Contract No. DA-36-034-ORD-1023
Project No. TB3-0007

FINAL REPORT

on

CONTRACT NO. DA-36-034-ORD-1023

by

The Staff
Electronic Computer Project

THE INSTITUTE FOR ADVANCED STUDY
ELECTRONIC COMPUTER PROJECT
PRINCETON, NEW JERSEY

Contract No. DA-36-034-ORD-1023
Project No. TB3-0007

FINAL REPORT

on

CONTRACT NO. DA-36-034-ORD-1023

by

The Staff
Electronic Computer Project

THE INSTITUTE FOR ADVANCED STUDY
ELECTRONIC COMPUTER PROJECT
April 1954

PREFACE

The following report has been prepared in accordance with the terms of Contract No. DA-36-034-ORD-1023 and constitutes the Final Report called for under the terms of that contract. It is divided into two parts (I and II) covering the salient engineering work and the mathematical investigations for the period July 1, 1952 through June 30, 1953. The actual carrying out of the calculations indicated in Part II was done under the terms of the contract. The mathematical preparations, i.e. the numerical analysis, programming and coding were carried out under the terms of Contract No. N-7-ONR-388, T. O. I and Contract No. N-6-ORI-139, T. O. I between the Institute for Advanced Study and the Office of Naval Research. Since the objectives of the three contracts are substantially overlapping it was felt desirable to include all this material in one report. In this fashion it is hoped to give the maximum possible information to all interested agencies.

John von Neumann
Project Director

Institute for Advanced Study

ACKNOWLEDGMENTS

The present report represents the combined efforts of the staff of the Electronic Computer Project of the Institute for Advanced Study and in particular of the following individuals: Hewitt D. Crane, Bruce Gilchrist, Herman H. Goldstine, James H. Pomerene, Daniel L. Slotnick, and Bryant Tuckerman. In addition, the editors drew freely on material from unpublished works by: N. A. Barricelli, W. R. Beam, G. Birkhoff, B. C. Carlson, S. Chandrasekhar, G. Estrin, F. J. Murray, H. H. Goldstine, J. B. Rosen, M. Schwarzschild, J. von Neumann, and E. Zarantonello.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

FIGURES LIST

PART I

PART II

PART I - ENGINEERING

I. INTRODUCTION

This report describes the operation of and engineering improvements on the electronic computer at the Institute for Advanced Study during the period from 1 July 1952 to 30 June 1953. The engineering discussion is restricted to those features of the present machine which were added during this period. The underlying design philosophy and a full technical description of the machine prior to this period is given in the final report on Contract Nos. W-36-034-ORD-7481 and DA-36-034-ORD-19 (Project No. TB3-0007 F).

II. SUMMARY

At the beginning of the period covered by this report the computer was in operation for two shifts each weekday using teletype tape as an input-output medium. In October 1952 a regular series of routine tests was instituted to provide a daily check on the operating state of the machine. In November 1952 a punched card input-output system was placed in operation using an IBM type 514 reproducing punch to accomplish a ten-fold increase in loading speed and a twenty-fold increase in punch-out speed. A 2048 word Magnetic Drum memory was developed and operated on a limited test basis in May and June 1953.

III. ENGINEERING WORK

A. IBM Input-Output and Magnetic Drum.

1. IBM Reproducer (Model 514). In order to be able to transfer information into and out of the machine at a rate faster than that obtainable with the teletype equipment it was decided to incorporate a standard IBM 514 Reproducing punch as a part of the input-output equipment. Using 40 columns of a standard IBM card and allowing each row (i.e. each emitter position) for a different 40-digit binary word, we can pack twelve 40-digit words to a card. At a speed of 100 cards a minute we are then able to read in or punch out a complete memory load (1024 words) in less than a minute.

Magnetic Drum. In order to increase the size of the available memory, it was decided to add a 2048 word secondary magnetic drum memory to the machine. This auxiliary memory communicates only with the main Williams memory so that under orders from the main control, information is read into or from the auxiliary memory, from or to the Williams memory. A magnetic drum five inches in diameter had been built earlier and is described in the "First Progress Report on a Multi-Channel Magnetic Drum Inner Memory for Use in Electronic Digital Computing Instruments". A similar drum eight inches in diameter was finally built and is presently being used.

This section then shall cover the complete circuitry for the IBM and Drum equipment.

Since use of these two pieces of equipment are mutually exclusive events (i.e. at any moment only one or the other operates) much of the new circuitry is shared by both units. For instance, the digital

information for the IBM and Drum is obtained from exactly the same place in the machine so that only one information cable connects both of these units to the machine. Similarly, since the control functions required for the operation of both the IBM and Drum are identical (this will become clearer as we go on) one control circuit operates both (except for such things as processing the synchronizing signals from the timing tracks of the drum and the emitter pulses from the IBM).

We shall cover first the circuitry for the digital information. To do this we shall have to describe the processes of reading and writing onto the drum and the method of punching and reading IBM cards.

2. Drum recording heads. Early tests showed that a packing of 50 digits per inch is very feasible (and with proper adjustments 75 digits per inch can also be reliably handled). With an eight inch drum and a spacing of approximately 45 digits to the inch, we pack 1024 spots around the periphery. To attain a capacity of 2048 40-digit words it was decided to use 80 tracks (i.e. two groups of 40 each) with 1024 spots around each track and to switch as required between the two groups.

In order to arrive as soon as possible at a workable solution commercially available heads were used. Several types of heads were tested as follows: The magnetic material was first polarized by letting the drum rotate while biasing the magnetic coating with D.C. through the head. Then the drum was stopped and signals with the opposite polarity and varying current strength were recorded in various positions. The head was then connected to an amplifier and the drum rotated. The graph below shows the amplitudes of the reading pulses as a function of the recorded current, obtained with Ratheon head RX3009.

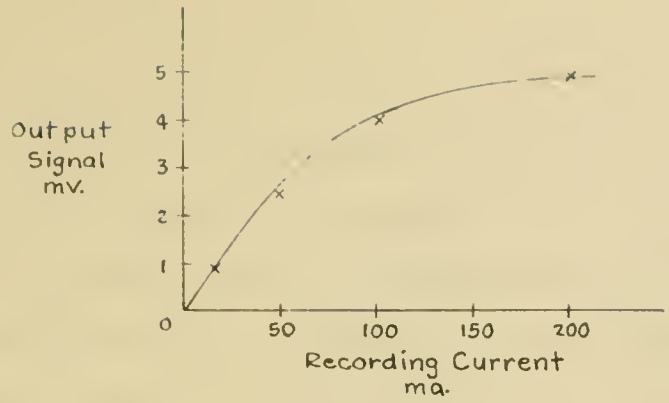


Figure 1

One half of the head was then rewound with five times the number of original turns, and essentially stronger signals were obtained. However, the resulting higher inductance would have necessitated voltage during writing that would have been too high for the simple writing circuits proposed. In any case tests done at the same time with heads described below, showed them to be better adapted for our use.

Tests on the Brush heads of the BK-1500 series gave the following results.

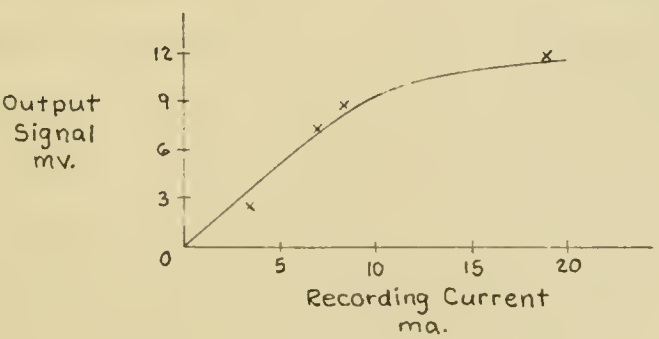


Figure 2

It was decided to use this type of head, and six multiple units of 14 heads each (or 84 total tracks) were obtained (these are standard units except for a two mil air-gap). In addition, a multiple unit with three heads was obtained for Timing-Track use. For operation, the tracks are originally erased to a magnetically neutral position (by a continually decreasing A.C. signal) and 0 and 1 information are stored on the drum by pulses of opposite polarity.

3. Magnetic materials. Two types of magnetic materials were obtained from the Minnesota Mining and Manufacturing Company, a black iron oxide and a red iron oxide. It was found that the red iron oxide saturated at lower fields but gave weaker output signals than the black material. The signal-noise ratio was the same for both materials. By noise here is meant the output voltage obtained after erasing a track with currents that just produce saturation. The red oxide was selected mainly because of the smaller writing currents required to produce saturation, the smaller reading signals not being so great a problem.

For coating, the drum was first slightly etched with dilute nitric acid. Then a primer was applied by allowing the drum to rotate slowly in a lathe while the spraying gun was moved automatically across the drum. This priming coat then was "baked-in" by heating with an infra-red lamp for several hours. Finally the oxide was sprayed on to a thickness of about one mil but was not "baked-in", as per supplier's instructions. Experiments showed that the thickness of the magnetic coating was not too critical, except that for thicknesses above two mils the tracks became difficult to erase.

4. Pulsers (or writing units). Information from the machine

is fed into the pulsers which in turn issue writing pulses at the appropriate times. These times are defined by the synchronizing pulses derived from the timing tracks. The heads obtained from Brush had the center tap available which made the pulser circuit extremely simple.

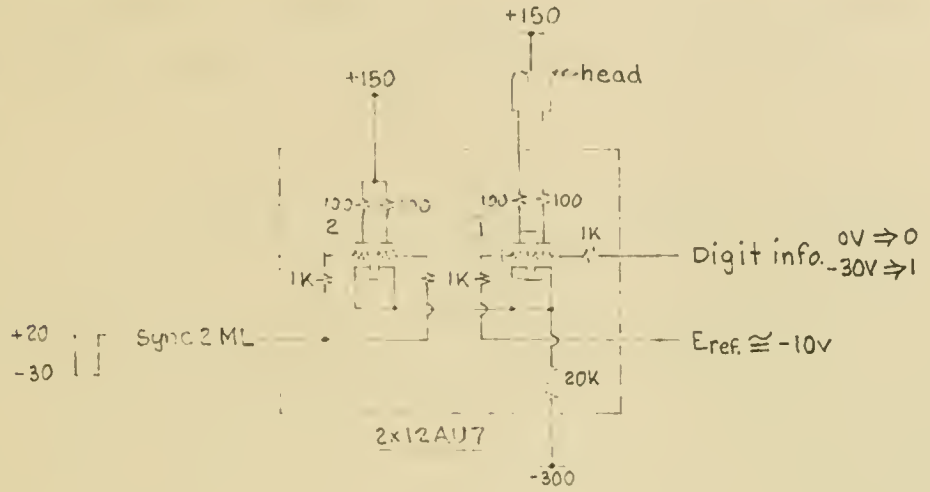


Figure 3

The pulser is essentially a transfer gate with three inputs. Two 12AU7's comprise the tube complement of each of the 40 pulsers. During standby the grids of tube 2 are at about plus 20 volts, so that tube 1 is completely cut-off and no current flows through the heads. During a Sync 2ML pulse the grids of tube 2 drop to about 30 volts negative, so that the current is switched to one or the other half of the first tube and therefore through one half of the head. The left grid of tube 1 has a fixed voltage of -10 and the right grid has a level of 0 volts for a zero digit and about -30 volts for a one digit. For a zero then at Sync 2ML time the right half will conduct and for a one the left half thus writing the appropriate information on the drum.

The writing current of 15 ma is enough to drive the magnetic material from saturation in one direction to saturation in the other. The use of the full double triode to act as a standby tube is a matter of history. Initially it was planned to use 30 ma writing current (and 5687 tubes instead of 12AU7's) and under our design considerations for tubes a single section could not handle this amount of current continuously.

5. Reading from the drum - Amplifier. Experiments showed that an amplification of about 1500 is necessary with the head and magnetic material chosen. The simplest solution seemed to be a condenser-coupled two-stage amplifier using a double triode. Several amplifiers with different tubes were built. A 12AX7 amplifier was then adopted -- circuit is shown below. The amplifier does not have very good high frequency response, but it was felt that this would not be a serious drawback, since even if the packing of the drum could be increased later by careful adjustments of the heads, the heads themselves and not the amplifier would be the limiting factor. In this case the drum would have to be run at a correspondingly slower speed.

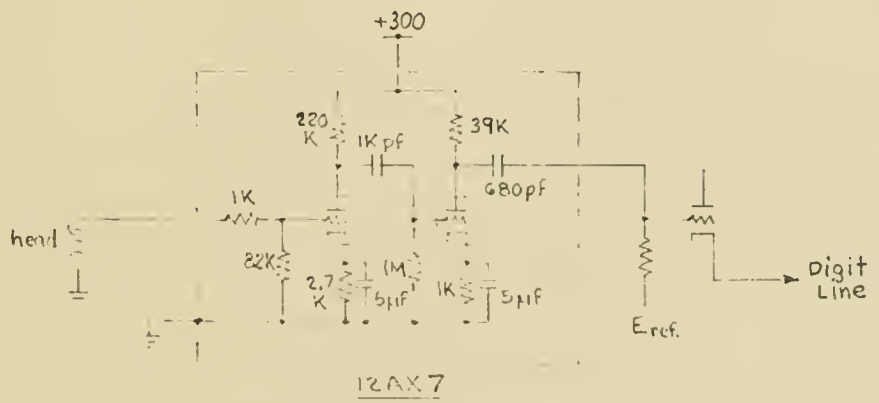


Figure 4

6. Reading and Punching IBM cards. There are two feed units in the IBM 514 reproducing punch -- the read unit and the punch unit. Cards may be fed into either or both of the units according to the operation being performed. The relation of the two units to each other, and the sequence in which the cards pass the operating stations in the two units is indicated below.

