is incremented by RCL ϕ, STO + 1.

6. To keep track of the number of equations in the set, store m, the number of equations, in both D and E initially. On each cycle, as an equation is processed, the value in D is decreased by 1. A test tells if all equations have been processed, that the iteration is complete, that time may be incremented, and D reset to m. That is, the program may be used for any number m of equations up to 4.

<u>Example</u>. The equations for the motion of a particular hydrostatic pendulum are $d^2x/dt^2 + 3\,dy/dt + 4x = 0$, $d^2y/dt^2 - 3dx/dt + 4y = 0$, with $x_0 = 1$, $y_0 = 0$, $x'_0 = 0$, $y'_0 = 4$. You can almost smell--but not quite--that the exact solution is

$$x = \cos 4t \qquad y = \sin 4t ,$$

which is indeed the case.

The equivalent system of four equations is

$$x' = u \qquad y' = v \qquad u' = -4x - 3v \qquad v' = -4y + 3u .$$

The values for t = h are readily found by expanding, using

$$x_0 = 1 \quad y_0 = 0 \quad u_0 = 0 \quad v_0 = 4$$

$$x_0' = 0 \quad y_0' = 4 \quad u_0' = -16 \quad v_0' = 0$$

$$x_0'' = -16 \quad y_0'' = 0 \quad u_0'' = 0 \quad v_0'' = -64$$

$$x_0''' = 0 \quad y_0''' = -64 \quad u_0''' = 256 \quad v_0''' = 0$$

to get

$$x_1 = 1 - 16h^2/2 \qquad y_1 = 4h - 64h^3/6$$

$$u_1 = -16h + 256h^3/6 \qquad v_1 = 4 - 64h^2/2 \,,$$

which are recognized as the leading terms in the series for cos 4t, sin 4t, - 4 sin 4t, 4 cos 4t, respectively. For h = 0.025 we have, to 4 places,

$$x_1 = 0.9950 \quad y_1 = 0.0998 \quad u_1 = -0.3993 \quad v_1 = 3.9800 \,,$$

but these values are stored in registers 3, 7, S1, S5 as they are computed to maintain full accuracy. (Remember f P ↔ S.) Also store the initial values in 2, 6, S0, S4, store m = 4 in D and E, and store h = 0.025 in 0 and 1. Now load the program.

Next program the functions by GTO.121, switch to W/PRGM, and then

f LBL 3, f P ↔ S, RCL 1, f P ↔ S, h RTN;

f LBL 7, f P ↔ S, RCL 5, f P ↔ S, h RTN;

f LBL B, RCL 3, 4, x, CHS,f P ↔ S, RCL 5, 3, x, -,

f P ↔ S, h RTN;

g LBL a: RCL 7, 4, x, CHS,f P ↔ S, RCL 1, 3, x, +,

f P ↔ S, h RTN .

Switch to RUN and press A. In the following tabulation we will record only x and y at intervals of 2h = 0.05. The numbers beneath are the exact values, where we first shift to the radian mode by h RAD to get cos 4t and sin 4t.

t	h = 0.025 x	h = 0.05 x	h = 0.025 y	h = 0.05 y
0.05	.9801 (.9801)		.1986 (.1987)	
0.10	.9210 (.9211)	.9482	.3894 (.3894)	.3158
0.15	.8253 (.8253)	.8687	.5646 (.5646)	.4938
0.20	.6967 (.6967)	.7525	.7173 (.7174)	.6573
0.25	.5403 (.5403)	.6070	.8414 (.8415)	.7935
0.30	.3623 (.3624)	.4371	.9319 (.9320)	.8981
0.35	.1699 (.1700)	.2499	.9853 (.9854)	.9670
0.40	-.0292 (-.0292)	.0528	.9994 (.9996)	.9972

For this example and with a spacing of h = 0.025, accuracy and absolute stability are excellent, although the running time is about 24 sec per iteration and 21 sec per display of the four variables. On the other hand, a spacing of h = 0.05 caricaturizes the solution. This points up the wisdom of doing a second run with half the original interval to determine whether major changes are occurring.

AD-A054 955 RAND CORP SANTA MONICA CALIF F/G 9/2
HAND CALCULATOR PROGRAMS FOR STAFF OFFICERS. (U)
APR 78 E W PAXSON
UNCLASSIFIED RAND/R-2280-RC NL

3 OF 3
AD
A054 955

END
DATE
FILMED
7 -78
DDC

20.5 USER INSTRUCTIONS

1 20. FOURTH ORDER DIFFERENTIAL EQUATIONS 2

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	PUT EQUATIONS IN STANDARD FORM, DETERMINE x_1, y_1, u_1, v_1			
2	STO x_o IN 2, x_1 IN 3 y_o IN 6, y_1 IN 7			
3	f P←→S, STO u_o IN 0, u_1 IN 1 v_o IN 4, v_1 IN 5			
4	f P←→S, STO h IN 0 AND 1			
5	STO m IN D AND E			
6	LOAD PROGRAM			
7	GTO.121.W/PRGM. LBL 3 X LBL 7 Y LBL B U LBL a V			
8	SWITCH TO RUN. PRESS A			
9	SEE t AS h PAUSE			
	SEE x_n, y_n, u_n, v_n FOR 5 SECONDS EACH			
10	USE R/S IF MORE TIME NEEDED TO RECORD			

20.6 FOURTH ORDER DIFFERENTIAL EQUATIONS

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
001	001 *LBLA	21 11			057 RCLi	36 45	x_n
	002 3	03			058 4	04	
	003 STOI	35 46			059 x	-35	
	004 GTOC	22 13		060	060 +	-55	
	005 *LBLC	21 13			061 5	05	
	006 GSBi	23 45	X		062 ÷	-24	x_{n+1}
	007 ISZI	16 26 46	4		063 ISZI	16 26 46	
	008 STOi	35 45	x^1_n		064 ISZI	16 26 46	
	009 RCL0	36 00			065 ISZI	16 26 46	5
010	010 x	-35			066 STOi	35 45	x_{n+1} IN R_5
	011 2	02			067 RCLD	36 14	
	012 x	-35	$2h\ x^1_n$		068 1	01	
	013 DSZI	16 25 46	3		069 –	-45	
	014 DSZI	16 25 46	2	070	070 STOD	35 14	
	015 RCLi	36 45	x_{n-1}		071 X=0?	16-43	m EQNS ?
	016 +	-55			072 GTO4	22 04	OUTPUT
	017 ISZI	16 26 46	3		073 ISZI	16 26 46	
	018 RCLi	36 45	x_n		074 ISZI	16 26 46	7
	019 DSZI	16 25 46	2		075 GTO1	22 01	$y_{n+1}, u_{n+1}, v_{n+1}$
020	020 STOi	35 45	x_n		076 *LBL4	21 04	
	021 R↓	-31	TO GET p_{n+1}		077 RCLE	36 15	
	022 ISZI	16 26 46	3		078 STOD	35 14	
	023 ISZI	16 26 46	4		079 GSB2	23 02	x_{n+1} FROM R_5 TO R_3
	024 ISZI	16 26 46	5	080	080 RCL1	36 01	
	025 STOi	35 45	p_{n+1}		081 PSE	16 51	TIME SHOWN
	026 RCLD	36 14			082 RCL3	36 03	
	027 1	01			083 PRTX	-14	f-x- SEE x_n
	028 –	-45	m - 1		084 RCLD	36 14	
	029 STOD	35 14			085 1	01	
030	030 X=0?	16-43	m EQNS ?		086 –	-45	
	031 GTO0	22 00			087 X=0?	16-43	m EQNS ?
	032 ISZI	16 26 46	6		088 GTOA	22 11	
	033 ISZI	16 26 46	7		089 RCL7	36 07	
	034 GTOC	22 13	Y, U, V IN ORDER	090	090 PRTX	-14	SEE y_n
	035 *LBL0	21 00			091 RCLD	36 14	
	036 RCLE	36 15			092 2	02	
	037 STOD	35 14			093 –	-45	
	038 RCL0	36 00			094 X=0?	16-43	m EQNS ?
	039 ST+1	35-55 01	t + h		095 GTOA	22 11	
040	040 GSB2	23 02			096 P⇄S	16-51	
	041 3	03			097 RCL1	36 01	SEE u_n
	042 STOI	35 46	3		098 PRTX	-14	
	043 GTO1	22 01			099 RCLD	36 14	
	044 *LBL1	21 01		100	100 3	03	
	045 GSBi	23 45	p^1_{n+1}		101 –	-45	
	046 ISZI	16 26 46	4		102 X=0?	16-43	m EQNS ?
	047 RCLi	36 45	x^1_n		103 GTO5	22 05	
	048 +	-55			104 RCL5	36 05	
	049 RCL0	36 00			105 PRTX	-14	SEE v_n
050	050 x	-35			106 GTO5	22 05	
	051 2	02			107 *LBL5	21 05	
	052 x	-35			108 P⇄S	16-51	
	053 DSZI	16 25 46	3		109 GTOA	22 11	NEXT ITERATION
	054 RCLi	36 45	p_{n+1}	110	110 *LBL2	21 02	
	055 +	-55			111 P⇄S	16-51	
	056 DSZI	16 25 46	2		112 RCL7	36 07	S_{n+1}

