

<sup>5</sup> S. BERGMAN, "Two-dimensional subsonic flows of a compressible fluid and their singularities," Amer. Math. Soc., *Trans.*, v. 62, 1947, p. 452-498.

<sup>4</sup> S. A. CHAPLYGIN, "On gas jets," *Scientific Memoirs*, Moscow Univ., Math. Phys. Sec., 21 (1902), p. 1-21. (English translation published by Nat. Adv. Com. for Aeronautics, Technical Note No. 1064, 1944, and also by Brown University, 1944.)

<sup>6</sup> See page 462 of the reference of note 3.

<sup>6</sup> See the reference in note 2 and equations (2.8) and (2.9) of the reference in note 3.

<sup>7</sup> See page 465 of the reference of note 3, where symbols  $F$  and  $P$  were used for the  $F_1$  and  $F_2$  of the present paper.

<sup>8</sup> See page 472 of the reference of note 3.

<sup>9</sup> We note that when we are carrying out the integrations in (11) and (12),  $\lambda_n$  are continued to complex values  $\lambda_n + i\Delta_n$ , and  $Z_n = \lambda_n + i\Delta_n$  and  $\bar{Z}_n = \lambda_n - i\Delta_n$  become two independent variables. Also see page 473 of the reference in note 3.

## Analysis of Problem Codes on the Maniac

The Los Alamos computer, the MANIAC, has solved problems of wide variety during the last few years. The mathematical structure of such problems has ranged from differential equations (particularly partial differential equations), integral equations, stochastic processes, purely algebraic problems to some in the domain of mathematical logics.

There are several reasons why a frequency analysis of the computer as used in several typical problems might be useful. From such a study one may learn of significant variations in such distributions from one type of problem to another. Further, one may reach conclusions about the selection of the computer vocabulary. Most importantly, however, one may use the quantitative results as guiding principles in the design of a new computer. The economy of computer design is connected with the question, "Is the desirability of a particular order<sup>1</sup> commensurate with the associated electronic hardware?" Finally, a frequency analysis enables one to form accurate estimates for the "running time" of a problem; this information aids considerably in efficient scheduling of computer time. A knowledge of operation times for subroutines enables one to make rather good time estimates of lengthy problems during the formulation stage.

These frequency distributions can, of course, be gathered by hand. A more obvious way is to have the computer itself perform the analyses. A routine has been developed for this purpose and is called the "Code Analyzer."

The Code Analyzer gives the following information about a computer problem:

1. the frequency of occurrence of each order as it appears in the code—more briefly, a "static" count
2. the distribution in per cent of these static counts
3. the frequency of performance of each order during the running of a representative cycle of the problem—more briefly, a "dynamic" count
4. the distribution in per cent of these dynamic counts
5. the total time consumed performing each order of the vocabulary; i.e., (3) multiplied by the time needed to perform one such order
6. the per cent of the total time used by each order
7. the totals for (1), (3), (5).

The count in (1) is obtained by simply scanning linearly through the code of the problem and recording the occurrence of each order. The distribution (2) is

the result of normalizing the distribution (1). To obtain (3), frequency of dynamic count, the problem must be run by means of an interpretive routine<sup>2</sup> (in our case it has actually the same vocabulary as the computer) which tallies one for the order as each instruction proper is performed. Distribution (4) is simply a normalization of (3). The operation times for the various individual orders are stored as constants within the Code Analyzer, and therefore only a simple multiplication is needed to obtain (5) in all cases except for the shift orders which are, of course, of variable duration. Each shift order is examined to determine the number of places shifted and the time computed accurately according to a simple linear expression.

Operation times were obtained by taking the time difference of a large number of passes through a control loop with and without each order. Dividing this time difference by the total number of passes, one obtains a realistic measure of the individual operation times that includes the access time for each instruction.

When the above statistical information has been gathered, the percentages and totals are calculated and all six columns (1 to 6, above) are printed, preceded by a column which lists the MANIAC's thirty-six orders. Totals (7) are printed below the appropriate columns. A brief explanation of the vocabulary symbols is given in Table 1.

Conceptually, the idea of the Code Analyzer is very simple. In practice, this would also be true provided a computer had only one medium of storage which was sufficiently capacious to contain *simultaneously* the Code Analyzer and the problem at hand. However, this condition does not obtain for the MANIAC which has one thousand and twenty-four locations of electrostatic storage (primary) and ten thousand locations of magnetic drum storage (secondary).

The communication between the two media gives rise to considerable complications during the interpretative process. As a result, the code for the Code Analyzer itself is by no means short; indeed, it has one thousand instructions, apart from numerical storage.