Figure 5

Cards fed in the punch unit first pass the six punch X brushes. The following station is the punching mechanism, consisting of 80 punches, or one per card column. The card passes the punches with its "12" edge first, so that all the appropriate columns of the card are punched in their "12" position first, followed by punching in the "11" position, etc., up until the last or "9" position. Thus the card is completely punched in twelve steps of the card cycle. A card passes the X brushes and punch dies during one card cycle as it passes through the unit.

The next and last station in the punch unit is the Punch brushes.

A card passes the punch brushes on the next card cycle as it passes through the unit. Thus when the "5" position of the first card is being read by the punch brushes, for example, the "5" position of the card immediately following is under the punches.

Cards fed in the reading unit first pass under the read X brushes. At the following station are the 80 Reproduce brushes, or one for each card column. A card passes the X brushes and reproducing brushes on one card cycle as it passes through the read unit. The next and last station is the 80 Compare brushes, or again one for each column of the card. The card passes the comparing brushes on the next card cycle, as it passes through the reading unit. Thus, when the "5" position of a card is passing over the Compare brushes, the "5" position of the card immediately following is over the Reproduce brushes.

When both units are used together for any operation, the cards feed simultaneously through the two units in such a way that at the same time that the "5" position of a card is being read at the Reproduce brushes the "5" position of another card is being punched at the punch unit. As the card in the reading unit passes on to the Compare brushes, the card in the punch unit passes on to the Punch brushes -- and at the time that the "5" position is being read at the Compare brushes, the "5" position of the other card is being read at the Punch brushes.

The Comparing unit of the Reproducer makes it possible to compare the punching in two cards (one in the read unit and one in the punch unit) for purposes of verification. When on "verify" if the punching in the two cards is different, the machine will stop and the comparing indicator unit will point out the column(s) in which discrepancies exist.

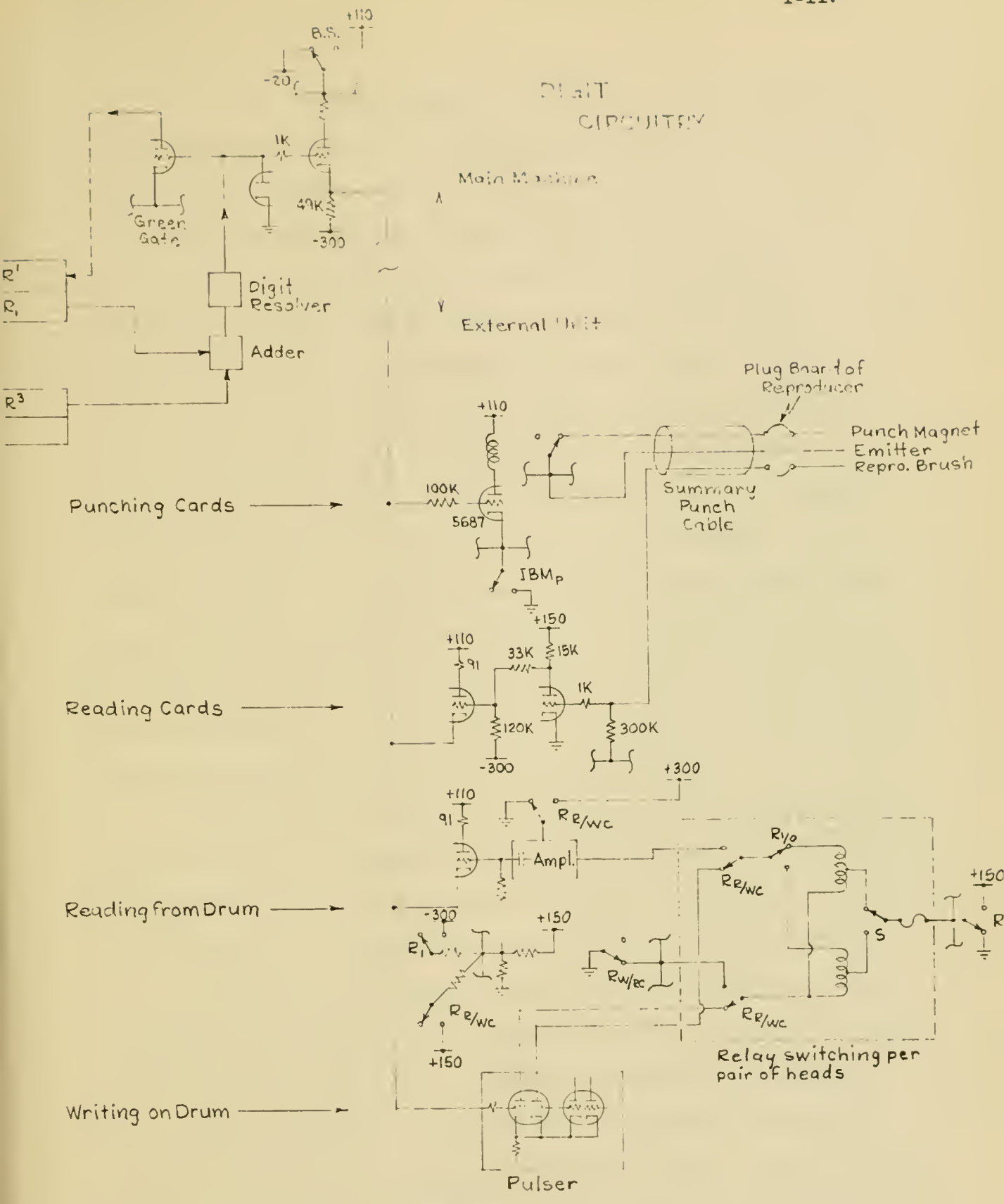

+110
B.S.
-20
1K
49K
-300
Green Gate
R'
R1
Digit Resolver
Adder
R3
Main Machine
External Unit
Plug Board of Reproducer
Punch Magnet
Emitter
Repro. Brush
Summary Punch Cable
Punching Cards
100K
5687
IBMp
+150
+110
91
33K
15K
1K
120K
300K
-300
Reading Cards
+300
RR/WC
Ampl.
R1/0
RW/RC
S
Reading from Drum
Relay switching per pair of heads
Writing on Drum
Pulser

Figure 6

7. Digital Information Circuitry. All the digits of a complete word are read from or written onto the drum at the same instant from terminal equipment in the main machine which is 40 digits "wide". As noted above the drum has two groups of tracks (40 tracks per group -- or a total of 80). Therefore it is necessary to switch between one or the other group. Further, the same heads are used for both reading and writing so we must also switch the heads to the appropriate circuitry for either case. Since the magnetic drum forms an auxiliary memory to the existing high-speed Williams memory, speed is not of primary concern. Approximately 200 milliseconds is the total time required to transfer a complete memory load to or from the drum (the drum rotates at approximately 600 RPM). Since the necessary switching indicated above appears rather bulky and complex if done with tubes it seemed rather reasonable to tolerate the relatively slower switching facilities of relays (adding 25 milliseconds or so to the total time). Therefore IBM "4 point" wire-contact relays were obtained and used for this purpose.

To read or write 40 digits simultaneously requires 40 Amplifiers and 40 Pulsers. The required switching between groups of heads and between amplifiers and pulsers for reading and writing, respectively, is indicated in the drawing of the Digital Information Circuitry above (Figure 6).

The output of the Digit Resolver in the main machine is a high impedance source that feeds the Green Gates of R^1. This is an excellent place to either remove from or introduce information to the machine.

With R_1 cleared the output of the Digit Resolver displays the information held in R^3. After an RII-Load-Clear order, then, the output of the Digit Resolver displays the binary number just taken from the given

Williams Memory position. A cathode follower hooked on to the digit resolver output (B.S. relay operated) then pipes this word over the digit information cable to be recorded either on the Drum or IBM.

With the B.S. relay deenergized the cathode follower on the digit resolver output looks electrically like a diode, and in this capacity can only pull the digit resolver negative. To transmit a word to the machine from the Drum or IBM, then, it is only necessary to arrange that the output of the digit resolver ($R_1 + R^3$) be zero (positive) in every stage, and let the digit line fall (negative) in only those stages in which a one is to be presented. Under these conditions the correct number is impressed upon the green gates of R^1 so that a simple Green Gate Accept followed by a Zig-Zag will bring the number into R_1 from where it is stored in the correct address in the Williams Memory. To insure that the digit resolver output is zero at the proper moments, the Complement Gates (from R^3) are nulled throughout the entire order and R_1 is precleared before each new word is brought in. Of course, the words brought in over the digit lines are obtained from the reading amplifiers of the Drum or the Reproduce brushes of the reproducer.

To elaborate then, we see from the drawing that a single wire is used (per digit) to serve the multiple purpose of reading into or out of the drum, or into or out of the IBM. At the "bumper strip" (output of the digit resolver) the cathode of the triode acts as a terminal point for the information wire that connects through a cable to the external equipment. During normal operate (or compute) the plates of all these 40 bumper strip triodes are returned via a relay point to -20 volts, so that all these tubes operate as diodes (the grid acting as the plate).

However, the diode is cut-off since the upper level of its grid is bumped at ground, and the cathode of the bumper strip triode is maintained at about +10 volts. Thus during "Compute" the external circuitry is completely dissociated (electrically) from the main machine.

a. Punching out to the IBM. (Note that for writing out on the drum or IBM the B.S. relay is energized making the bumper strip triodes look like full-fledged cathode followers piping information from the output of the digit resolver out onto the digit lines.) For this operation relays BS, IBM, IBM_P, R_1 and others are energized. With the cathodes of the 5687 "punch relay" tubes grounded this tube responds to the information on the digit line, operating the fast acting relay in its plate which in turn, via the Summary Punch Cable of the Reproducer, controls the punch magnets.

b. Writing on the drum. For this operation relays BS, R_1, $R_{W/RC}$ and others are energized. At Sync 2ML time the information residing on the digit lines is written onto the drum as previously discussed. Note - Sync 2ML signals come only during MD write orders.

c. Reading from the IBM. For this operation relays IBM, R_1 and others are energized. Each read brush from the reproducer is connected, via the summary punch cable, to an inverter which makes a positive-looking "1" coming from a read brush (hole in the card for a 1) into a negative-going "1" in order to feed the digit line in the correct sense. For reading into the machine from either the drum or IBM, the register R^1 is precleared to 1's (green gate transmits 0's) and the bumper strip triode is made to look like a diode. If a digit line is negative, then the output of its corresponding digit resolver is made to look negative, and

that digit in R^1 is not turned into a 0 at green gate. However, a positive digit line will cause its corresponding digit in R^1 to turn to a 0 at green gate. The network in the grid circuit of the read brush inverter insures that during all times other than an IBM read order, the inverter conducts and therefore presents a negative signal to its cathode follower, so that it is "out of the way" for other operations.

d. Reading from the Drum. For this operation relays R_1, $R_{R/WC}$ and others are energized. Reading a previously-stored "0" from the drum will result in a positive signal (and from a "1" a negative signal) at Green Gate time so that as in (c) above the correct information is gated into R^1.

To repeat then -- to read out of the machine, R_1 is cleared to 0 and after an RII Load operation the digit resolver output holds the number to be "taken out", either to be translated in the relays for IBM punching, or translated in the pulser for writing on the drum. For this process of reading out of the machine the plates of the bumper strip triodes are placed at plus 110 volts so that they act as cathode followers to drive the digit line according to the appropriate information.

To read into the machine, the output of the digit resolver is made to looklike zero (positive) so that information placed on the digit line by the read brushes of the IBM or the reading amplifiers of the drum, can be accepted into R^1 at green gate time, zig-zagged into R_1 and stored away in the Williams memory.

8. Sync Tracks. There are 1024 locations around every information track, at which information is stored. It is necessary to know when a head is exactly over any one of the 1024 spots so that we

may read or write at that spot, as the case may be, at exactly the right moment. To control this, a separate "sync" track (and head) is provided which has 1024 essentially equally spaced spots magnetically scribed around its periphery. The sync head signals when it is exactly over one of the 1024 "sync" spots, and it is only at these times that any reading or writing operations are done on the information tracks. That is, all tracks are so "tied together" that when an output pulse is obtained from the sync head we know that all other heads are in such position ("lined-up" on a complete 40-digit word) that we may read or write at that instant. The wire marked Sync 2ML (to the pulsers) is the output of the sync head amplifier after appropriate shaping and gating.

Just knowing when any spot is exactly under its reading head is not enough. We must also know when any particular spot is exactly under the reading head. That is, we must keep track of the addresses of every spot. This problem is discussed below.

Since the drum will be used to load or unload the higher speed William's memory it seems that the most flexible type of operation would be as follows: Allow the drum to have command of the machine (i.e. by code), reading out of, or writing into the William's memory a number of words N, starting at any position in the William's memory. This would be either to or from the magnetic drum, again starting at any position on the drum. The number N could be specified directly in the code or else an iterative routine could be made whereby one word is transferred at a time, the code then calling for the next word, etc. The first method would be difficult to handle because of the address requirements. That is, the order would require 10 digits to specify the starting William's

address, 10 digits to specify the starting drum address, and 10 digits (or less, depending upon how large N could be) to specify the number of words N. This would require the equivalent of three orders or one and a half words just to specify the one order. This would be very awkward to handle circuitwise. The second method indicated above could be handled in two ways -- neither of which seems too desirable. We could handle just one word at a time -- that is, operate on one word, then have the code determine the appropriate addresses for the next word, resynchronize the drum, etc. word by word. As noted above, synchronizing the drum for operation requires the switching of appropriate relays for either group of heads and for read or write. To synchronize for each word would not only be wasteful of time but would reduce the expected life of the relays by excessive cycling. However, for this method, the address requirements would not be so severe since complete specification of the order would require only 20 digits (10 for the starting William's address and 10 for the starting drum address). This could be rather simply handled circuitwise in two half orders or one full word. As an embellishment of the last method we could transfer words one at a time but maintain the synchronization between words if we had special orders to define the beginning and end of a transfer of N words by iterative code. The requirement of these special orders makes this method undesirable.