REGISTERS

0	1	2	3	4	5	6	7	8	9
h	t	x_{n-1}	x_n	x^1_n	p_{n+1}	y_{n-1}	y_n	y^1_n	q_{n+1}
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
u_{n-1}	u_n	u^1_n	r_{n+1}	v_{n-1}	v_n	v^1_n	S_{n+1}		

A	B	C	D	E	I
			m	m	

20.6 PROGRAM LISTING

STEP	KEY ENTRY	KEY CODE	COMMENTS
	113 ST05	35 05	
	114 RCL3	36 03	r_{r+1}
	115 ST01	35 01	
	116 P⇄S	16-51	
	117 RCL9	36 09	q_{n+1}
	118 ST07	35 07	
	119 RCL5	36 05	p_{n+1}
120	120 ST03	35 03	READY FOR p^1_{n+1}
	121 RTN	24	
130			
140			
150			
160			
170			
180			
190			
200			
210			
220			

LABELS					FLAGS
A USED	B U	C USED	D	E	0
a V	b	c	d	e	1
0 USED	1 USED	2 USED	3 X	4 USED	2
5 USED	6	7 Y	8	9	3

SET STATUS

FLAGS (ON OFF)	TRIG	DISP
0 ☐ ☐	DEG ☐	FIX ☐
1 ☐ ☐	GRAD ☐	SCI ☐
2 ☐ ☐	RAD ☐	ENG ☐
3 ☐ ☐		n____

21. CURVE FAMILIES AND MACH NUMBERS

21.1. REFERENCES

a. United States Air Force, *Flight Manual A-7D Aircraft*, T.O. 1A-7D-1S-32, 29 March 1971.

b. United States Air Force, AFSC, *Space Planners Guide*, 1 July 1965 (For Official Use Only).

c. G. H. Kaplan, L. E. Doggett, and P. K. Seidelmann, *Almanac for Computers, 1977*, United States Naval Observatory, Circular No. 155, 1 October 1976.

21.2. DISCUSSION

Military data for analytic or operational use are frequently presented as a family of curves $z = f(x,y)$, where y is the parameter naming the family's members. For example, Fig. 21.1 (taken from Ref. a) is a nomogram to determine, for the A-7D aircraft, the Mach number to maximize range for constant altitude cruise, given average gross weight and drag index.* To use the nomogram, start with the average gross weight on the upper left scale. Move horizontally to the pressure altitude. Drop vertically to the appropriate drag index curve. Move horizontally to the left to read the true Mach number for long-range cruise. Page after page of such nomograms appear in the mission-planning appendix of Ref. a. Similarly, page after page of similar nomograms appear in Ref. b, to be used in planning space missions.

Neither Ref. a nor Ref. b gives the equations of the curve families. When these equations can be found, and if they are relatively simple, they can be readily programmed. Frequently, however, as in the case of an ephemeris or almanac that tabulates the coordinates of celestial bodies for astronomical and navigational use, the underlying equations are extremely complex. To quote from Ref. c, these

---

*The drag index is not a drag coefficient. Its determination, as explained in detail in Ref. a, is a tabulation of the drag contributions of external stores by type and station. The clean aircraft has a drag index of 0.