The procedure is as follows: First, the problem to be analyzed is loaded into the storage normally, as if it were going to be run. Then a key instruction is inserted at the end of a typical cycle to indicate the terminal point of interpretation. With very short problems, or with subroutines, this stop is the end of the problem or the exit from the subroutine. The code of the problem residing in the electrostatic memory is then transferred to the drum at a place that the Code Analyzer will use as a simulated one thousand twenty-four word memory during interpretation. In the interest of speed, the numerical storage, both constant and dynamic, remains in the true electrostatic memory at all times. This is possible if there are several Code Analyzers, each occupying a different section of the electrostatic memory, so one may be chosen which does not conflict with the numerical storage of the problem being analyzed. The next step is to load the selected Code Analyzer into the memory. To allow for problems with prodigious storage, all of the code for the Code Analyzer does not reside in the electrostatic memory at one time. The Code Analyzer divides naturally into two parts since the counting is independent of the totaling, computing of percentages, and printing of the output. Thus, only the code for the counting is loaded immediately into electrostatic storage. The second part of the code is temporarily stored on the magnetic

drum. Then, with the help of an auxiliary read tape which contains the initial and final addresses of the problem code, the address where interpretation is to begin, and the initial and final addresses of the problem storage, the Code Analyzer proceeds with its work and prints the results as described above. In addition both the dynamic and static counts for each analysis are punched on paper tape.

TABLE 1

*A Brief Description of the MANIAC Vocabulary Symbols*

The letters are tetrads of binary digits. A, B, . . . , F correspond to the hexadecimal characters 10, 11, . . . , 15. Associated with the pair of vocabulary symbols for each order are three tetrads specifying the memory location of the operand. There are two instructions per word; the occurrence of some orders in duplicate is necessary for reference to the left or right portions of the word.

Order No.	Vocabulary Symbol	Interpretation
1	AA	Add number (N) in memory (M) to cleared accumulator (A)
2	AB	Subtract N in M from cleared A
3	AC	Recall N on magnetic tape to quotient register (Q)
4	AD	Record N in Q on magnetic tape
5	AE	Add absolute value of N in M to cleared A
6	AF	Subtract absolute value of N in M from cleared A
7	BA	Add N in M to uncleared A
8	BB	Subtract N in M from uncleared A
9	BC	Recall 50 words (one track) from drum to M
10	BD	Record 50 words in M to drum
11	BE	Add absolute value of N to uncleared A
12	BF	Subtract absolute value of N from uncleared A
13	CA	Unconditional transfer of control
14	CB	Unconditional transfer of control
15	CC	Conditional transfer of control
16	CD	Conditional transfer of control
17	CE	High speed print (Synchroprinter)
18	CF	Punch paper tape
19	DA	Multiplication (Round Off)
20	DB	Multiplication (No round off)
21	DC	Replacement of N in A to M
22	DD	Division
23	DE	Left shift
24	DF	Add associated address to number in A
25	EA	Slow speed print (Teletype)
26	EB	Place N from M in Q
27	EC	Place N from Q in M
28	ED	Drop sign of number in A
29	EE	Right shift
30	EF	Place associated address in A
31	FA	Address replacement in M
32	FB	Address replacement in M
33	FC	Half word replacement in M
34	FD	Half word replacement in M
35	FE (not used)	
36	FF	N on paper tape input to M
37	800	Add N from Q to A

These tapes provide the data in a convenient and flexible form to make cumulative analyses of various forms.

Results to date include analyses of some typical Laboratory problems. In Table 2 the analysis of a problem in hydrodynamics is shown. Analyses are also given in Tables 3 and 4 for a Monte Carlo problem and for one in mathematical logics, respectively. A major portion of the MANIAC's subroutines has been analyzed. The data for thirty-seven subroutines are shown summarized in Table 5. The running time for the analysis of the hydrodynamics problem was twenty-five