The method actually adopted is a combination of the above methods. It did not seem necessary to allow N (the number of words handled at one moment) to take on <u>any</u> integral value between 1 and 1024. Without much loss of flexibility it seemed as though we could restrict N to take on

only values which were some multiple of another number B, which we call a block. That is, the minimum number of words which can be communicated in either direction is a block B, but we can allow any number of these blocks to be transmitted at one time (up to a full memory load or 1024 words). Considerable discussion bore out the fact that a number B as high as 32 did not seem unreasonable. Using this figure, address requirements are now 20 digits: 10 digits to specify the starting William's address, 5 digits (i.e. to be able to specify any number from 1 to 32) to specify the starting block number on the drum, and 5 digits to specify the number of blocks to be handled at one time (again up to 32). This total of 20 address digits can be easily handled in two order spaces.

The magnetic drum now needs not only a track of 1024 synchronizing pulses around the drum (as noted above) but also something special to denote every 32nd word, to be able to detect the beginning of each new block. Also we must be able to identify the addresses of each block so that we may synchronize on the correct block. Two obvious ways of doing this are indicated in the sketches below.

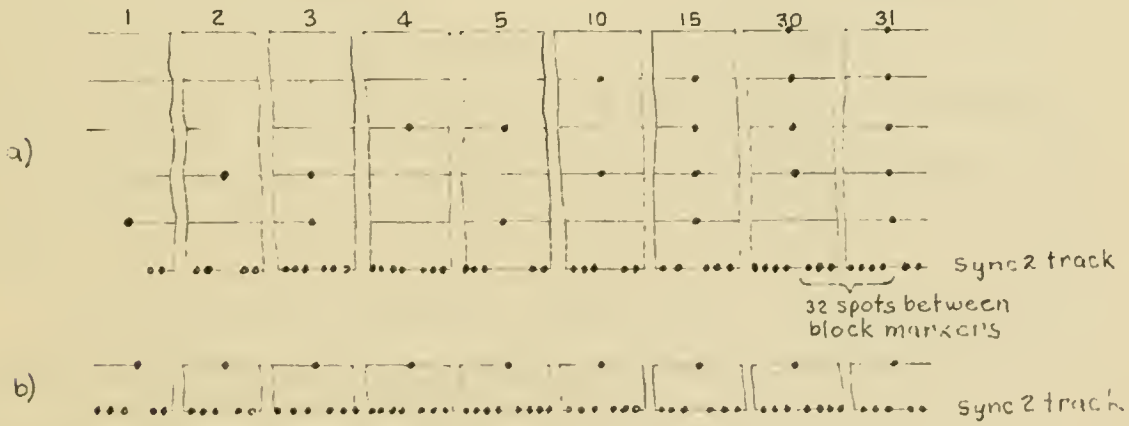


Figure 7

As indicated in the first sketch we may use five other sync tracks (besides the one mentioned above) with magnetic pulses so laid out along these five tracks that at the beginning of each new block the output of these five reading heads will have in binary form the number of the block. This information along with the number of the block to be recognized fed into a coincidence circuit would be sufficient for proper synchronizing. As indicated in the second sketch, another method would be to use one extra sync track with a marker magnetically scribed at the beginning of each block. These marks would be counted in an external counter circuit, so that the counter then keeps track of the block addresses. Component and circuitwise the selection of either system is not too exciting as it stands. However, one other fact enters which rules in favor of the latter system.

So far we have discussed only the requirements for the magnetic drum. However, the case for the IBM order is exactly the same. We read or punch only an integral number of cards, so that when we operate on N cards, we can say that we are handling N blocks of 12 words per block. To count the number of cards processed we could again either count the cards in an external counter, or we could put identifying numbers on each card. This latter system, however, is not only wasteful of card columns but also is inflexible in the sense that existing cards might be reshuffled into new decks in which case the number punchings become meaningless.

For this reason, an external counter system was adopted. Two sync tracks are required for the drum, then, one to define the position of every word around the drum (called sync 2) and another to define the

beginning of each block (called sync 1). In the reproducer the "Emitter" plays the role of the Sync 2 of the drum, in that every time one of the twelve positions of the card is in a position to be read or punched the emitter gives out a signal (there are twelve emitter signals per card). Another cam in the reproducer labelled P3 plays the role of the sync 1 track of the drum in that it signals the end of each card (or block). For a complete memory load we must use 1024/12 or 86 cards. To specify any number of cards between 1 and 86 requires 7 binary digits, so consequently a seven digit counter was used, even though only a five digit counter would be required for the drum. Note that in the case of the drum the counter is used twice per order. Timewise, the operation of synchronizing on the proper block comes before the operation of counting the number of blocks to be processed. Therefore, after synchronizing, the counter is cleared and used again for counting the number of blocks to be used. Of course, in the IBM case there is no equivalent process of synchronizing on the proper card, since we are forced to take the next card in the deck as it comes.

9. IBM and Drum Orders. We noted above that the complete drum or IBM order requires two complete machine orders (in order to buy enough address space). We shall briefly discuss the requirements and disposition of these orders. In what follows, for simplicity we shall refer mainly to the drum, although almost whatever is said also applies to the IBM.

Starting at a given William's address, the 32N (N = 1, 2, ..., 32) words operated upon will be written at or read from the consecutive Williams' addresses starting with the first given position. That is, at

any point in the process the next William's address will be obtained from the present address by adding a 1 to it. The present Dispatch Counter in the machine performs just this type function so that it was deemed desirable to use this unit for such a purpose. Normally the Dispatch Counter keeps track of the last position (plus one) in the memory which was regenerated, and also the address (plus one) from which the last order for the machine was obtained. In order not to interfere with the ordinary processes of regeneration, it was desirable to use the order rank of the Dispatch Counter for our purposes. The normal performance of an MD order will be for the machine to "stop", the external MD circuitry to take over control of the machine until the operation is completed and then revert the control to the normal machine channels so that it may go on automatically. Of course, for the machine to go on automatically it must have available a memory position from which to get its next order. However, use of the Dispatch Counter in the way indicated above would wipe out this memory of where to carry on from. One way around this is to separate the two orders required for one complete MD order into different words. The first order to appear would be a sort of priming operation (to be specified below) to prepare the magnetic drum circuitry so that when the second order appears later on it will be ready to completely execute the operation. This separation of the two orders into different words allows us to place the second order into the first phase of an order pair. This is followed in the second phase by an ordinary Transfer Control order, which allows the machine to carry on automatically after the completion of the MD order in the first phase, by allowing the machine to transfer control to the address accompanying

the transfer order.

So far then we have arrived at the following disposition of the two orders comprising a complete MD order. The two orders shall appear in different words; the second order to appear in the first phase of a word followed by a transfer control order; the first order to appear in some preceding word in a way not yet specified. From hereon in we shall refer to the first and second orders as orders A and B. We must have some combination of digits in orders A and B that sets them apart as MD orders. The second digit of the ten available for the order code was previously not used and had been arbitrarily made a 1 in all codes. It was decided to use this digit to specify MD operation, and further, in order not to obsolete the existing tapes it was specified that a 0 in this position shall indicate a MD order. The presence of this zero digit in order A and order B shall be required to initiate different functions; namely, for order A the priming operation (not yet specified) and for order B the actual carrying out of the operation. It would be easiest to detect these two functions based on the presence of the same zero digit in each order if the zero appeared once in one phase and then the other. Since the execution order is a first phase order, then the priming or A order is made a second phase order.

To distribute the 20 digits of address required between orders A and B, two possibilities would be to put the starting Williams address (10 digits) in the A order or the B order. The first possibility leads to far greater complications than the second. The priming or A order has no function in life other than to cause the priming address to be stored in a 10-digit register in the MD circuitry. If it were the

Williams address that was stored in the A order, then when we arrived at the B order we would nave to be able to communicate this address back to the Dispatch Counter while at the same time would have to communicate the address of order B to the drum circuitry for synchronizing to the new block, and detecting the proper number of blocks. Circuitry already exists in the main machine for communication between the first and second phase addresses and the Dispatch Counter (called B_3 gate) so if we put the starting William's address in order B, and the starting-block number, and number of blocks in address A, then all we need is the external 10 digit register to store the address of order A. Although it is true that communication also exists between the second phase address and the Dispatch Counter, if we put the William's address in order A and stored the address in the Dispatch Counter, then by the time we came to order B this address would have been wiped out. We have then the following disposition of the orders.

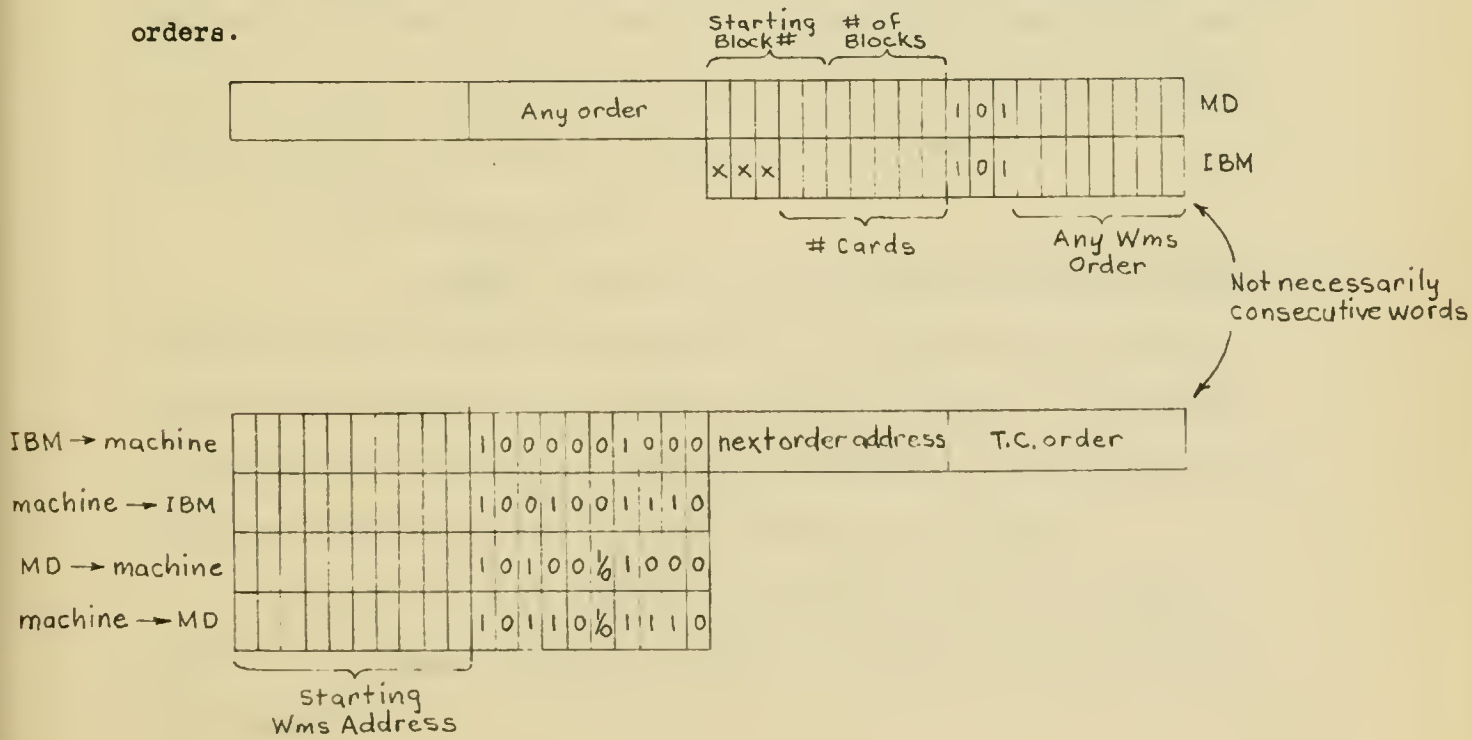


Figure 8

The particular digit combinations used to specify these orders were chosen so as to allow use of already-existing circuit facilities. The comparable IBM and Drum orders are identical except for the Wm/NWm digit which was so chosen that a "1" in this digit location specifies a Drum order and a "0" an IBM order. One further difference is the use of the L/R digit to specify, in MD orders, whether the first or second group of reading heads is to be involved in the order. Aside from these, the input orders for both IBM and Drum are identical. Also the two output orders are identical.

An input order is essentially an addition followed by a storage, repeated for each word transferred. The Summation digit sets up all the circuitry required for the addition process, and the combination of Summation and Trivial sets up all the circuitry required for storage.

An output order is essentially a series of RII Load orders. The combination of Non-Trivial, Summation and Round-off automatically sets up the circuitry required for carrying out RII load orders. The Clear digit provides for the proper clearing of R_1 to zero.

B. IBM and Drum Control.

1. General. There are four functions that the external circuitry must be capable of carrying out -- these are the "read-in" and "read-out" functions for both the IBM and MD units. The following discussion shall apply to the drum only, followed by a discussion of the additional features necessary for the operation of the IBM.

This will not be an exhaustive description of all the control circuits, but rather a guide for anyone who wishes to plow through all the circuitry.