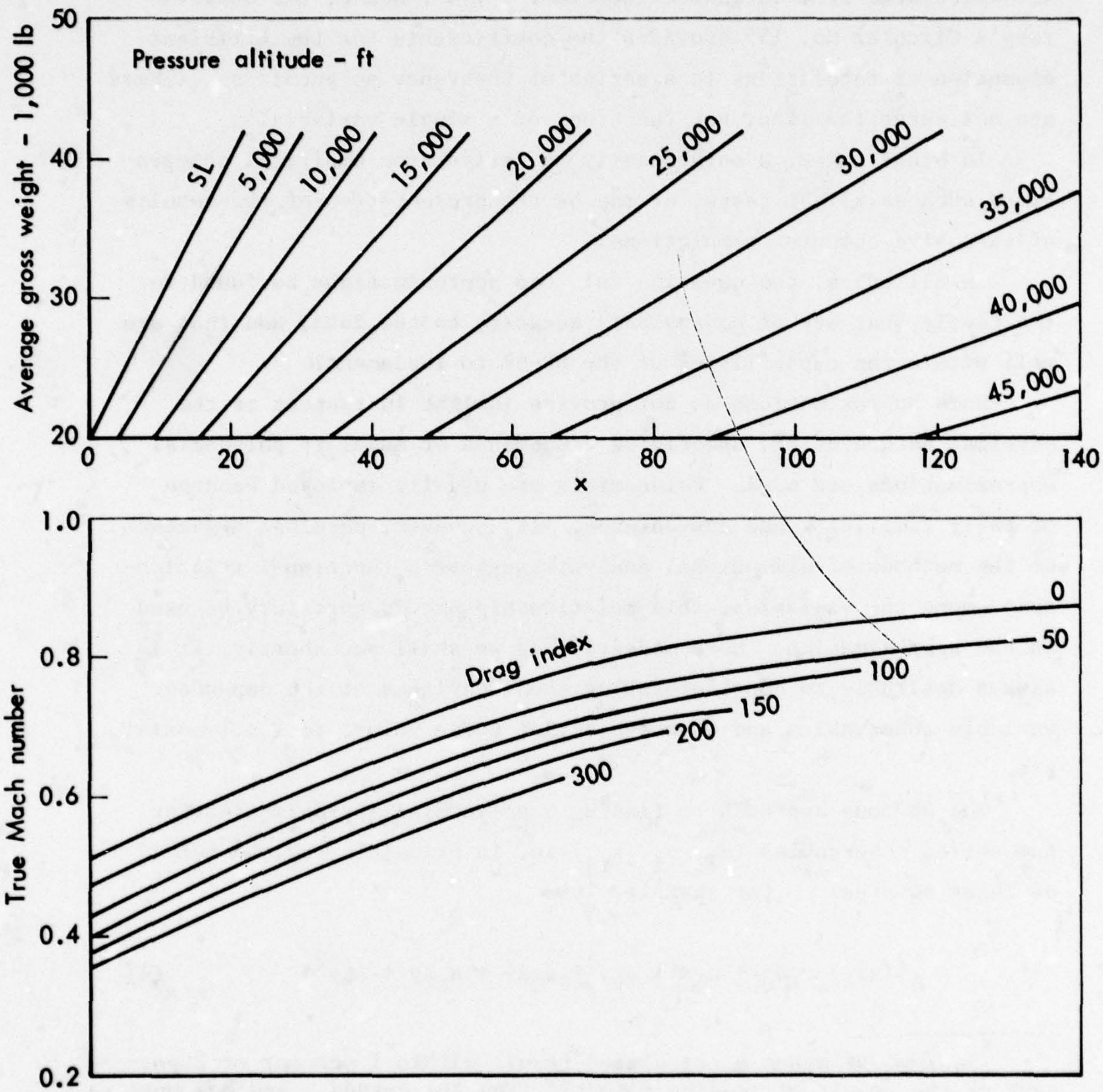


Fig. 21.1— Optimum Mach number

tabulations "should ideally be replaced by concise mathematical expressions for direct calculations. Such expressions must take the form of mathematical approximations, however, since the precise data . . . are calculated from extensive theories. . . ." Hence, the Observatory's Circular No. 155 provides the coefficients for the efficient expansion of tabulations in a series of Chebyshev polynomials. (These are *not* curve families, but functions of a single variable.)

In other cases, a curve family may arise from empirical observations such as flight tests, or may be the presentation of the results of extensive computer simulations.

In all cases, the question is: Can approximations be found for the family that are of *equivalent** accuracy to the data, and that are well within the capabilities of the HP-67 to implement?

Such approximations do not provide insight in respect to the physical nature of the underlying phenomenon or model if polynomial approximations are used. Polynomials are usually employed because of their simplicity and convenience. If, however, physical arguments or the methods of dimensional analysis suggest a functional relationship among the variables, this relationship should certainly be used in the approximation. More modestly, as we shall see shortly, it is always desirable to consider taking the logarithms of the dependent variable observables and then subjecting these values to a polynomial fit.

The obvious approach to finding a polynomial approximation for the set of observables (x_i, y_i, z_{ij}) is, in principle, by the method of least squares.** For example, let

$$f(x,y) = a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 \qquad (1)$$

**Equivalent* means a calculated result within 1 percent or 2 percent of the result of reading directly from the curves. And one must keep in mind that (a) the width of plotted curves and the way in which they are drawn can introduce discrepancies of this order in the basic readings of values on which to base an approximation, and (b) the accuracy of the presented data is usually unknown.

**In more modern guise, by orthonormalizing codes. See John Todd (ed.), *Survey of Numerical Analysis*, Chap. 10, McGraw-Hill, New York, 1962.

be the predictor. Then choose the coefficients a_i to minimize

$$G = \sum_{i,j} \left[z_{ij} - f(x_i, y_i) \right]^2 . \qquad (2)$$

Take the six partial derivatives of G with respect to the coefficients and equate each to zero. Form the indicated sums and get six linear algebraic equations for the coefficients $a_0, \ldots, a_5$.*

This is principle. In practice we have two problems:

1. Even this "best" approximation by a second-order polynomial may be of unacceptable accuracy.
2. No method in two dimensions is known that is equivalent to the use of Chebyshev polynomials in one-dimensional fitting which gives the "best" (most economical) fit for a polynomial of given order.

Frankly experimental methods are used to get an acceptable fit for the curve families encountered. As a working tool, the efficient Chebyshev approximation program available as Program 14 in the HP-67/HP-67 Stat Pac 1 is exploited. It is also advisable to stare hard and long at the graphs of the particular family to get ideas from the geometry (one version of the "low cunning" approach to ad hoc computing). Generalization to your problems of the methods employed in the examples below cannot be guaranteed.

21.3. TRUE MACH NUMBER

Staring at the upper set of straight-line segments in Fig. 21.1 gives the image that they might all originate as a sheaf from a common origin. Use of a straightedge shows that this surmise is at least approximately true. Hence, try as the functional form for the fit, the relation

---

*A program for the solution of 6 equations in 6 unknowns is given in the HP-67/HP-97 Users Library Solutions book "High-Level Math," programmed by R. E. DeBolt.

$$\dot{x} = m(H) \cdot G - 20 ,$$

where m(H) is the slope for altitude H in kft, G is the average gross weight in klb, and x is the dummy variable for the nomogram. We now build the following table:

(1)	(2)	(3)	(4)	(5)	(6)	(7)
H	m(H)	Δ	ℓn m(H)	Δ	exp (quad)	$a_0 = 0$
0	1.000		0		1.000	1.000
		0.198		0.1807		
5	1.198		0.1807		1.196	1.196
		0.251		0.1902		
10	1.449		0.3709		1.442	1.442
		0.317		0.1978		
15	1.766		0.5687		1.752	1.753
		0.348		0.1799		
20	2.114		0.7486		2.146	2.147
		0.455		0.1949		
25	2.569		0.9435		2.649	2.649
		0.772		0.2628		
30	3.341		1.2063		3.295	3.296
		0.892		0.2366		
35	4.233		1.4429		4.130	4.131
		1.086		0.2220		
40	5.309		1.6649		5.219	5.220
		1.206		0.2092		
45	6.515		1.8741		6.645	6.646

Column (2) is obtained by reading differences from the figure and dividing. Considerable noise can be introduced by this process. (I used an 8X loupe. It is probably better to plot the values to a large scale and fair a curve through the points, and then read off values.) The increasing first differences of column (3) suggest that an exponential form be used since the exponent will be much flatter. This is shown by the differences of column (5).

A quadratic fit to ℓn m(H) should be good. We find from the HP-67 Stat Pac 1 program (six minutes to run) that

$$\ell n\,\overline{m}(H) = -2.04 \times 10^{-4} + 0.0351H + 1.56 \times 10^{-4}\,H^2 . \quad (3)$$

A comparison of columns (1) and (6) which is $\bar{m}(H)$ shows the adequacy of the fit. Column (7) shows that the constant term may safely be put equal to 0.

The lower family of curves in Fig. 21.1 requires a true two-dimensional fit. This family is well behaved in that a second-order polynomial fit looks promising (Eq. (1)). Moreover, try the fit with the coefficient a_4 of the cross term xy put equal to 0, because the curves are so nearly parallel. That is, $\partial f/\partial x = a_1 + 2a_3x + a_4y$ and the dependence on y is weak.

Proceed as follows.

1. Read from the curves the values of M at 35 points, using increments of 20 for x and 50 for y (the drag index). Do not use y = 300, reserving it for an extrapolation check.

2. Use the HP Stat Pac 1 Program 14 to get the following direct and cross-fits. (The second-order Chebyshev fit is used and the time per case is well under 10 min.)

$$
\begin{aligned}
f(x,0) &= 0.530 + 0.00500x - 1.8155 \times 10^{-5} x^2 \\
f(x,50) &= 0.477 + 0.00527x - 2.054 \times 10^{-5} x^2 \\
f(x,100) &= 0.430 + 0.00539x - 1.987 \times 10^{-5} x^2 \\
f(x,150) &= 0.398 + 0.00571x - 2.232 \times 10^{-5} x^2 \\
f(x,200) &= 0.378 + 0.00543x - 1.875 \times 10^{-5} x^2 \\
f(0,y) &= 0.531 - 0.00122y + 2.286 \times 10^{-6} y^2 \\
f(20,y) &= 0.611 - 0.00091y + 1.143 \times 10^{-6} y^2 \\
f(40,y) &= 0.690 - 0.00085y + 1.143 \times 10^{-6} y^2 \\
f(60,y) &= 0.771 - 0.00102y + 1.857 \times 10^{-6} y^2 \\
f(80,y) &= 0.809 - 0.00090y + 1.571 \times 10^{-6} y^2
\end{aligned}
$$

3. The near constancy of the coefficients is promising. Using simply their means,

$$
\begin{aligned}
f(x,y) &= f(0,y) + 0.00536x - 1.993 \times 10^{-5} x^2 \\
f(x,y) &= f(x,0) - 0.00098y + 1.600 \times 10^{-6} y^2 .
\end{aligned}
$$

From these

$$f(x,y) = f(0,0) + 0.00536x - 1.993 \times 10^{-5} x^2$$
$$- 0.00098y + 1.600 \times 10^{-6} y^2 ,$$

where $f(0,0) = 0.53$.