TABLE 2  
*Analysis of the Code for a Problem in Hydrodynamics*

Vocabulary Symbol	Static Count	Percentage of Static Count	Dynamic Count	Percentage of Dynamic Count	Time	Percentage of Time
AA	156	13.5	3499	12.3	314.9	5.4
AB	8	0.6	309	1.0	27.8	0.4
AC	0	0.0	0	0.0	0.0	0.0
AD	0	0.0	0	0.0	0.0	0.0
AE	0	0.0	0	0.0	0.0	0.0
AF	1	0.0	40	0.1	3.6	0.0
BA	93	8.0	2745	9.6	247.0	4.2
BB	68	5.8	2149	7.5	193.4	3.3
BC	0	0.0	0	0.0	0.0	0.0
BD	0	0.0	0	0.0	0.0	0.0
BE	0	0.0	0	0.0	0.0	0.0
BF	4	0.3	4	0.0	0.3	0.0
CA	34	2.9	332	1.1	16.6	0.2
CB	49	4.2	936	3.3	46.8	0.8
CC	13	1.1	447	1.5	20.1	0.3
CD	24	2.0	770	2.7	34.6	0.6
CE	0	0.0	0	0.0	0.0	0.0
CF	0	0.0	0	0.0	0.0	0.0
DA	93	8.0	2498	8.8	2592.9	45.1
DB	3	0.2	80	0.2	83.0	1.4
DC	157	13.6	4163	14.6	249.7	4.3
DD	49	4.2	1154	4.0	1197.8	20.8
DE	28	2.4	763	2.6	106.8	1.8
DF	6	0.5	12	0.0	0.8	0.0
EA	0	0.0	0	0.0	0.0	0.0
EB	118	10.2	3225	11.3	209.6	3.6
EC	104	9.0	2430	8.5	160.3	2.7
ED	13	1.1	37	0.1	1.4	0.0
EE	11	0.9	407	1.4	53.7	0.9
EF	19	1.6	86	0.3	6.4	0.1
FA	38	3.2	1101	3.8	88.0	1.5
FB	42	3.6	1063	3.7	85.0	1.4
FC	0	0.0	0	0.0	0.0	0.0
FD	0	0.0	0	0.0	0.0	0.0
FF	17	1.4	0	0.0	0.0	0.0
800	3	0.2	83	0.2	6.1	0.1
Totals	1151		28333		5747.4	

minutes, the Monte Carlo problem took seven minutes, and the logical problem took twenty minutes. Each subroutine analysis was a matter of seconds.

Examination of these preliminary results provides quantitative estimates to such matters as the effect on the running time for a problem if the operation time of some order were decreased by some factor. In other words, attention may be focussed where the "shoe pinches most." For example, a decrease of fifty per cent in the multiplication time alone would cut the running time by 30 per cent in some of the typical problems. We also see that some orders could have been

TABLE 3  
*Analysis of the Code for a Monte Carlo Problem*

Vocabulary Symbol	Static Count	Percentage of Static Count	Dynamic Count	Percentage of Dynamic Count	Time	Percentage of Time
AA	37	18.2	2111	14.4	189.9	6.6
AB	2	0.9	112	0.7	10.0	0.3
AC	0	0.0	0	0.0	0.0	0.0
AD	0	0.0	0	0.0	0.0	0.0
AE	2	0.9	499	3.4	44.9	1.5
AF	0	0.0	0	0.0	0.0	0.0
BA	20	9.8	1326	9.1	119.3	4.1
BB	11	5.4	730	5.0	65.7	2.2
BC	0	0.0	0	0.0	0.0	0.0
BD	0	0.0	0	0.0	0.0	0.0
BE	0	0.0	0	0.0	0.0	0.0
BF	0	0.0	0	0.0	0.0	0.0
CA	4	1.9	89	0.6	4.4	0.1
CB	12	5.9	395	2.7	19.7	0.6
CC	3	1.4	231	1.5	10.3	0.3
CD	10	4.9	786	5.3	35.3	1.2
CE	0	0.0	0	0.0	0.0	0.0
CF	0	0.0	0	0.0	0.0	0.0
DA	3	1.4	909	6.2	943.5	32.9
DB	4	1.9	356	2.4	369.5	12.9
DC	38	18.7	2972	20.4	178.3	6.2
DD	2	0.9	499	3.4	517.9	18.0
DE	9	4.4	1085	7.4	146.4	5.1
DF	1	0.4	89	0.6	6.2	0.2
EA	0	0.4	0	0.0	0.0	0.0
EB	9	4.4	864	5.9	56.1	1.9
EC	7	3.4	454	3.1	29.9	1.0
ED	8	3.9	210	1.4	8.4	0.2
EE	4	1.9	356	2.4	67.6	2.3
EF	6	2.9	109	0.7	8.1	0.2
FA	4	1.9	26	0.1	2.0	0.0
FB	0	0.4	0	0.0	0.0	0.0
FC	1	0.4	89	0.6	7.1	0.2
FD	2	0.9	89	0.6	7.1	0.2
FF	0	0.0	0	0.0	0.0	0.0
800	2	0.9	178	1.2	13.1	0.4
Totals	201		14564		2861.8	

sacrificed without too much pain or regret. Examination of subroutine times has led to a re-evaluation of some of the methods used and in some cases has resulted in vast improvements.

It is planned to continue gathering statistics of problems computed on the MANIAC to establish a more quantitative basis for some of our ideas and conjectures.