First, let us state that the control distinguishes five main instants of time: (1) the initial request for a Drum order, (2) the first synchronizing instant when the "zero" position of the drum is under the reading head, (3) the beginning of information transfer when the drum reaches the first block to be dealt with, (4) the end of information transfer when we have gone through the last block to be handled, and (5) the shift back of control from the external circuitry to the main machine, for automatically carrying on the problem.

It would be too difficult and confusing to completely describe from beginning to end the operations involved in executing an order. Instead, each signal or unit as it would normally appear in the execution of the order is described in sequence below.

2. EX-G2 signal. The STEP and INTERNAL-EXTERNAL digits of the first phase order digits (G2 gate) are mixed in such a way that a 1 and 0, respectively, in these digits causes the EX-G2 signal line to go negative indicating the request for an external order (occurs for any of the four external orders).

3. RELAYS R1 through R6. These six relays operate in sequence for all external orders and provide a delay of six relay-operate times during which the appropriate head switching relays, etc., operate. The first relay R1 picks up at EX-G2 time.

4. ↓MD Signal. The William's digit in the first phase order is used to specify whether the external order is an IBM or Drum order. A 1 in this digit specifies a Drum order. This digit is mixed with the EX-G2 signal, so that a 1 and negative signal, respectively, cause the ↓MD line to go negative hence initiating a Drum order.

5. $\overline{\text{IBM Signal}}$. A 0 in the William's digit specifies that the external order is to be an IBM order, and the $\overline{\text{IBM}}$ line goes negative for this case at EX-G2 time, initiating an IBM order.

6. R/NR Digit. Note that in the list of external orders described previously, a R/NR (Roundoff/No Roundoff) digit of 1 implies that information will be read out of the machine to the Drum or IBM; a 0 implies that information will be coming in.

7. L/R Digit. This digit is used to specify (for Drum orders only) which group of 40 heads on the Drum are to be used for that particular order.

8. $R_{1/0}$, $R_{R/WC}$, $R_{W/RC}$, and IBM Relays. The EX-G2, MD, L/R, and R/NR (actually inverted and called $\widehat{R/NR}$) signals are mixed to energize the appropriate master relays; the $R_{1/0}$ relay energizes for the use of the group-1 heads, and is de-energized for the use of the group-0 heads. The $R_{R/WC}$ Relay is energized for Reading from the drum and is de-energized for Writing and normal Compute. The $R_{W/RC}$ Relay is energized for Writing onto the drum and is de-energized for Reading and normal Compute. The IBM relay is energized for all IBM orders. Note that the "safety" relay has no logical function except to prevent the $R_{W/RC}$ relay from operating (and hence preventing writing power from being applied to the heads) at any undesired moments, such as during power turn off and on, etc.

9. Seven-Digit Binary Counter. (Similar in design to the Dispatch Counter in the main machine, which is described in a previous report.) This counter is used to count Sync 1 pulses (i.e. blocks in the case of the Drum and cards in the case of IBM). The counter is used

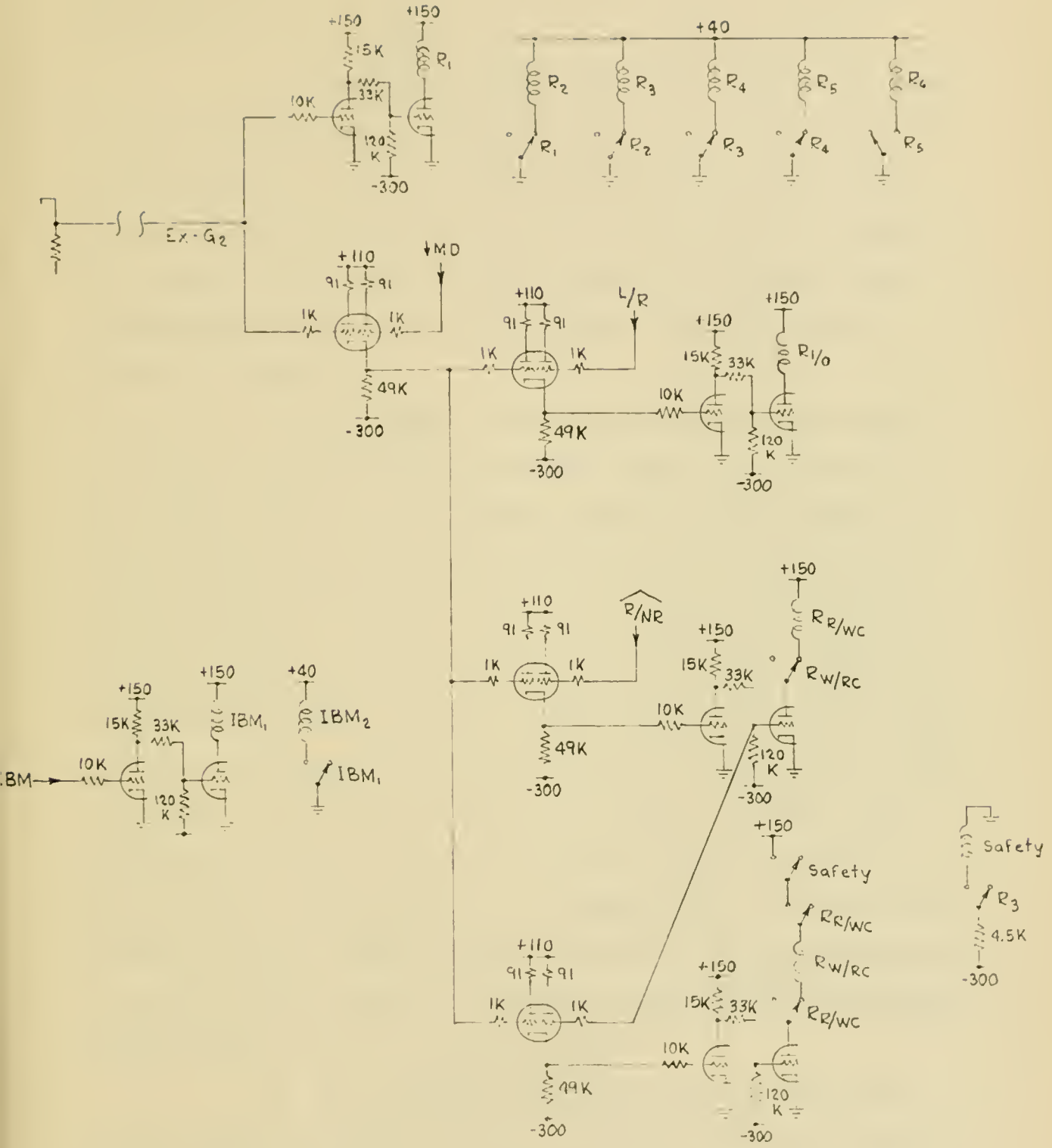

+150
15K
33K
10K
120 K
-300
R1
+40
R2
R3
R4
R5
R6
Ex - G2
+110
91
1K
M.D
49K
L/R
R1/0
R/NR
RR/WC
RW/RC
Safety
IBM1
IBM2
4.5K

Figure 9

twice per MD order, once to synchronize to the proper starting block, and following that, to count the number of blocks being handled.

10. Coincidence circuit. This circuit like wise is used twice per MD order; once to compare the output of the first five stages of the counter with the starting block number stored in the 10-digit register (and issue a signal at coincidence), and secondly, to compare the counter with the second five-digit number in the register (which specifies the number of blocks to be handled), and again to issue a signal at coincidence.

11. T_{STT} Toggle. The Recognition Circuit noted above is of a simple design which allows coincidence signals to be issued for many numbers after the first legitimate signal (but never before). On account of this feature we must insure that the counter always starts counting from zero, so that its count is always less than (or equal to) the desired number (in order not to get any false signals). After the completion of the Relay Delay time (signified by a point of the R5 relay operating in the T_{STT} turn-on circuit) we know that we are ready to start operations at will, so that at the very next Sync 1, 2 signal (which states that the drum is at its zero position) we clear the counter to zero and turn on the T_{STT} toggle. This toggle in turn, opens the path between the output of the coincidence circuit and the T_{BLOCK} toggle turn-on circuit so that at the first coincidence signal (the drum at the proper starting block) we turn on the T_{BLOCK} toggle.

12. T_{BLOCK} Toggle. This toggle actually controls the digit information transfer circuitry. It comes on as indicated above and stays on until the second coincidence signal is obtained which

indicates that we have completed the required information transfer.

13. $T_{BO/3}$ Toggle. As indicated before, the Dispatch Counter in the machine controls the Williams addresses associated with the information transfers. The first Williams address to be handled comes to the Dispatch Counter from the address digits of the first phase order via a B_3 gate signal. The next Williams address is obtained by adding a one to the first address via B_0 signal operation. All following addresses are obtained in the same way, that is, for any external order there occurs one and only one B_3 signal. By the time that the B_3 signal is obtained the counter and recognition circuits have completed their first task (i.e. synchronizing to the proper starting block), so that this signal is piped over from the main machine to the drum and is used to clear the counter to zero and to turn on the $T_{BO/3}$ toggle. This toggle has several functions, one of which is to "swap" the input of the Coincidence Circuit from the "left" five digits of the priming register to the other five digits of the register which specify the number of blocks to be handled. It also controls the routing path of the coincidence signal from the compare circuitry. That is, with the toggle OFF the coincidence signal, when it comes the first time, is used to turn on the T_{BLOCK} toggle. With the toggle ON, however, the routing path is switched so that the second time a coincidence signal is obtained (at the completion of information transfer) it is used to turn off the T_{BLOCK} toggle.

14. Null Order Gates. At the unique combination of $T_{BO/3}$ ON and T_{BLOCK} OFF a Null Order Gates signal is sent to the machine which kills the request for either the G2 or G4 order gates (which are used to

read the first or second phase orders into the main control). This in turn kills the EX-G2 signal requesting the external order, so that the six Delay Relays begin to fall back to normal. When the fifth relay has returned to its original state it turns off the T_{STT} toggle which in turn turns off the $T_{BO/3}$ toggle. The turning-off of this toggle kills the Null Order Gates signal, so that the main control in the machine can now read the second phase Transfer Order and carry on the rest of the problem. Note that during the Null Order Gates signal the main control internally "switched" the order request from the first to the second phase.

15. Reading into the Williams Memory. As described in a previous section, the digital information enters the machine at the Digit Resolver. It is arranged that at these times the digit resolver output is zero so that the input information overrides the resolver output. This information is then "accepted" into RI upper and zig-zagged down to RI lower, from which place it is stored. As noted before, at Sync 2 time the digit-information lines hold the proper information from the drum (voltage-wise), so that to accept it into RI it is only necessary to turn on the T_{BLOK} (in the main machine) toggle; this in turn with the Summation = 1 digit of the Read-in order, is enough to carry out automatically the normal arithmetic processes of addition. When the information has reached RI lower we are ready to request a storage of it at the proper Williams address. This is accomplished by piping over to the drum, the shift counter satisfy signal (which signals the end of the arithmetic process) and using it to turn on the T_{YES} toggle of the drum, which in turn issues a Williams request to the main

machine. The memory Sync signal which signals that the Williams request has been accepted by the Williams Local Control, is piped back to the drum to turn off the T_{YES} toggle, so that only the one Williams request per word is obtained. The next Williams request comes when the next word is brought up from the Drum and is accepted into RI, etc.

To insure that the output of the digit resolver is zero (internally) it is necessary that RI lower be zero and that either RIII upper is zero or the complement gates are nulled. The same signal that is used to turn on the T_{BLOK} toggle (at drum Sync 2 time) is used to preclear RI lower, which wipes out from this register the previously-read number. Note that this is an early enough time to clear RI since Green Gate (which accepts the information into RI) does not occur until about 15 microseconds after T_{BLOK} turn-on, because of the imposed Carry Delay. The RIII condition is taken care of by nulling the complement gates.

One other factor that must be taken care of in the read-in process is the Shift Counter Clear. Every time the Memory is used the Sync Signal clears the Shift Counter to zero in preparation for the next arithmetic process. However, in the read-in order, since an arithmetic process occurs before any memory process, we must separately insure that the Shift Counter starts out cleared to zero. This is accomplished by the separate circuit shown in Figure 10.

16. Writing Onto the Magnetic Drum. As noted previously, at Sync 2ML time the Pulsers write onto the drum whatever information is present on the digit-information lines. During external orders that take information out of the machine the information wires hold whatever information is residing on the output of the Digit Resolver. The operation then

is to place on the output of the digit resolver, at any instant, the correct word to be written onto the drum, call for a Sync 2ML signal, and at the completion of that to call for the next word to be brought out of the Williams memory and be made to reside at the output of the digit resolver in preparation for the next Sync 2ML signal. However, this case is slightly different from the case of reading <u>from</u> the drum, in that a machine operation must precede the first drum operation, i.e. the correct number is sitting at the digit resolver ready for the first Sync 2ML signal. In the case of reading from the drum, the first Sync 2IMS signal from the drum calls for the first main machine arithmetic operation. For writing on the drum then, as soon as T_{BLOCK} comes on, we must ask for a Williams request (as indicated in the timing chart on the drawing), and from then on get Williams requests only at the end of each Sync 2ML signal.