4. Programming the last expression for $f(x,y)$ and checking its output against the 35 observed values yields a good fit. But using the HP improves one's "nose for numbers." There are some systematic biases in the fit. The dependence on x is somewhat strong, and the dependence on y can be weakened slightly. Adjusting to

$$f(x,y) = 0.53 + 0.0052x - 2 \times 10^{-5} x^2 - 0.001y + 1.5 \times 10^{-6} y^2 , \quad (4)$$

the mean absolute error with respect to the 35 observations is 0.0076 and the maximum error is 0.016.

5. It is now trivial to program Eqs. (3) and (4). The eight constants can be stored in primary registers 0 through 7 and recorded via f W/DATA on side two of the program card. Running time is three seconds. The output over the entire nomogram agrees to ±0.01.

22. TEN-POINT GAUSSIAN INTEGRATION

22.1. REFERENCE

a. M. Abramowitz and I. A. Stegun (eds.), *Handbook of Mathematical Functions*, National Bureau of Standards Applied Mathematics Series 55, U.S. Department of Commerce, 3d Printing, March 1965.

22.2. DISCUSSION

This section gives a utility program to evaluate definite integrals with high accuracy. If the integrand becomes infinite at some point within the limits of integration, divide the integral into two parts, using as limits values slightly less and greater than that point. Accuracy is checked by varying these values and reevaluating. In fact, even if the integrand does not exhibit this behavior, accuracy can be checked by dividing the interval into two or more parts.

22.3. EQUATIONS

Gauss's formula for an arbitrary interval is (Ref. a, p. 887, 25.4.30):

$$\int_a^b f(y)\,dy \doteq \frac{b-a}{2} \sum_{i=0}^{4} w_i \left\{ f\left(\frac{b-a}{2} x_i + \frac{b+a}{2}\right) + f\left(-\frac{b-a}{2} x_i + \frac{b+a}{2}\right) \right\}. \tag{1}$$

The abscissae x_i are the zeros of the Legendre orthogonal polynomial $P_5(x)$. The weights w_i are given by a formula involving $P_5'(x)$.

The values are

$$\begin{aligned} x_0 &= 0.148874339 & w_0 &= 0.295524225 \\ x_1 &= 0.433395394 & w_1 &= 0.269266719 \\ x_2 &= 0.679409568 & w_2 &= 0.219086363 \\ x_3 &= 0.865063367 & w_3 &= 0.149451349 \\ x_4 &= 0.973906529 & w_4 &= 0.066671344 \end{aligned}$$

These are keyed into a data card with x_i stored in primary registers 0 to 4, and w_i in the secondary registers.

22.4. PROGRAM NOTES

The program is straightforward and has no features of interest.

PROBLEM

Write a program to evaluate elliptic integrals of the first kind:

$$F(\phi,\alpha) = \int_0^{\phi} (1 - \sin^2 \alpha \cdot \sin^2 \theta)^{-1/2} \, d\theta \ .$$

(1) Load data and program cards. The integral will be evaluated in the radian (h RAD) mode, but ϕ and α are usually given in degrees.

(2) STO 0 in A and ϕ in degrees in B.
STO α in degrees in E.

(3) GTO A and switch to W/PRGM. Key in the steps:

h RAD, RCL B, g → R, STO B, RCL E, g → R, f sin, g x^2 STO E.

(4) Switch to RUN, GTO E, switch to W/PRGM, and key in the steps:

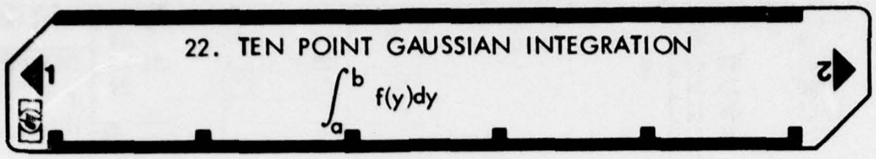
f sin, g x^2, RCL E, x, CHS, 1, +, f$\sqrt{x}$, h 1/x.

(5) Switch to RUN and press A. DSP 8.

$\phi = 30°$, $\alpha = 40°$, F(30,40) = 0.533 427 45
$\phi = 65°$, $\alpha = 60°$, F(65,60) = 1.348 926 43.

These agree exactly with the tabular entries (Table 17.5) of Ref. a. Running time is about 30 sec.

22.5 USER INSTRUCTIONS



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	LOAD BOTH SIDES OF DATA CARD			
2	LOAD PRGM CARD. GTO E.			
3	SWITCH TO W/PRGM DEFINE f(y), y IS STORED IN 9 BY THE PROGRAM, h RTN NOT NEEDED			
4	SWITCH TO RUN			
5	a STORE IN A, b STORE IN B			
6	PRESS A			
7	TO CHECK ACCURACT, DIVIDE INTERVAL INTO TWO OR MORE PARTS. DO FOR EACH PART AND ADD.			
	EXAMPLE: f(y) = 1/y STEP .058, h 1/y			
	a = 1 STO A	1		
	a = 5 STO B	5		
	PRESS A. SEE 1.609437902			
	$\int_1^5 dy/y = \ln 5$			
	ln 5 = 1.609437912			

22.6 TEN POINT GAUSSIAN INTEGRATION

STEP	KEY ENTRY		KEY CODE	COMMENTS
001	001	*LBLA	21 11	INITIALIZE FOR
	002	0	00	INDIRECT ADDRESSING
	003	STOI	35 46	
	004	STOD	35 14	
	005	RCLB	36 12	
	006	RCLA	36 11	
	007	-	-45	
	008	2	02	
	009	÷	-24	
010	010	STOC	35 13	(b-a)/2
	011	RCLB	36 12	
	012	RCLA	36 11	
	013	+	-55	
	014	2	02	
	015	÷	-24	
	016	STOB	35 12	(b+a)/2
	017	RCLC	36 13	
	018	STOA	35 11	(b-a)/2
	019	*LBLB	21 12	
020	020	RCLi	36 45	x_i
	021	RCLA	36 11	
	022	×	-35	
	023	RCLB	36 12	
	024	+	-55	
	025	STO9	35 09	
	026	GSBE	23 15	$f(y_i)$
	027	P⇄S	16-51	
	028	RCLi	36 45	w_i
	029	P⇄S	16-51	
030	030	×	-35	
	031	STOC	35 13	$w_i f(y_i)$
	032	RCLi	36 45	
	033	CHS	-22	$-x_i$
	034	RCLA	36 11	
	035	×	-35	
	036	RCLB	36 12	
	037	+	-55	
	038	STO9	35 09	
	039	GSBE	23 15	NEXT f
040	040	P⇄S	16-51	w_i
	041	RCLi	36 45	
	042	P⇄S	16-51	
	043	×	-35	
	044	RCLC	36 13	
	045	+	-55	
	046	RCLD	36 14	
	047	+	-55	PARTIAL SUM
	048	STOD	35 14	
	049	4	04	
050	050	ISZI	16 26 46	
	051	RCLI	36 46	
	052	X≤Y?	16-35	$i \leq 4$?
	053	GTOB	22 12	LOOP
	054	RCLD	36 14	
	055	RCLA	36 11	
	056	×	-35	
	057	STOD	35 14	INTEGRAL
	058	RTN	24	
	059	*LBLE	21 15	DEFINE f(y)
060	060	RTN	24	y IS STORED IN R_9
070				