TABLE 4

*Analysis of the Code for a Problem in Mathematical Logics*

Vocabulary Symbol	Static Count	Percentage of Static Count	Dynamic Count	Percentage of Dynamic Count	Time	Percentage of Time
AA	147	19.4	2025	16.5	182.2	2.1
AB	6	0.7	20	0.1	1.8	0.0
AC	0	0.0	0	0.0	0.0	0.0
AD	0	0.0	0	0.0	0.0	0.0
AE	2	0.2	4	0.0	0.3	0.0
AF	1	0.5	1	0.0	0.0	0.0
BA	70	9.2	1018	8.3	91.6	1.0
BB	30	3.9	412	3.3	37.0	0.4
BC	0	0.0	0	0.0	0.0	0.0
BD	0	0.0	0	0.0	0.0	0.0
BE	0	0.0	0	0.0	0.0	0.0
BF	0	0.0	0	0.0	0.0	0.0
CA	26	3.4	30	0.2	1.5	0.0
CB	29	3.8	84	0.6	4.2	0.0
CC	28	3.7	625	5.1	28.1	0.3
CD	35	4.6	1075	8.7	48.3	0.5
CE	1	0.1	180	1.4	453.6	5.2
CF	0	0.0	0	0.0	0.0	0.0
DA	0	0.0	0	0.0	0.0	0.0
DB	1	0.1	9	0.0	9.3	0.1
DC	90	11.9	1745	14.2	104.7	1.2
DD	0	0.0	0	0.0	0.0	0.0
DE	45	5.9	978	7.9	106.6	1.2
DF	27	3.5	71	0.5	4.9	0.0
EA	4	0.5	4	0.0	6647.6	77.1
EB	27	3.5	1031	8.4	67.0	0.7
EC	20	2.6	626	5.1	41.3	0.4
ED	12	5.1	12	0.0	0.4	0.0
EE	31	4.1	93	0.7	20.6	0.2
EF	13	1.7	450	3.6	33.7	0.3
FA	19	2.5	423	3.4	33.8	0.3
FB	19	2.5	825	6.7	66.0	0.7
FC	8	1.0	15	0.1	1.2	0.0
FD	10	1.3	30	0.2	2.4	0.0
FE	0	0.0	0	0.0	0.0	0.0
FF	5	0.6	15	0.1	600.0	6.9
800	19	2.5	435	3.5	32.1	0.3
Totals	725		12236		8621.0	

TABLE 5  
*Analyses of the Codes for Some of the Subroutines, Summarized*

Vocabulary Symbol	Static Count	Percentage of Static Count	Dynamic Count	Percentage of Dynamic Count	Time	Percentage of Time
AA	450	16.7	5659	17.6	509.3	3.5
AB	21	0.7	123	0.3	11.0	0.0
AC	0	0.0	0	0.0	0.0	0.0
AD	0	0.0	0	0.0	0.0	0.0
AE	7	0.2	23	0.0	2.0	0.0
AF	10	0.3	10	0.0	0.9	0.0
BA	187	6.9	2643	8.2	237.8	1.6
BB	83	3.0	1833	5.7	164.9	1.1
BC	0	0.0	0	0.0	0.0	0.0
BD	0	0.0	0	0.0	0.0	0.0
BE	0	0.0	0	0.0	0.0	0.0
BF	3	0.1	38	0.1	3.4	0.0
CA	66	2.4	324	1.0	16.2	0.1
CB	68	2.5	476	1.4	23.8	0.1
CC	92	3.4	1572	4.9	70.7	0.4
CD	101	3.7	2601	8.1	117.0	0.8
CE	42	1.5	378	1.1	952.5	6.6
CF	0	0.0	0	0.0	0.0	0.0
DA	15	0.5	81	0.2	84.0	0.5
DB	15	0.5	227	0.7	235.6	1.6
DC	349	13.0	5079	15.8	304.7	2.1
DD	11	0.4	81	0.2	84.0	0.5
DE	136	5.0	2184	6.8	251.1	1.7
DF	70	2.6	429	1.3	30.0	0.2
EA	6	0.2	6	0.0	9971.4	69.8
EB	95	3.5	1634	5.1	106.2	0.7
EC	61	2.2	1045	3.2	68.9	0.4
ED	227	16.2	227	0.7	9.0	0.0
EE	54	2.0	255	0.7	33.1	0.2
EF	65	2.4	717	2.2	53.7	0.3
FA	53	1.9	906	2.8	72.4	0.5
FB	67	2.4	2214	6.9	177.1	1.2
FC	29	1.0	36	0.1	2.8	0.0
FD	27	1.0	46	0.1	3.6	0.0
FF	5	0.1	15	0.0	600.0	4.2
800	58	2.1	1147	3.5	84.8	0.5
Totals	2572		32009		14283.3	

EUGENE H. HERBST  
 N. METROPOLIS  
 MARK B. WELLS

Los Alamos Scientific Laboratory  
 Los Alamos, New Mexico

Work performed under the auspices of the Atomic Energy Commission.

<sup>1</sup> An "order" here refers to a single arithmetical or logical operation; e.g., add, multiply, transfer control. The ensemble of orders constitutes the computer vocabulary. An "instruction" refers to an order and its associated address.

<sup>2</sup> An interpretive routine is one which simulates a computer control and arithmetic unit. In effect, it translates and performs a coded sequence of instructions.