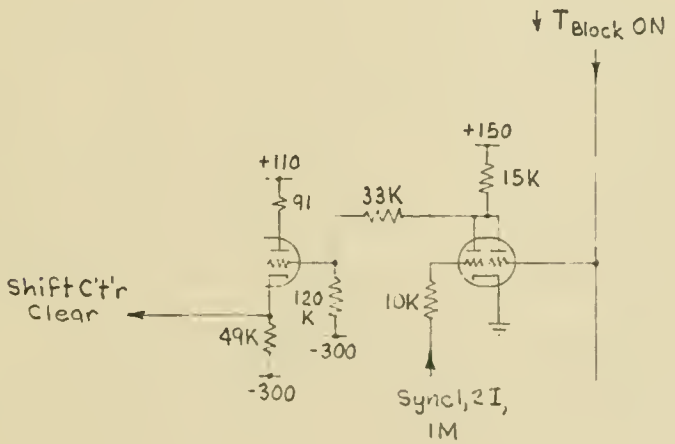


Figure 10

17. Special IBM circuitry. Operation of the IBM unit is not as flexible as the operation of the Drum in that one cannot synchronize at will on any card with which to start the operation (as one can call for any block with which to start a Drum operation). In view of this fact, after T_{STT} toggle comes on we turn T_{BLOCK} toggle on immediately instead of waiting for the first counter-satisfy signal (as we do for the Drum). This is accomplished by the extra diode which via the ↓IBM signal opens the path for the T_{STT} toggle to turn on the T_{BLOCK} toggle as soon as T_{STT} comes on.

The Sync 1 and Sync 2 signals for the IBM are obtained from the Cam P3 and Emitter, respectively (their relative timing is shown in the Drawing No. 0-1544). Because the Sync 1 signal comes only after the 12 Sync 2 signals for a card, the 7-digit block or card counter was arranged so that Sync 2 times the counter counts from the False to the True rank. Therefore after the very first Sync 2 signal of a card or block, the counter holds the proper count. This is important in the case of the IBM since in order to stop the Reproducer on the right card, the STT relay must be dropped out before Cam P5 time. The Cam P5 timing and the STT relay circuit are also shown on the Drawing No. 0-1544.

The STT relay is "picked up" for every IBM operation by the IBM relay. It remains energized throughout the order by one of its own points and the 5687 tube. The grid of this tube is kept positive throughout the entire order until the very first time that the Cam P5 sensing signal is accompanied by a counter satisfy signal at which time the holding triode grid goes negative causing the STT relay to drop out, and no cards beyond the one in process are fed through the unit.

18. Sync Signals. There are various sync signals used throughout the unit. The code used for the name of these signals is as follows:

I stands for IBM
M stands for Magnetic Drum
S stands for Short pulse
L stands for Long pulse

During a Drum or IBM order Sync 1; Sync 1,2; and Sync 2 (long and short) pulses are generated. The lengths of these signals are approximately as follows:

	Drum	IBM
Sync 1,2	20 microseconds	20 milliseconds
Sync 1	20 microseconds	20 milliseconds
Sync 2S	2 microseconds	2 microseconds
Sync 2L	40 microseconds	15 milliseconds

In the Sync Chassis these signals are mixed to form the following composite signals:

Sync 1 IM - This wire carries Drum sync 1 pulses during a Drum order, and IBM sync 1 pulses during an IBM order

Sync 2 IML - Similarly for the long sync 2 pulses

Sync 2 IMS - Similarly for the short sync 2 pulses

Sync 1,2 IM - Similarly for the sync 1,2 pulses

Sync 1,2I;1M - This wire carries Drum sync 1 pulses during a Drum order, and the IBM sync 1,2 pulse during an IBM order

Sync 2 ML - This wire carries only the long Sync 2 Drum pulses during a Drum order.

NOTE: It is proposed to build a graphing unit capable of transcribing

from the magnetic drum to an external cathode ray tube for the purpose of plotting (graphing) computed information. This shall work automatically during machine computation, and shall be in spirit a pure accessory, not interfering with machine operation. Because the necessary equipment is neither built nor even completely designed as yet it is impossible at this time to give an exhaustive report on it. However, at a very few places on the drawings references to a graphing function will be seen.

C. Cathode Ray Tube Testing Program.

1. The Williams memory uses standard 5CP1-A cathode ray tubes selected from regular manufacturer's stock. This selection is done in our laboratory with the very helpful cooperation of both Allen B. DuMont Laboratories and the Radio Corporation of America. Such selection is made practical because the qualities tested for have no effect on ordinary usages.

All tubes received are first subjected to a "flaw" test. Flaws are local inhomogeneities in the phosphor surface, probably due to minute foreign particles, which reduce the dash signal available from the point.

Flaw Location

A flaw location test is used first to discover where on the phosphor surface the flaws, if any, are situated. Figure 11 shows a block diagram of this test. Note that the beam current of the tube under test is held constant and that the deflected spot will in general trace an ellipse. If the path of this ellipse changes very slowly between successive tracings and if the secondary emission properties of the phosphor struck by the beam are constant, we expect no A.C. signal

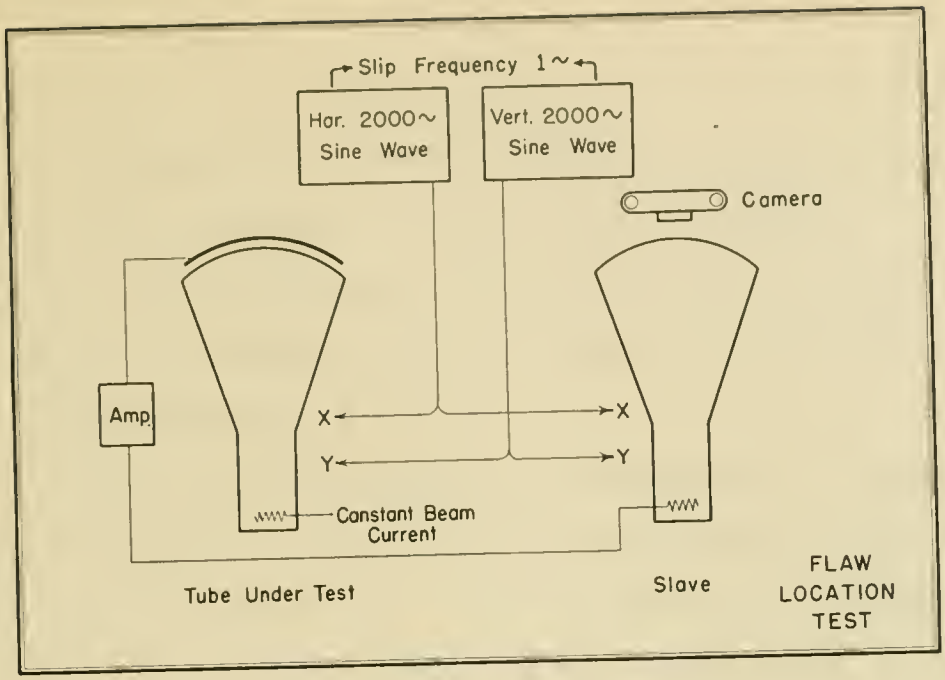


Figure 11.

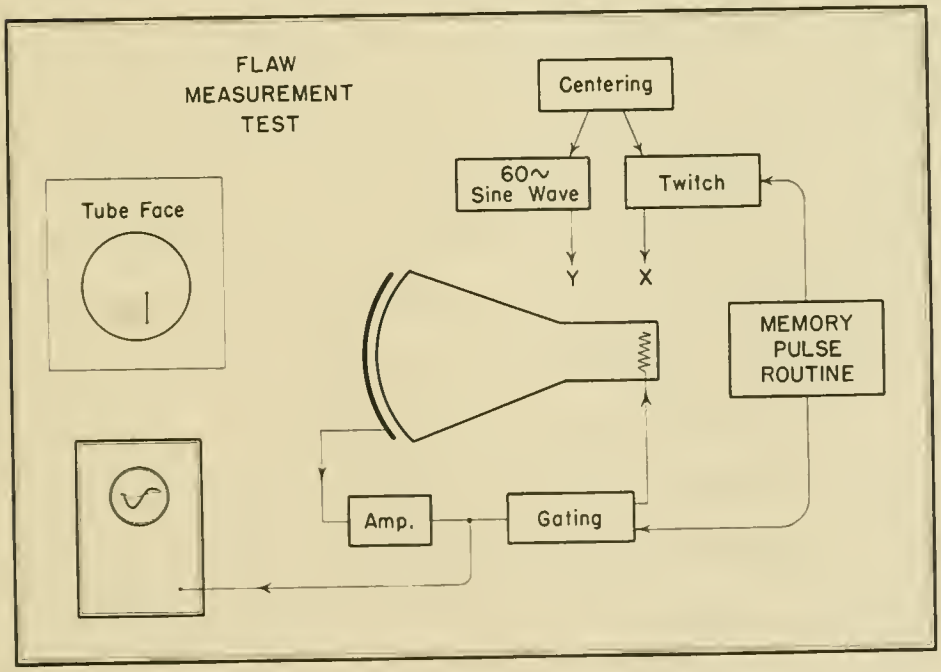


Figure 12.

at the external pickup screen. However, if a discontinuity in secondary emission is encountered, an output signal will be obtained and can be used as shown to intensify the trace of a slave viewing tube. A small frequency difference between horizontal and vertical sweeps will cause the ellipse to precess and cover every point on the phosphor within a rectangle (approximately 3" x 3").

For maximum detail a slow rate of precession is desirable and a camera is necessary to integrate the slow sweep into a continuous picture. For the initial test, however, visual observation at a somewhat greater sweep rate is sufficient to locate the more severe flaws. These are marked on the face of the slave tube with a grease pencil.

Flaw Evaluation

Once the flaw locations are known they can be measured by storing a dash on the flaw and comparing the signal obtained with the normal signal. The need for very precise preliminary location can be obviated by the technique shown in block diagram form in Figure 12. The standard routine for storing a dash is applied to the grid and horizontal deflection of the tube under test while a 60 cycle sine wave of variable amplitude is applied to the vertical deflection. This results in a stretched out dash covering all points along a vertical line. With the use of D.C. centering controls the "line of dashes" can easily be centered to pass through a given marked flaw by watching for the partial output signal deterioration on the oscilloscope. The magnitude of the flaw is then given as the ratio of flaw signal to normal signal. Thus an 0.8 flaw gives 80% normal amplitude.

Spill Test

A certain number of new tubes are immediately rejected by this initial flaw test; these are given no more processing. The survivors are next subjected to a spill or read-around test. A block diagram of this test is shown in Figure 13. The set up is actually a complete one tube memory system with all points of a 1024 raster uniformly regenerated and with provision for variable frequency action references to five selected points -- typically the four corners and the center. The raster is cleared to dots, dashes are stored at the five points, and then each of the five bombarded with increasing frequency until one or more of the neighboring dots changes to a dash. The failure frequency at a point determines an arbitrary read around figure for the point according to the following relation:

$$R.A. = \frac{\text{action frequency}}{\text{clock frequency}} \times 1000.$$

Flaw Re-test

Additional tubes are rejected by the spill test. The survivors are again subjected to a more thorough flaw test in which a photograph is made of the flaw scan and becomes a part of the records on each tube. Figure 14 shows a typical flaw scan photograph.

Sparking

Some flaws can be removed by a sparking technique first described by W. E. Mutter.[1] A small tesla coil, or vacuum leak detector, is used to apply a spark to the face of the cathode ray tube. The method found

---

1 W. E. Mutter, "Improved Cathode Ray Tube for Application in Williams Memory System", Electrical Engineering, Vol. 71, April 1952, pp. 352-356.

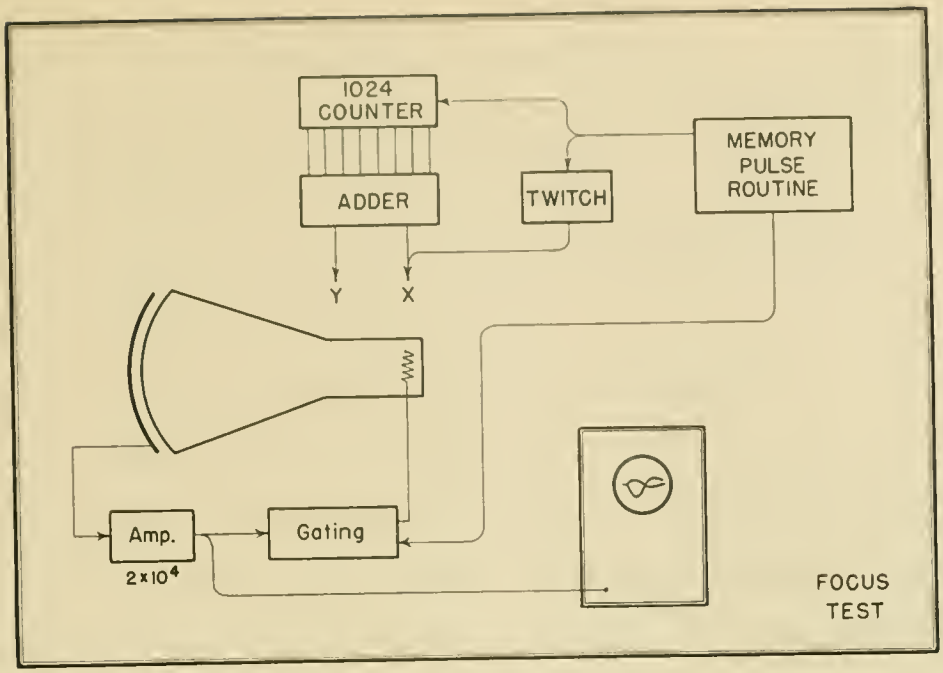


Figure 13.