PRI

x_i	0.148874339	0
	0.433395394	1
	0.679409568	2
	0.865063367	3
	0.973906529	4
	0.000000000	5
	0.000000000	6
	0.000000000	7
	0.000000000	8
	0.000000000	9
	0.000000000	A
	0.000000000	B
	0.000000000	C
	0.000000000	D
	0.000000000	E
	0.000000000	I

SEC

w_i	0.295524225	0
	0.269266719	1
	0.219086363	2
	0.149451349	3
	0.066671344	4
	0.000000000	5
	0.000000000	6
	0.000000000	7
	0.000000000	8
	0.000000000	9
	0.000000000	A
	0.000000000	B
	0.000000000	C
	0.000000000	D
	0.000000000	E
	0.000000000	I

REGISTERS

0	1	2	3	4	5	6	7	8	9
									y
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I
a	b	$w_i f(y_i)$	INTEGRAL		

23. TRUTH TABLES

23.1. REFERENCES

a. John E. Pfeiffer, "Symbolic Logic," *Scientific American*, December 1950, pp. 22-24.

b. E. C. Berkeley, *Giant Brains*, John Wiley & Sons, New York, 1949.

c. Walter E. Cushen, "Symbolic Logic in Operations Research," in *Operations Research for Management*, J. F. McCloskey and F. N. Trefethen (eds.), The Johns Hopkins Press, Baltimore, 1954.

d. R. M. Smidt and I. L. Reis, "Symbolic Logic and Plant Location," *Journal of Industrial Engineering*, Vol. 14, No. 1, January-February 1963, pp. 18-21.

e. P. F. Strawson, *Introduction to Logical Theory*, Methuen & Co. Ltd., London, John Wiley & Sons, New York, 1952.

f. H. M. Semarne, "Symbolic Logic in Language Engineering," *Proceedings of the Western Joint Computer Conference*, May 3-5, 1960, pp. 61-71.

23.2. DISCUSSION

This section describes a calculus of propositions based on a binary algebra for logical connections and operations particularly suited for calculator implementation as well as easy algebraic manipulation.

Letters of the alphabet will stand for propositions. To illustrate, choosing propositions to be used in an example to follow, let

a stand for "A man is a mathematician."

b stand for "A man likes whisky at night."

c stand for "A man likes Mozart in the morning."

d stand for "A man waits 20 minutes for a bus."

In the two-valued calculus, a proposition has a truth value of 0 (false) or 1 (true), $a = 0$ or $a = 1$. For example, writing $a = 1$ means "A man is a mathematician."

Propositions may be operated on or connected by logical operations. A truth table corresponds to each such operation. Ordinary language will be used instead of special symbols. For each basic logical operation, a truth table shows the truth value of the operation for all combinations of truth values of its component propositions. Such basic truth tables can be summarized in binary algebraic form. This is verified in the right-hand column of the tabulation below.

Operation		Truth Table				Binary Algebraic Form
NOT a	a	0	1			1 - a
	NOT a	1	0			
a AND b	a	0	1	0	1	ab
	b	0	0	1	1	
	a AND b	0	0	0	1	
a AND/OR b	a	0	1	0	1	a + b - ab
	b	0	0	1	1	
	a AND/OR b	0	1	1	1	
a OR ELSE b	a	0	1	0	1	a + b - 2ab
	b	0	0	1	1	
	a OR ELSE b	0	1	1	0	
IF a THEN b	a	0	1	0	1	1 - a + ab
	b	0	0	1	1	
	IF a THEN b	1	0	1	1	
(Note that a false proposition can imply either truth or falseness. Also observe that 'NOT (IF a THEN b)' is 'a AND NOT b'.)						
NOT BOTH a AND b	a	0	1	0	1	1 - ab
	b	0	0	1	1	
	NOT BOTH a AND b	1	1	1	0	
(This is the negation of AND.)						
NEITHER a NOR b	a	0	1	0	1	1 - a - b + ab
	b	0	0	1	1	
	NEITHER a NOR b	1	0	0	0	
(This is the negation of AND/OR.)						
a LIKE b	a	0	1	0	1	1 - a - b + 2ab
	b	0	0	1	1	
	a LIKE b	1	0	0	1	
(This is equivalence and is the negation of OR ELSE.)						

Binary algebra has some interesting features not found in ordinary algebra. Since the variables can take on only the values 0 and 1, $a^2 = a$ and $ab(1 - a + ab) = ab - ab + ab = ab$, which is readily checked by the truth table:

a	0	1	0	1
b	0	0	1	1
ab	0	0	0	1
1 - a + ab	1	0	1	1
ab(1 - a + ab)	0	0	0	1 .

On the other hand, a proposition appearing on both sides of an = sign cannot be cancelled, because this could be division by 0.

The use and manipulation of these binary algebraic functions can be illustrated by a problem that Walter Pitts of the Massachusetts Institute of Technology set long ago on an examination (Ref. a). Suppose 1 through 4 below are known to be true:

1. *If a mathematician does not have to wait 20 minutes for a bus, then he either likes Mozart in the morning or whisky at night, but not both.*
2. *If a man likes whisky at night, then he either likes Mozart in the morning and does not have to wait 20 minutes for a bus or he does not like Mozart in the morning and has to wait 20 minutes for a bus or else he is no mathematician.*
3. *If a man likes Mozart in the morning and does not have to wait 20 minutes for a bus, then he likes whisky at night.*
4. *If a mathematician likes Mozart in the morning, he either likes whisky at night or has to wait 20 minutes for a bus; conversely, if he likes whisky at night and has to wait 20 minutes for a bus, he is a mathematician--if he likes Mozart in the morning.*

Then:

When does a mathematician wait 20 minutes for a bus?

To solve this problem, first translate each condition from English to the language of propositions and logical connectives,* and then express the result in binary algebra. Because each of the four above conditions is true, each algebraic expression is put equal to 1 and then simplified.

1. IF(a AND NOT d) THEN(b OR ELSE c).

$$1 - a(1 - d) + a(1 - d)(b + c - 2bc) = 1 ,$$

$$a(1 - d)(b + c - 2bc - 1) = 0 . \qquad (1)$$

2. IF b THEN(((c AND NOT d) OR ELSE(NOT c AND d)) OR ELSE(NOT a)).

To simplify, set A = (c AND NOT d) OR ELSE(NOT c AND d).

Then $A = c(1 - d) + d(1 - c) - 2cd(1 - c)(1 - d)$

$= c + d - 2cd$.

That is, the proposition A is equivalent to the proposition "c OR ELSE d". Making this substitution, we obtain

$$1 - b + b(c + d - 2cd + 1 - a$$

$$-2(1 - a)(c + d - 2cd) = 1,$$

$$b(-a + (c + d - 2cd)(2a - 1)) = 0 . \qquad (2)$$

3. IF(c AND NOT d) THEN b.

$$1 - c(1 - d) + cb(1 - d) = 1 ,$$

$$c(1 - d)(1 - b) = 0 . \qquad (3)$$

4a. IF(a AND c) THEN(b OR ELSE d).

$$1 - ac + ac(b + d - 2bd) = 1 ,$$

$$ac(b + d - 2bd - 1) = 0 . \qquad (4)$$

---

*Ref. e is helpful in this respect.