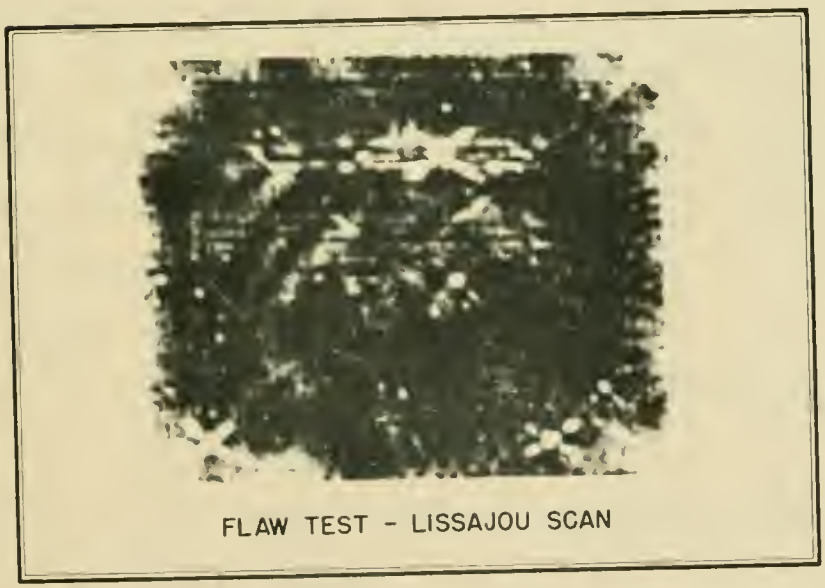


Figure 14.

most effective in this laboratory is to place the tube in its carton with the third anode connection grounded and to sweep the spark across the entire face according to the following routine:

1) a zig-zag or television type scan from side to side,

2) a spiral from outisde to inside,

3) a spiral from inside to outside, and finally

4) a zig-zag from top to bottom.

The sparking technique has been found most effective for tubes initially having quite numerous flaws. The tubes obtained in recent months have relatively few flaws, most of which are not improved by sparking, so this technique is not regularly used.

Screening

Tubes which have passed all the foregoing tests are prepared for eventual memory use. This preparation includes application of the external pickup screen, cleaning and inspection of base pins, and installation of a short connection wire to the third anode snap button. The screen application process used in this laboratory results in a good permanent bond and is set forth in detail here for those interested:

Williams Screening Technique

a. Prepare screens.

i. 3 7/16 x 3 1/4 80 mesh brass screens with small tab on 3 7/16 side.

ii. Heated in air atmosphere oven until they turn blue, to remove temper.

iii. Allowed to cool slowly in room temperature.

iv. Washed in diluted HNO_3.

v. Washed in $NAHCO_3$ solution.

vi. Rinsed in water and dried.

b. Tubes are cleaned with soft rag and GE glyptal thinner #1511F.

c. Three consistency glyptal solutions are prepared from thinner and #1286 GE glyptal.

i. Sol. 1 for screens, very thin such that when screen is dipped and drained, no cement remains in interstices.

ii. Sol. 2 is slightly thicker, to be painted on tube face.

iii. Sol. 3 is rather thick, being used to seal screen edges in final step.

d. i. Soak screen in a pan of Sol. 1.

ii. Paint a base of Sol. 2 on tube face and let dry.

iii. Place tube in test rack and adjust raster to test screen size.

iv. Place wet screen on tube.

v. Apply a domed, large coarse mesh screen and place over this two 1/2" sponge rubber mats with a 1" central hole for air circulation.

vi. Apply pressure plate to this sandwich. Pressure plate has central 1" hole with small blower attached.

vii. In 10-15 minutes cement is dry and tube is removed after screen centering is checked by raster.

viii. Excess cements is cleaned off by thinner.

ix. Fine emery cloth is used to buff screen surface.

x. Sol. 3 is used with a small brush to seal off any loose screen edge.

xi. Tab is pulled up for soldering.

1. Acceptance criteria.

The current acceptance limits on the foregoing tests are as follows:

Flaws: 0.75 to 1.0, where the focus of the tube under test is adjusted to maximize dash degradation, i.e. give a minimum valuation.

Spill: R.A. = 100 or more for the five points tested. This acceptance limit is not a complete guarantee of satisfactory spill performance in the memory since it is based on only five points out of 1024. However, it is not feasible to apply the more exhaustive test to all tubes received.

2. Test results.

More than 1000 5CP1A's of various manufacturers have been tested from the beginning of the memory program. The original 40 tube complement in the memory was selected from a group of 86 obtained from surplus. These were surprisingly good, in fact, 10 out of the 40 are still in service. However, operational test and experience brought about a tightening of the acceptance specifications to the present limits.

Detailed results on a recent group of 250 DuMont 5CP1A tubes are given in the following tables:

Total tubes in this test	250	
less preliminary rejects	-6	
Total given flaw tests	244	
less flaw and other rejections	-174	
Total given R.A. test	70	
less R.A. rejects	-22	
Total accepted	48	(19%)

Table 1.

Total test results on 250 new 5CP1A's

Flaw values:	0-0.5	0.6	0.7	0.8	0.9	1.0
# of tubes :	75	36	60	30	22	21

Table 2.

Flaw Test results on 244 5CP1A's

R.A. :	<100	100	200	300	400	500
# of tubes:	22	23	14	7	0	4

Table 3.

R.A. Test results on 70 5CP1A's

3. Tube Life.

As of 1 July 1953, 10 of the original complement of 40 tubes were still in the memory with an active (B+ on) metered service of 7500 hours plus a few hundred hours of initial service before metering began. Removals from the memory were mainly for improving the read around performance and margin with respect to flaws. These data are summarized in tables 4 and 5.

No. of tubes	Hrs. In Service
10	7500
11	5000
15	2500
4	< 2500

Table 4.

CRT's in Memory - July 1953

No. of tubes	Reason for Removal
16	Flaws
29	Spill
5	Emission failure
6	Miscellaneous
56	

Table 5.

CRT's Removed from Memory to July 1953

IV. MAINTENANCE.

A. Introduction.

The maintenance program for the machine is intended to maximize productive operation time by minimizing time lost due to troubles. The scheduled maintenance program aims at detecting weaknesses before they cause trouble and detecting actual troubles before computation begins. Troubles not so predicted or detected must be corrected by unscheduled maintenance as they occur.

It appears neither possible nor desirable to reduce unscheduled maintenance to zero since this would require an excessive amount of routine test time. Conversely it seems very undesirable to reduce the routine time to zero since some troubles may not be immediately detected by the calculating group. Such a policy would lead to much lost time at best and at worst to multiple malfunctions which greatly complicate trouble shooting.

The maintenance policy chosen devotes a period of 1 to 2 hours at the beginning of each day to a series of routine tests. The nature and duration of the tests has been and will continue to be evolved according to operating experience. Other tests are performed at less frequent intervals such that a fairly comprehensive review of machine performance is obtained about every three months. The results of both scheduled and unscheduled maintenance are used to single out unreliable components for redesign or replacement.

The most frequent, i.e. daily, tests concern the known most critical components and parameters. The arithmetic unit and the arithmetic processes in the memory were designed, as described in the final

report, to be as truly binary (go-no go) as possible with carefully chosen margins on component and tube variations with the result that in practice the routine arithmetic testing is confined to coded check problems (Diagnostic code, Arithmetic Test code).

Certain portions of the memory, however, have a proportional rather than a binary response. In the analogue deflection, or beam switching circuits, a 50 volt swing per deflection plate is necessary to produce a 32 step "x" or "y" display but a noise of 0.2 volts, producing about a 1/2 spot diameter disturbance, can cause an error. Such noises do not occur as an unavoidable random background but rather will be entirely absent for a period of weeks and then appear quite strongly as a result of some more or less evident component failure. Therefore a short check is made each day for memory noise. (Diagnostic code.)

The video amplifier-beam turn on loop for the memory tubes also has a proportional response in which the D.C. state is of particular interest. Conflicting requirements of read around and storage reliability necessitate a careful adjustment of the focus, beam current, and astigmatism parameters for each tube. Drifts or perturbations in these parameters degrade the safety margin of the memory. Daily limit checks and programmed tests are used to evaluate the centering of these parameters. (Discriminator bias test, Read around test code.)

Other routine tests are made on the machine supply voltages and input-output equipment.

B. Scheduled Maintenance.

1. Routine Tests.

The routine tests are performed once every 24 hours at the beginning of each daytime shift. If the machine is not running a continuous schedule, this time coincides with turn-on for the new day of operation. These tests comprise the following list in approximate time sequence.

a. Power and heater checks.

i. Turn on computer heaters and check heater voltages. The machine heater circuits are made separate for each logical subgroup such as "Accumulator toggles", "Accumulator Gates", etc. and each circuit is provided with a secondary voltmeter and a primary variac, circuit breaker, and ammeter. The gross functioning of each heater circuit is checked at this time.

ii. Turn on main high current D.C. power supplies. These are thyratron regulated units which furnish all the D.C. for the machine either directly or through auxiliary regulators. Their voltages at the machine are checked to be +380, +240, +110, and -300 volts, respectively, within meter observation error. No day to day variation is ordinarily found in these voltages.

iii. Turn on computer D.C. while checking for the appearance of a raster on the memory tubes. A majority of the important machine voltages enters into the production of the memory deflection. Observation of the raster gives both a gross check of the power system and protects the memory tubes against thermal overloading of the phosphor due to an undeflected electron beam.

iv. Observe with an oscilloscope the ripple voltages on the +380, +240, +110, and -300 volt thyratron supply outputs. About

3 volts peak to peak is the normal ripple for the heavily loaded -300 volt supply and about 1 volt peak to peak for the others. The ripple magnitude is a check on the dynamic regulation of the supply, and the relative magnitudes of the six ripple components gives a check on the balance of the thyratrons and of the phase voltages.

v. Auxiliary regulators of the conventional series type are used to supply moderate current plate and reference levels. In general, several triode units are connected in parallel to pass the required current. One regulator each day is checked for approximately equal current sharing among the paralleled units as a way of detecting emission deterioration.

vi. Read and record A.C. and D.C. running time meters.

vii. Check memory inspect (or "strobe") pulse. The effective duration of this pulse is important in determining the read around performance of the memory. Duration and voltage limits of this pulse are measured at the distribution point with a D.C. oscilloscope.

viii. Observe and record bleeder current meter reading. All memory CRT adjustment voltages -- astigmatism, focus and beam intensity -- are derived from the second anode high voltage bus by a bleeder network. The current through this bleeder is indicated on an accurate meter as a check on resistor or second anode supply voltage drifts.

ix. Observe and record "twitch" current meter reading. Twitch refers to the separation of the double dots used in the Institute for Advanced Study memory to store a binary "1". The switched current which ultimately produces this separation is made adjustable to permit a best average setting for the population of 40 tubes. Drifts in this

current will result in read around deterioration.

b. Operation checks.

i. Discriminator bias test. The discriminator for each memory tube contains a time and voltage sampling gate system. If the time parameter is left undisturbed, the voltage gating feature may be used as a form of peak voltmeter. Figure 15 shows the normal dot and dash output signals as solid lines along with the normal time "inspection" period.

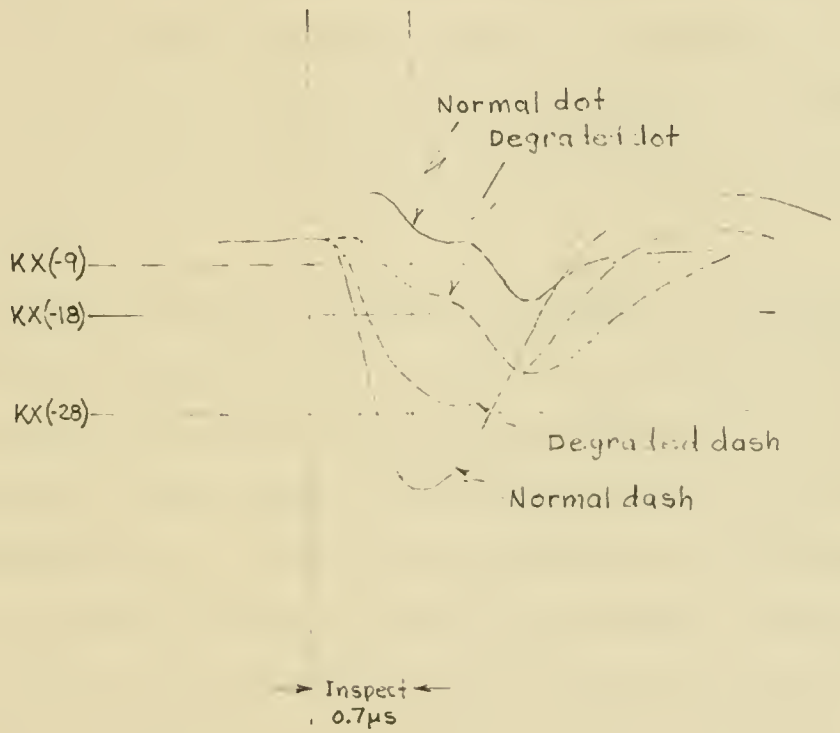


Figure 15.

A typical normal voltage sampling level is shown by the horizontal solid line marked Kx(-18). The attenuation factor K is determined for each individual stage by the bias potentiometer setting.

If the common bus supplying all potentiometers is varied about its nominal potential of -18 volts, a very useful limit check is obtained. For example, the voltage can be increased negatively until some or all dash signals fail to work the gate; and it can be increased positively until some or all dot signals are interpreted as dashes (i.e. work the gate). More restrictively, a negative perturbation can be chosen such that the weakest tolerable dash signal just works the gate and a positive perturbation chosen such that the poorest tolerable dot signal just fails to work the gate. Once determined, these limits can be used as a rapid check for unfavorable changes in amplifier signal output, since any decrease in output will cause some dashes to become dots at the one limit and any increase will cause some dots to become dashes at the other limit.

In practice, this test is performed dynamically. An 80 digit word is circulated through RI and RII by means of the regular $2^{0}R^{1}$ --> $2^{-39}R_{2}$ left end-around gate and an auxiliary (test only) $2^{0}R^{2}$ --> $2^{-39}R_{1}$ left end around gate. Store orders (using addresses from the order counter) alternated by 1 S. L. orders are executed at a rate sufficient to traverse all points in the memory about once per second. The 80 digit word is made repetitive modulo 10 so that every 10th memory stage receives the same information. The slave viewing tube is provided with a comparing function such that with any two chosen stages as inputs, the beam "brights up" only if the two inputs are unlike. With the slave observing, say, stages 0 and 10 and with the store-shift left routine running, setting the discriminator bias bus to its negative limit (-28 volts) and then to its positive limit (-9 volts) should in neither case

result in any disagreements, or errors, as shown on the slave. Actually, the upper (-9 volts) limit is set first and all comparisons made: 0 with 10, 20, 30; 1 with 11, 21, 31, ...; 9 with 19, 29, 39. Then the lower (-28 volts) limit is set and all comparisons made again. The comparing of equally perturbed stages does not lead to trouble since even equally weak stages will rarely fail at precisely the same points.

For the bias limits chosen, no errors should be observed. Any stage showing an error is given individual attention.

ii. Flaw scan. The discriminator bias test is intended to check the major parameters which influence the dot and dash signals: beam current and amplifier gain. The flaw scan is a more specialized test directed at the local inhomogeneities of the storage surface known variously as flaws, blemishes, or phoneys. The dash, or "1", signal from a flaw is smaller than normal and may be zero for a severe flaw. To test for flaws a word consisting of all "1's" is stored at each point in the memory using the address from the order counter. The discriminator bias is set at its negative (-28 volts) limit to make dash recognition most difficult. The memory is cleared to zero and the progressive writing in of 1's is begun. The slave is again used to compare the stage under observation with some other one. The sequence: clear to zero, write in 1's is repeated 10 times for each stage with the requirement that no failures to store 1's shall be observed. If there are storage failures, the raster must be shifted to avoid the flaw.

iii. Flaw survey. With all 1's stored in the memory, the superimposed 1024 dash outputs of each stage are examined on an oscilloscope for isolated low amplitude dashes. Even though such dashes will

have passed the bias and flaw scan tests they are considered undesirable because a small raster drift may worsen them. Within limits, the raster is shifted to avoid them.

iv. IBM tests. The IBM tests begin with a check on the verifying (or comparing) circuits which should normally indicate by column any discrepancy between a card from the read hopper as read by the comparing brushes and the corresponding card from the punch hopper as read by the punch brushes. The memory is set to all 1's (or as left from test iii) and punched out while verifying against a deck of cards consisting of words (i.e. rows) alternately all 1's and all 0's to achieve maximum loads. All 40 active columns in the comparing unit should show errors. This test is repeated as above but with the memory set to all 0's. Failures to verify have been found due to card feed misalignment and dirty or damaged brushes. Maintenance work on the IBM machines proper, other than clearing card jams, must usually be done by the IBM field engineer.

If the verification test is satisfactory, the alternate 1's and 0's deck described above is loaded into the memory where it should form a simple geometric pattern consisting of alternate columns of dots and dahses. The correctness of loading into any one stage is checked by observing this pattern and then the other 39 stages checked, using the slave, by comparison with the first. If the loading was correct, the memory is punched out while verifying against the input deck.

v. Arithmetic Test. The arithmetic test code (see appendix) is loaded and allowed to run for a minimum period of five minutes. If arithmetic troubles are suspected, the code, or a portion

thereof, is run longer.

vi. Read Around Test. A read around test code, such as code #134 (see appendix) is used to test every point in the memory for read around rates beginning with 16 and increased by increments of 2 up to 32. Read around failures are printed out on cards as they occur. At present the minimum failure level is 24 for the worst tube. Any tube failing below this level is readjusted or replaced.

vii. Diagnostic Code Test. A composite input consisting of five complete 86 card diagnostic code decks is loaded automatically. After the fifth deck is loaded, the code is allowed to proceed at full speed for a minimum period of five minutes or until the mathematical operation begins for the day. This code is a meteorological problem in which essentially all computations are done two different ways and required to check precisely. Also the data field is periodically summed and compared with an established correct sum. The orders and data occupy the entire memory. Either memory noise or arithmetic errors will produce an error stop.

2. Non-routine tests.

In addition to the daily tests which are intended to check the operability of the machine at a given time, a less frequent class of tests is made at intervals of one to three months which are intended to check the safety margins or range of operability.

The most effective of these tests consists in a detailed set of voltage measurements made on the registers, adder, and digit resolver for the several possible per-stage digit combinations. These measurements reveal both resistor drifts and slow tube deterioration very clearly.

The other infrequent tests consist of qualitative oscilloscopic observations of the dynamic behavior of the machine, particularly the 360 gates associated with the registers. Here the interest is mainly in looking for large deviations from the average behavior.

C. Unscheduled Maintenance.

The time devoted to unscheduled maintenance has decreased rather steadily through the period covered by this report; partly due to the adoption of the routine procedures already described and partly due to circuit improvements. Apart from very infrequent tube failures in the arithmetic unit, the major part of the actual down time is chargeable to the memory. The principal trouble with the memory has been with random errors, or noise pickup. Such noise has usually been very infrequent for a period of weeks, then has risen to a relatively high frequency of occurrence, and has in each case been traced to a faulty component or components other than the cathode ray tubes themselves.

Probably the most significant aspect of unscheduled maintenance is the proportion of total machine operating time it consumes. Basic time records have been kept for the machine since 27 October 1952. These are presented in table 6. The categories listed are defined as follows:

Routine: All time spent in performing the tests described in "routine maintenance" elsewhere in this report.

Engineering: All time spent in engineering improvements or additions to the machine.

Unscheduled Maintenance: All time spent in diagnosing and repairing

Table 6.

Week ending	BASIC RECORDS: Routine	Engineering	Unscheduled Maintenance	Operation	Total	DERIVED: (UnM + Op) Available	(Op / Av) % Operation
31 Oct. 52	14.5	32.5	6.0	26.0	79.0	32.0	81
7 Nov.	11.0	22.5	6.0	41.0	80.5	47.0	87
14 Nov.	10.5	17.0	37.5	14.5	79.5	52.0	28
21 Nov.	9.5	24.0	27.5	18.0	79.0	45.5	40
28 Nov.	9.5	11.5	10.0	33.0	64.0	43.0	77
5 Dec.	10.5	8.5	9.5	51.5	80.0	61.0	84
12 Dec.	10.0	19.0	9.0	42.0	80.0	51.0	82
19 Dec.	11.0	27.5	5.5	36.0	80.0	41.5	87
26 Dec.	6.5	5.5	1.5	23.0	36.5	24.5	94
2 Jan.	-	-	(Holiday)	-	-	-	-
Total	93.0	168.0	112.5	285.0	658.5	397.5	73
9 Jan. 53	9.2	18.1	26.9	25.8	80.0	52.7	49
16 Jan.	8.3	4.8	35.7	31.2	80.0	66.9	47
23 Jan.	10.1	22.5	21.8	25.6	80.0	47.4	54
30 Jan.	9.5	37.0	17.6	17.7	81.8	35.3	50
6 Feb.	8.5	36.3	6.8	28.5	80.1	35.3	81
13 Feb.	7.8	29.7	9.3	34.9	81.7	44.2	79
20 Feb.	7.8	3.9	6.1	62.2	80.0	68.3	91
27 Feb.	8.4	2.5	10.7	58.1	79.7	68.8	84
6 Mar.	8.1	0.0	18.6	65.6	92.3	84.2	78
13 Mar.	10.5	26.0	10.2	57.3	104.0	67.5	85
20 Mar.	12.5	6.1	0.1	85.1	103.8	85.2	99
27 Mar.	7.8	10.2	9.2	77.0	104.2	86.2	89
Total	108.5	197.1	173.0	569.0	1047.6	742.0	74
2 Apr. 53	7.4	6.4	4.6	69.4	87.8	74.0	94
12 Apr.	13.1	16.1	4.7	78.7	112.6	83.4	94
17 Apr.	9.1	8.2	21.5	65.1	103.9	86.6	75
24 Apr.	10.8	7.7	3.3	76.0	97.8	79.3	96
1 May	9.2	11.8	3.3	87.7	112.0	91.0	95
8 May	7.3	11.4	39.1	44.1	101.9	83.2	53
15 May	9.2	2.4	44.5	47.9	104.0	92.4	52
22 May	11.9	0.0	11.6	80.5	104.0	92.1	87
29 May	12.7	2.0	32.9	83.6	131.2	116.5	72
7 June	8.3	8.4	7.8	82.6	107.1	90.4	91
14 June	12.3	15.7	23.9	97.3	149.2	121.2	80
21 June	11.8	9.4	5.4	105.9	132.5	111.3	95
28 June	10.8	7.4	5.5	124.0	147.7	129.5	96
Total	133.9	106.9	208.1	1042.8	1491.7	1250.9	83
5 July 53	7.2	9.3	11.3	125.2	153.0	136.5	92
12 July	12.9	3.5	19.8	99.8	136.0	119.6	83
19 July	13.3	8.6	8.5	103.8	134.2	112.3	92
26 July	8.7	3.4	11.7	124.2	148.0	135.9	91
30 July	8.0	0.0	3.0	57.2	68.2	60.2	95
Total	50.1	24.8	54.3	510.2	639.4	564.5	91
GRAND TOTAL	385.5	496.8	547.9	2407.0	3837.2	2954.9	80

conditions which have interrupted computation or threaten to cause errors.

Operation: All "machine on" time not accounted for by the above categories. This would include input-output, code debugging, idle time, and running time. The idle time has been kept to a minimum by the pressure of mathematical work.

APPENDIX

Arithmetic Test Code:

The arithmetic test code is intended to test for correct execution of the major machine orders. It is essentially a series of sub-routines, one for each order. Within each sub-routine standard operands are used by the order concerned and the result checked against a stored correct answer. In almost every case, two sets of operands are used: one to present maximum digit loads to the gate drivers concerned and one to present minimum loads. From a gate loading standpoint all numbers which may be handled by the order in question lie between these extremes.

If a given subroutine elicits an error, a card (12 words) is punched out identifying the type of error and the order concerned as well as all quantities entering into the computation. Note that an operand can be correctly stored in the memory but transferred incorrectly into the arithmetic unit. Therefore the operands are first positioned in the arithmetic unit (e.g. the multiplier is transferred from the memory into R_2 in the multiplication cases) then are temporarily stored from this position (by the two orders R_2 --> R_1, store for R_2 quantities such as the multiplier). It is these temporarily stored images of the operands which are provided at the error punch out.

The sequence in which processes are tested involves some important considerations. The machine can detect its own errors only by using processes which may themselves be in error. The comparison of the computed value with the stored correct answer necessarily involves a subtraction; and the discrimination on the zero or non-zero error must be made by a

conditional transfer of control. Therefore the transfer control orders are checked first. If these are correct, it is then possible to check the summation orders. If both are correct, then the other orders can be tested.

The scope of this test is summarized in the following table:

Arithmetic Test Sequence

Test	Remarks
1. Total Sum of Memory = 0	Memory contents check.
2. Transfer control	Tests all cases of conditional and unconditional transfers of control.
3. Summation (Digit)	Considering any stage of the adder, tests the 8 combinations of the two input digits and the carry.
4. Summation (Sign)	Tests the complement gate selector in the 8 combinations of the sign of the incoming number and the magnitude and +/- digits of the order.
5. R_2 to R_1	Tests R_2 --> R^3 gating under minimum and maximum digit loads. (R^3 --> R_1 transfer has already been tested by 4).
6. 1 Shift Right 2 Shift Left 4 Shift Right 8 Shift Left 16 Shift Right 32 Shift Left	Tests individual digits of the shift counting channel.
7. 31 Shift Right	Maximum digit load on shift count.
8. Gate loads	Tests all RI and RII gates under minimum and maximum digit loads.
9. Multiplication	Tests 8 combinations of sign of multiplier, sign of multiplicand, and Roundoff-No Roundoff.

10.	Multiplication	Tests special case of small negative product which rounds to zero.
11.	Multiplication	Alternating Accept-Reject (i.e. multiplier alternates zeros and ones). Supplements 9.
12.	Division	Tests 4 combinations of sign of numerator and sign of denominator.

Read Around Test Code:

This code tests the read around, or spill, properties of each point in the memory by clearing the four neighbors (left, right, upper, lower) of the point to zero K times and then storing a one at the point n times, after which the four neighbors are examined to see if any have been changed to ones. If any errors are found, a card is punched out giving the frequency n , the bombarding word (initially all ones), the stage and neighbor failing, and the other parameters of the code.

The code is stored initially in the upper half of the memory and first tests all points in the lower half at the specified initial frequency. Then the code transfers itself to the lower half and tests the upper half at the same frequency. After this the code is re-transferred to the upper half, the frequency increased by a specified increment, and the process repeated until the given maximum test frequency has been used.

When a given stage (i.e. memory cathode ray tube) has failed h times it is removed from the test, as much as possible, by changing the corresponding digit in the bombarding word to a zero. Another parameter in the code determines by what integer the last bombarded address is

increased to obtain the next test point. If this is other than one, then the necessary several passes are made through the half memory before testing is transferred to the other half.