4b. IF c THEN(IF (b AND d) THEN a).

$$1 - c + c(1 - bd + abd) = 1 \ ,$$

$$bcd(1 - a) = 0 \ . \qquad (5)$$

To answer the question, put a = 1 and d = 1 in (1) through (5), since these two propositions must be true. The interpretation of the question is delicate. The question is rephrased here as: What values of b and c are associated with a = 1 and d = 1 to make all conditions of the problem true? The set of conditions reduces to

$$bc = 0 \ .$$

This means NOT BOTH b AND c (that is, 1 - bc = 1), which may be expressed

A. When he likes neither Mozart in the morning nor whisky at night.

B. When he likes whisky at night and not Mozart in the morning.

C. When he likes Mozart in the morning and not whisky at night.*

Of course, many questions other than the one above could be asked. To be exhaustive, this means: Find all values for the set of propositions (a,b,c,d) that satisfy all given conditions. The program of this section is designed to examine systematically all cases, here 16 in number. Programming the conditions and running the program yields only eight satisfactory combinations of (a,b,c,d).

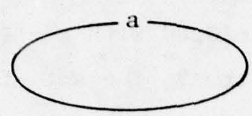
	a	b	c	d
	0	0	0	0
	0	1	0	0
	0	0	1	0
	0	0	0	1
→	1	0	0	1
→	1	1	0	1
	0	0	1	1
→	1	0	1	1 .

*Something of a furor erupted in the Letters section of the *Scientific American* of February 1951 because Pfeiffer had not given an answer in his article. Pfeiffer and Morris gave the first part of C as the answer. Krause gave the first part of B, and Bomgren gave answer A.

This is the truth table method. The lines indicated by arrows are the answers to the given problem.

The logical structure of the problem may also be captured by using the truth table to construct a Venn diagram.

Suppose a point placed on this page represents a man. Group together all men who are mathematicians so that this subset is enclosed:

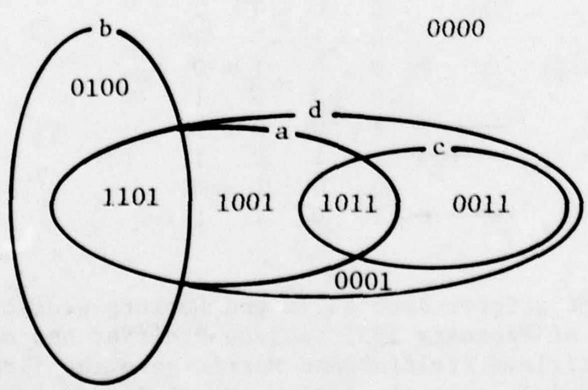


then all inside men have the value a = 1, those outside the value a = 0.

From the truth table, whenever a = 1, d = 1. But there are cases where a = 0 and d = 1. Hence the *set* a is properly contained in the *set* d. Logically this means "if a then d", and algebraically "a = ad". Similarly, c is contained in d. But for a and c, by the last four lines of the truth table, there are the combinations 10, 01, 11. Hence a and c have points in common--they intersect.

Turning to the set b, since bc = 0 in all cases, the sets b and c are disjoint--they do not intersect. But b intersects a and also intersects d. However, because of the line 1101, and because there is no line 0101, these latter two intersections are the same subset.

Putting all of this together yields the following diagram:



Returning to the original algebraic formulation, it is readily seen that all conditions are satisfied by appeal to this diagram. For example, since bc = 0 and abd = ab, in (2) above ab = bd.

The original question asked by this problem is not clear. It would be better to ask, *What can be said about a mathematician?*

The answer to this question is:

- He always waits 20 minutes for a bus;
- He does not like both whisky at night and Mozart in the morning, although he may like one or the other.

The calculus of propositions, a branch of symbolic logic, has found applications in optimizing switching circuit design, in determining insurance eligibility (an example from Ref. b will be given in the next section), in deciding on plant location (Ref. d), and in the interpretation of contracts and law. Walter Cushen, in a fascinating chapter in Ref. c, discusses applications to production engineering and to conflicts formulated as multi-move games.

23.3. EQUATIONS

None.

23.4. PROGRAM NOTES

Since the program flow is somewhat complex, a flowchart is provided.

Suppose there are n propositions. Then there are 2^n cases to examine, which are actually numbered 0, 1, ..., $2^n - 1$. Each case number in its turn is reduced to its binary form, but stored backwards in R_1 to R_n. Indirect addressing is used.

Each condition is examined in turn, the examination halting at the first condition that is not satisfied. If all conditions are satisfied, the binary case number is displayed. R/S continues to the next valid case. The end is signalled by the decimal display of $2^n - 1$.

As programmed, there is space for 7 propositions. If a problem requires more than 7, make some slight programming changes. Primary registers 8 and 9 are shifted to secondary 8 and 9. Then up to 18 propositions can in principle be handled. For large problems, however, there is apt to be a large number of conditions and there may not be adequate space available to program them. Moreover, execution time will be long. It is wise to do as much algebraic manipulation on the set of conditions as is feasible to simplify the set.

In problems where some propositions are held constant, the constant value(s) (0 or 1) are stored manually in some register(s) above R_n, but only if these are needed in programming the conditions.

EXAMPLE

This is a group insurance problem taken from Ref. b (pp. 161-165). The rules (conditions) applying to employees are:

1. Any employee, to be insured, must be eligible for insurance, must make application for insurance, and must have such application for insurance approved.
2. Only eligible employees may apply for insurance.
3. The application of any person eligible for insurance without medical examination is automatically approved.
4. (Naturally) an application can be approved only if the application is made.
5. (Naturally) a medical examination will not be required from any person not eligible for insurance.

The propositions are 5 questions about an employee to be answered "yes" (1) or "no" (0). These are:

a: Is the employee eligible for insurance?
b: Has the employee applied for insurance?
c: Has the employee's application for insurance been approved?
d: Does the employee require a medical examination for insurance?
e: Is the employee insured?

The conditions are translated:

1. IF e THEN(a AND b AND c)

$$e(1 - abc) = 0 .$$

2. IF b THEN a

$$b(1 - a) = 0 .$$

3. IF a AND b AND NOT d THEN c

$$ab(1 - d)(1 - c) - 0 .$$

4. IF c THEN b

$$c(1 - b) = 0 .$$

5. IF NOT a THEN NOT d

$$d(1 - a) = 0 .$$

The question is "What are the possible statuses of employees who are not insured?" This means that e must be put equal to 0. But then the first condition is irrelevant since a false proposition implies any proposition.

SOLUTION

Load program. Key GTO B. Switch to W/PRGM. Now key in the condition in the order 2, 4, 5, 3. That is, the simplest conditions are entered first to save execution time. The program steps are:

072	1	079	1	086	1	093	1	RCL 1
	RCL 1		RCL 2		RCL 1		RCL 4	x
	-		-		-		-	RCL 2
	RCL 2		RCL 3		RCL 4		1	x
	x		x		x		RCL 3	h RTN
	f x ≠ 0		f x ≠ 0		f x ≠ 0		-	
	h RTN		h RTN		h RTN		x .	

(Note that if conditions 2 to 5 were multiplied,

$$abcd(1 - a)(1 - b)(1 - c)(1 - d) = 0 ,$$

which is true for *all* cases. The multiplication has destroyed the meaning of the individual conditions by absorption.)

Now switch to RUN. Key 4 (the number of propositions) and Press A. See 0. This is actually the status 0000 for a,b,c,d. On successive presses of R/S you will see 1000, 1110, 1001, 1101, 1111, 15 (the number of cases minus 1).

Using the definitions of a,b,c,d, these 5 statuses rapidly translate into an answer to the question.

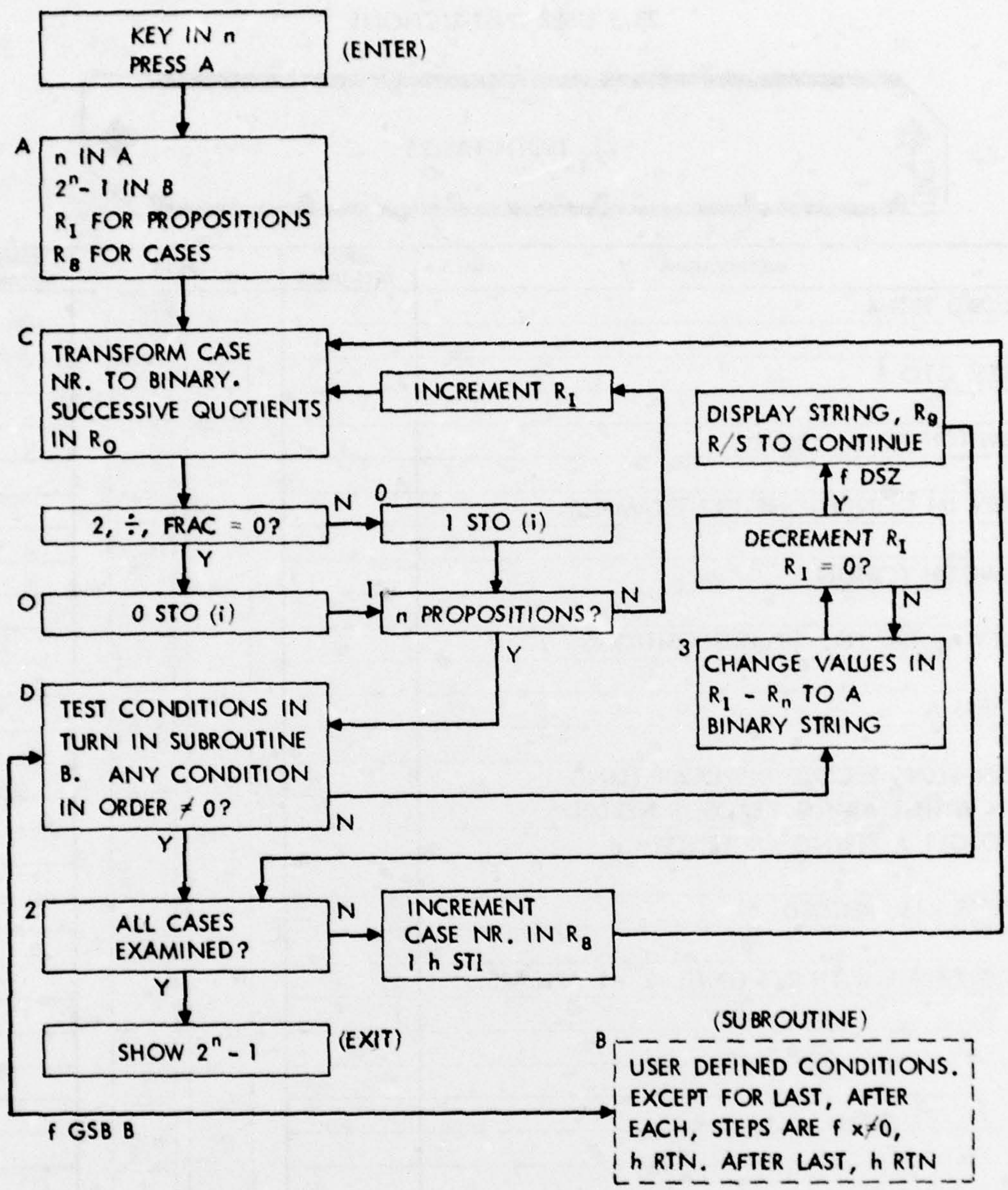


Fig. 23.1—Truth table program flowchart

23.5 USER INSTRUCTIONS

23. TRUTH TABLES

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	LOAD PRGM			
2	KEY GTO B			
3	SWITCH TO W/ PRGM			
4	KEY IN CONDITIONS (SEE EXAMPLE)			
5	SWITCH TO RUN			
6	KEY n, THE NR. OF PROPOSITIONS			
7	PRESS A			
8	ON STOP, RECORD DISPLAY FILLING IN WITH LEADING ZEROS IF NEEDED TO GET A STRING OF LENGTH n			
9	PRESS R/S. RECORD.			
10	CONTINUE WITH R/S UNTIL 2^n-1 APPEARS.			

23.6 TRUTH TABLES

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	
002	STOA	35 11	n
003	2	02	
004	X⇄Y	-41	
005	Y^x	31	
006	1	01	
007	-	-45	
008	STOB	35 12	2^{n-1}
009	DSP0	-63 00	
010	1	01	
011	STOI	35 46	1 in R_I
012	0	00	
013	STO0	35 00	FIRST CASE NR
014	STO8	35 08	
015	*LBLC	21 13	
016	RCL0	36 00	
017	STOC	35 13	
018	2	02	
019	÷	-24	
020	INT	16 34	
021	STO0	35 00	
022	RCLC	36 13	
023	2	02	
024	÷	-24	
025	FRC	16 44	REMAINDER
026	X=0?	16-43	
027	GTO0	22 00	
028	1	01	
029	STOi	35 45	STO i
030	*LBL1	21 01	
031	RCLI	36 46	
032	RCLA	36 11	
033	X=Y?	16-33	
034	GTOD	22 14	WHEN NR OF BINARY
035	ISZI	16 26 46	DIGITS = n, GTO D
036	GTOC	22 13	LOOP
037	*LBL0	21 00	
038	STOi	35 45	STO 0
039	GTO1	22 01	
040	*LBLD	21 14	
041	GSBB	23 12	
042	X≠0?	16-42	ALL PROPS ≠ 0?
043	GTO2	22 02	
044	0	00	
045	STO9	35 09	
046	*LBL3	21 03	
047	RCLi	36 45	NR. of PROPS.
048	1	01	
049	0	00	
050	RCLA	36 11	
051	RCLI	36 46	
052	-	-45	
053	Y^x	31	10^{n-i}
054	×	-35	
055	ST+9	35-55 09	ACCUM BIN. NR.
056	DSZI	16 25 46	EXIT LOOP ON 0
057	GTO3	22 03	LOOP
058	RCL9	36 09	VALID PROP.
059	R/S	51	STRING IN R_9.
060	*LBL2	21 02	
061	RCLB	36 12	
062	RCL8	36 08	
063	X=Y?	16-33	CASES = $2^n - 1$?
064	RTN	24	
065	1	01	
066	ST+8	35-55 08	INCREMENT CASE NR
067	STOI	35 46	
068	RCL8	36 08	
069	STO0	35 00	
070	GTOC	22 13	
071	*LBLB	21 12	DEFINE CONDITIONS.

REGISTERS

0	1	2	3	4	5	6	7	8	9
QUOTIENTS	BINARY REPRESENTATION OF CASE NR IN REVERSE							CASES	STRING

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I
n	$2^n - 1$				PROPOSTIONS

Appendix

97/67 KEY CODE CONVERSIONS*

97 CODE	67 CODE	mnemonic
00	00	0
01	01	1
02	02	2
03	03	3
04	04	4
05	05	5
06	06	6
07	07	7
08	08	8
09	09	9
24	35 22	RTN
31	35 63	Y↑X
32	31 52	LN
33	32 52	e↑x
34	32 72	->P
41	31 62	SIN
42	31 63	COS
43	31 64	TAN
44	31 72	->R
51	84	R/S
52	35 62	1/X
53	32 54	X↑2
54	31 54	SQRX
55	31 82	%
56	21	S+
-11	31 23	FIX
-12	32 23	SCI
-13	35 23	ENG
-14	31 84	PRTX
-21	41	ENT↑
-22	42	CHS
-23	43	EEX
-24	81	/
-31	35 53	R DOWN
-35	71	X
-41	35 52	X><Y
-45	51	-
-51	44	CLX
-55	61	+
-62	83	.
16 24	31 24	RND
16 31	35 64	ABS
16 32	31 53	LOG
16 33	32 53	10↑X
16 34	31 83	INT
16 35	32 74	->HMS
16 36	31 74	HMS->
16 41	32 62	SIN↑-1
16 42	32 63	COS↑-1
16 43	32 64	TAN↑-1
16 44	32 83	FRC
16 45	32 73	D->R
16 46	31 73	R->D
16 51	35 72	PSE
16 52	35 81	N!
16 53	31 21	X MEAN
16 54	32 21	S
16 55	32 82	%CH
16 56	35 21	S-
16-11	35 84	SPC
16-13	35 74	PREG
16-14	32 84	PRST
16-21	35 41	DEG
16-22	35 42	RAD
16-23	35 43	GRAD
16-24	35 73	PI
16-31	35 54	R↑
16-32	32 61	X<>Y?
16-33	32 51	X=Y?
16-34	32 81	X>Y?
16-35	32 71	X<=Y?
16-41	35 24	X><I
16-42	31 61	X<>0?
16-43	31 51	X=0?
16-44	31 81	X>0?
16-45	31 71	X<0?
16-51	31 42	P><S
16-53	31 43	CLRG
16-55	35 83	HMS+
16-61	31 41	WDTA
16-62	32 41	MRG
16-63	35 82	LSTX
21 00	31 25 00	*LBL0
21 01	31 25 01	*LBL1
21 02	31 25 02	*LBL2
21 03	31 25 03	*LBL3
21 04	31 25 04	*LBL4
21 05	31 25 05	*LBL5
21 06	31 25 06	*LBL6
21 07	31 25 07	*LBL7
21 08	31 25 08	*LBL8
21 09	31 25 09	*LBL9
21 11	31 25 11	*LBLA

97 CODE	67 CODE	mnemonic
21 12	21 25 12	*LBLB
21 13	31 25 13	*LBLC
21 14	31 25 14	*LBLD
21 15	31 25 15	*LBLE
22 00	22 00	GTO0
22 01	22 01	GTO1
22 02	22 02	GTO2
22 03	22 03	GTO3
22 04	22 04	GTO4
22 05	22 05	GTO5
22 06	22 06	GTO6
22 07	22 07	GTO7
22 08	22 08	GTO8
22 09	22 09	GTO9
22 11	22 11	GTOA
22 12	22 12	GTOB
22 13	22 13	GTOC
22 14	22 14	GTOD
22 15	22 15	GTOE
22 45	22 24	GTOi
23 00	31 22 00	GSB0
23 01	31 22 01	GSB1
23 02	31 22 02	GSB2
23 03	31 22 03	GSB3
23 04	31 22 04	GSB4
23 05	31 22 05	GSB5
23 06	31 22 06	GSB6
23 07	31 22 07	GSB7
23 08	31 22 08	GSB8
23 09	31 22 09	GSB9
23 11	31 22 11	GSBA
23 12	31 22 12	GSBB
23 13	31 22 13	GSBC
23 14	31 22 14	GSBD
23 15	31 22 15	GSBE
23 45	31 22 24	GSBi
35 00	33 00	STO0
35 01	33 01	STO1
35 02	33 02	STO2
35 03	33 03	STO3
35 04	33 04	STO4
35 05	33 05	STO5
35 06	33 06	STO6
35 07	33 07	STO7
35 08	33 08	STO8
35 09	33 09	STO9
35 11	33 11	STOA
35 12	33 12	STOB
35 13	33 13	STOC
35 14	33 14	STOD
35 15	33 15	STOE
35 45	33 24	STOi
35 46	35 33	STI
36 00	34 00	RCL0
36 01	34 01	RCL1
36 02	34 02	RCL2
36 03	34 03	RCL3
36 04	34 04	RCL4
36 05	34 05	RCL5
36 06	34 06	RCL6
36 07	34 07	RCL7
36 08	34 08	RCL8
36 09	34 09	RCL9
36 11	34 11	RCLA
36 12	34 12	RCLB
36 13	34 13	RCLC
36 14	34 14	RCLD
36 15	34 15	RCLE
36 45	34 24	RCLi
36 46	35 34	RCLI
36 56	34 21	RCLS+
-63 00	23 00	DSP0
-63 01	23 01	DSP1
-63 02	23 02	DSP2
-63 03	23 03	DSP3
-63 04	23 04	DSP4
-63 05	23 05	DSP5
-63 06	23 06	DSP6
-63 07	23 07	DSP7
-63 08	23 08	DSP8
-63 09	23 09	DSP9
-63 45	23 24	DSPi
16 21 00	35 51 00	SF0
16 21 01	35 51 01	SF1
16 21 02	35 51 02	SF2
16 21 03	35 51 03	SF3
16 22 00	35 61 00	CF0
16 22 01	35 61 01	CF1
16 22 02	35 61 02	CF2
16 22 03	35 61 03	CF3
16 23 00	35 71 00	F0?
16 23 01	35 71 01	F1?
16 23 02	35 71 02	F2?

97 CODE	67 CODE	mnemonic
16 23 03	35 71 03	F3?
16 25 45	32 33	DSZi
16 25 46	31 33	DSZI
16 26 45	32 34	ISZi
16 26 46	31 34	ISZI
21 16 11	32 25 11	*LBLa
21 16 12	32 25 12	*LBLb
21 16 13	32 25 13	*LBLc
21 16 14	32 25 14	*LBLd
21 16 15	32 25 15	*LBLe
22 16 11	22 31 11	GTOa
22 16 12	22 31 12	GTOb
22 16 13	22 31 13	GTOc
22 16 14	22 31 14	GTOd
22 16 15	22 31 15	GTOe
23 16 11	32 22 11	GSBa
23 16 12	32 22 12	GSBb
23 16 13	32 22 13	GSBc
23 16 14	32 22 14	GSBd
23 16 15	32 22 15	GSBe
35-24 00	33 81 00	ST/0
35-24 01	33 81 01	ST/1
35-24 02	33 81 02	ST/2
35-24 03	33 81 03	ST/3
35-24 04	33 81 04	ST/4
35-24 05	33 81 05	ST/5
35-24 06	33 81 06	ST/6
35-24 07	33 81 07	ST/7
35-24 08	33 81 08	ST/8
35-24 09	33 81 09	ST/9
35-24 45	33 81 24	ST/i
35-35 00	33 71 00	ST*0
35-35 01	33 71 01	ST*1
35-35 02	33 71 02	ST*2
35-35 03	33 71 03	ST*3
35-35 04	33 71 04	ST*4
35-35 05	33 71 05	ST*5
35-35 06	33 71 06	ST*6
35-35 07	33 71 07	ST*7
35-35 08	33 71 08	ST*8
35-35 09	33 71 09	ST*9
35-35 45	23 71 24	ST*i
35-45 00	33 51 00	ST-0
35-45 01	33 51 01	ST-1
35-45 02	33 51 02	ST-2
35-45 03	33 51 03	ST-3
35-45 04	33 51 04	ST-4
35-45 05	33 51 05	ST-5
35-45 06	33 51 06	ST-6
35-45 07	33 51 07	ST-7
35-45 08	33 51 08	ST-8
35-45 09	33 51 09	ST-9
35-45 45	33 51 24	ST-i
35-55 00	33 61 00	ST+0
35-55 01	33 61 01	ST+1
35-55 02	33 61 02	ST+2
35-55 03	33 61 03	ST+3
35-55 04	33 61 04	ST+4
35-55 05	33 61 05	ST+5
35-55 06	33 61 06	ST+6
35-55 07	33 61 07	ST+7
35-55 08	33 61 08	ST+8
35-55 09	33 61 09	ST+9
35-55 45	33 61 24	ST+i

*Supplied by courtesy of R. W. Edelen.