PART II - MATHEMATICS

I. TOTAL DIFFERENTIAL EQUATIONS

1.0 Introduction. Integration methods and accuracy.

In order to gain some feeling for the difficulties that might arise in the solution of systems of total differential equations a few diverse problems of this type were handled. In each case the problem chosen was considered to have some inherent interest from an applied mathematical, from a numerical analysis, and from a machine operation or maintenance point of view.

A large group of solutions of an astrophysical problem whose solution involved the integration of a system of three first order non-linear differential equations was computed. The so-called Heun method of integration was used in all of them. The group was determined by a number of parameters which varied from solution to solution. By smoothness criteria it was thus possible to check on possible errors in computation.

As a second step in this program a number of solutions of a slightly modified form of the Bessel equation was obtained by means of the well-known Runge-Kutta method. The primary interest here from the point of view of numerical analysis was two-fold: the equation has singularities in its coefficients and the solution function varies over a large range in a quite small interval so that in order to keep a fixed number of significant figures throughout the calculation a "floating point" system was adopted. The integration disclosed a few errors in published tables, which will be discussed below.

Some solutions of the wave equation were carried out by means of the Runge-Kutta method and were of a preliminary and exploratory nature.

This work did not therefore reveal much of interest from the point of view of this report and will not be discussed further.

A problem in electro-magnetic theory involving not only the integration of a second order differential equation but also the location in the complex plane of the proper values of this equation was done. In this problem the Milne method of integration was used as an experiment. We describe below in the appropriate place the methods used. It is of interest, however, to note that regions of the complex plane were examined for proper values by means of the well-known theorem in complex variable theory which enables one to count zeros and poles in a region by knowing certain information on the boundary of that region.

The last problem in this group was concerned with the design of a particle accelerator and involved the solution of a pair of second order total differential equations. The problem had several points of interest from our point of view and will be discussed in detail later. These points briefly are that the total path of integration consisted of a number of segments in each of which the differential equation had a different form and in each of which small random perturbations were imposed. It was desired to know under what conditions the system would be stable in the sense that small variations in the form of the equation would produce, after many revolutions, a small variation in the basic orbit.

The three integration methods.

We now describe, in an abstract setting, the three integration procedures to indicate the mathematical questions involved.

To this end we consider a system of first order differential equations

$$y_i' = f_i(x,y) = f_i(x; y_1, \ldots\ldots, y_n) \quad , \quad (i = 1, 2, \ldots\ldots, I)$$

subject to the initial conditions

$$y_i(0) = y_{i0}$$

where $y_i' = dy_i/dx$

The Heun method can now be described as follows: Let $(x,y) = (x, y_1, \ldots\ldots y_n)$ be known at $x = x_s$ and let it be (x_s, y_s). We describe how to form $y(x_s + dx)$ by the Heun procedure where $dx > 0$ is an arbitrary interval. Define

$$k_{i,1} = dx \cdot f_i(x_s, y_s) \quad , \quad k_{i,2} = dx \cdot f_i(x_s + dx, y_s + k_1)$$

$y_i(x_s + dx)$ by the relation

$$y_i(x_s + dx) = y_{is} + \tfrac{1}{2}(k_{i1} + k_{i2}) \quad , \quad i = 1, 2, \ldots, I$$

This method clearly involves the evaluation of the functions f_i twice in each interval dx. We can estimate its accuracy in two ways. First we give a heuristic discussion which helps reveal the nature of the method.

Let us suppose that the functions $f_i(x,y)$ do not involve the vector function y, i.e. the problem is one purely of effecting a quadrature. Then with the help of Taylor's expansion we have

$$y_i(x_s + dx) = y_{is} + \tfrac{1}{2} dx \cdot f_i(x_s + dx)$$
$$= y_{is} + dx \cdot y_{is}' + \tfrac{1}{2} dx^2 y_{is}'' + \cdots\cdots$$

since $y_i' = f_i(x)$. Thus the error, i.e. the difference between $y_i(x_s + dx)$ and the true value $\overline{y}_i(x_s + dx)$ is expressible as

$$|\overline{y}_i(x_s + dx) - y_i(x_s + dx)| \sim \tfrac{1}{12} dx^3 \cdot y_{is}'''$$

The error is therefore crudely proportional to the third differential of the solution function.

We now give a more precise evaluation of the error. We expect this will serve only to bring out the dependence of the error on the $\partial f_i / \partial y_j$. It is not difficult to see that

$$k_{i2} = dx \left\{ y'_{is} + dx\, y''_{is} + \tfrac{1}{2} dx^2 \left[y'''_{is} - y''_{js} f_{y_j} \right] + \cdots \right\}$$

where repeated indices indicate summation on that index. Thus

$$y_i (x_s + dx) = y_{is} + dx \cdot y'_{is} + \tfrac{1}{2} dx^2 y''_{is} + \tfrac{1}{4} dx^3 (y'''_{is} - y''_{js} f_{y_j}) + \cdots$$

and

$$\left| \bar{y}_i (x_s + dx) - y_i (x_s + dx) \right| \sim \tfrac{1}{12} dx^3 \left| (y'''_{is} - 3 y''_{js} f_{i y_j}) \right|$$

For the moment we suspend discussion of the Heun method and turn attention to that of Runge-Kutta. We now define

$$l_{i1} = dx \cdot f_i (x_s, y_s) \quad , \quad l_{i2} = dx \cdot f_i \left(x_s + \tfrac{dx}{2}, y_s + l_{i1}\right)$$

$$l_{i3} = dx \cdot f_i \left(x_s + \tfrac{dx}{2}, y_s + l_{i2}\right), \quad l_{i4} = dx \cdot f_i (x_s + dx, y_s + l_{i3})$$

and describe how to form $y(x_s + dx)$ by the Runge-Kutta procedure. Define

$$y_i (x_s + dx) = y_{is} + \tfrac{1}{6} \left(l_{i1} + 2 l_{i2} + 2 l_{i3} + l_{i4} \right), \quad i = 1, 2, \ldots I$$

This method involves twice as much computation as does that of Heun since the functions f_i must be evaluated four times in each interval dx . We now give error estimates paralleling those above and also estimate the relative size of the fundamental step for each method of comparable accuracy.

As before we first suppose the functions f_i are independent of y . Then we see that

$$y_i(x_s+\Delta x) = y_{is} + \frac{\Delta x}{6}\left\{ f_i(x_s) + 4f_i\left(x_s+\frac{\Delta x}{2}\right) + f_i(x_s+\Delta x)\right\}$$

$$= y_{is} + y'_{is}\cdot\Delta x + \frac{1}{2}\Delta x^2 y''_{is} + \frac{1}{6}\Delta x^3 y'''_{is} + \frac{1}{24}\Delta x^4 y^{iv}_{is} + \frac{5}{576}\Delta x^5 y^{v}_{is} + \cdots$$

Thus if, as before, $\overline{y_i}(x_s+\Delta x)$ is the true value of y at $x = x_s+\Delta x$, then

$$|\overline{y_i}(x_s+\Delta x) - y_i(x_s+\Delta x)| \sim \frac{1}{2880}\Delta x^5 |y^{v}_{is}|$$

Let us now compare the sizes of the fundamental steps one can take in each of these methods. I.e. for comparable accuracy what is the size of Δx_R, the step possible with the Runge-Kutta method, compared to Δx_H, that one for the Heun method? Suppose that 2^{-15r} is the precision desired, i.e.

$$\frac{1}{12}\Delta x_H^3 |y'''| \leq 2^{-15r}, \quad \frac{1}{2880}\Delta x_R^5 |y^{v}| \leq 2^{-15r}$$

where $15r$ measures the number of binary places to be kept.

We recall Markoff's theorem which states that if $P(x)$ is a polynomial of degree n and is such that $|P(x)| \leq 1$ on the interval $(-1,+1)$, then $|P'(x)| \leq n^2$ on this interval.

Let us therefore assume $y(x)$ satisfies the hypotheses of Markoff's result. Then we may replace our inequalities by

$$\frac{1}{12} n^6 \Delta x_H^3 \sim 2^{-15r}, \quad \frac{1}{2880} n^{10} \Delta x_R^5 \sim 2^{-15r}$$

Thus

$$\Delta x_H \sim 2^{-5r} n^{-2} 12^{1/3}, \quad \Delta x_R \sim 2^{-3r} n^{-2} \cdot 2880^{1/5}$$

and

$$\Delta x_R = \left(\frac{2880^{1/5}}{12^{1/3}}\right)\cdot 2^{2r}\Delta x_H = 2.15\cdot 2^{2r}\Delta x_H$$

In passing we note that this estimate is independent of n, the order of the polynomial. Now $r=1$ corresponds to $10^{-4.5}$ and $r=2$ to 10^{-9}.

For these values of τ we have

$$dx_R \sim 8.6\, dx_H \,, \quad dx_R \sim 37.7\, dx_H$$

Thus we see that the Runge-Kutta scheme is considerably more efficient than is the Heun method, which we recall involves only one-half as much calculation per step, and the estimates above give a measure of this efficiency.

We return now to the Runge-Kutta method to give a more precise error estimate. It can be seen after a good deal of calculation that the error estimate, for the case $I = 1$, is

$$\frac{1}{2880} dx^5 \left| y^v + 5 f_y y^{iv} + 10 (\mathcal{D} f_y - f_y) y''' + 10 (2\mathcal{D} f_y - f_y + 3 f_y^3) y'' \right|,$$

where

$$\mathcal{D} f_y = f_{xy} + y' f_{yy} .$$

We discuss next some cases where these classical estimates must be either viewed carefully or even abandoned. In obtaining these estimates we assumed tacitly that the contributions of the terms omitted from our series expansions were negligible. This is not always the case.

Suppose for example that $y(x) = x^a$. Then

$$y^{(n)}(x) = \frac{\Gamma(a+1)}{\Gamma(a-n+1)} x^{a-n}$$

That

$$dx^n y^{(n)}(x) = \frac{\Gamma(a+1)}{\Gamma(a-n+1)} \left(\frac{x}{dx}\right)^{a-n} (dx)^n$$

and we may therefore expect that for x near dx the estimates involving dx^3 , dx^5 are to be replaced by ones in dx^a. In fact, we see that all terms in the Taylor expansion of y about $x = dx$ are of the same order, namely dx^a . We shall examine this situation in a little more detail below.

Suppose then that our differential equation is

$$y' = a x^{a-1}$$

Then the Heun method -- in this case the trapezoid rule -- gives

$$y_H(2dx) = y(dx) + \frac{a}{2} y(dx)(1+2^{a-1}) = y(dx)\left(1+\frac{a}{2}(1+2^{a-1})\right) =$$

$$= (2\,dx)^a \left(\frac{1+a(1+2^{a-1})/2}{2^a} \right),$$

and the Runge-Kutta method -- in this case the Simpson rule -- gives

$$y_R(2dx) = y(dx) + \frac{a}{36} y(dx)\left(6+16(3/2)^a + 3\cdot 2^a\right) =$$

$$= y(dx)\left(1+\tfrac{1}{36}\left(6+16\cdot(3/2)^a+3\cdot 2^a\right)\right) =$$

$$= (2\,dx)^a \left(\frac{1+a(6+16(3/2)^a+3\cdot 2^a)/36}{2^a} \right)$$

Thus the absolute error for the former method is

$$\varepsilon_H = 2^a \left(\frac{1+a(1+2^{a-1})/2}{2^a} - 1 \right)(dx)^a$$

and for the latter is

$$\varepsilon_R = 2^a \left(\frac{1+a(6+16(3/2)^a+3\cdot 2^a)/36}{2^a} - 1 \right)(dx)^a .$$

These confirm our previous remark that the error estimates are no longer proportional to dx^3 or dx^5 but to dx^a ; this is particularly severe when $a < 1$. In passing we note, as expected, that

$\varepsilon_H = 0$ for $a = 0, 1, 2$ and $\varepsilon_R = 0$ for $a = 0, 1, 2, 3, 4$.

It is possibly of interest to see what happens when

$$y' = 1/x$$

Then the Heun method gives

$$y_H(2dx) = y(dx) + .14 = \ln(2dx) + (3/4 - \ln 2)$$

and the Runge-Kutta method

$$y_R(2dx) = y(dx) + 2.7\,6 = \ln(2dx) + (25/36 - \ln 2)$$

Thus the true estimates ε_H , ε_R become

$$\varepsilon_H = 3/4 - \ln 2 , \qquad \varepsilon_R = 25/36 - \ln 2 .$$

These are independent of dx , the interval length, so that no refining of the integration interval can improve the absolute error.

The classical estimates E_H , E_R give in this case near $x = dx$

$$E_H \sim \frac{1}{3}\frac{dx^3}{x^3} \sim \frac{1}{3} , \qquad E_R \sim \frac{1}{120}\frac{dx^5}{x^5} \sim \frac{1}{120} .$$

Let us next turn to a consideration of what can happen when f contains y . To this end we consider the differential equation

$$y' = \frac{a}{x}\, y$$

Here f_y , the quantity entering both the Heun and Runge-Kutta classical estimates, is proportional to $1/x$.

Since the solution we seek, $y = x^a$, is of the same form as that discussed above we may again expect our estimates to depend not on dx^3 or dx^5 but on dx^a . Our main purpose is to investigate what effect, if any, the f_y has on the error. In this case the calssical estimates E_H , E_R for the two methods at hand are

$$E_H \sim \frac{1}{12} dx^3 \left| y''' - 3 y'' f_y \right| = \frac{1}{6} \left| a(a^2-1) \right| dx^a ,$$

$$E_R \sim \frac{1}{720} \left| a(a-1)(9a^3 - 11a^2 + 19a - 6) \right| dx^a$$

We see that

$E_H = 0$ for $a = 0, 1$ and $a = -1$;

$E_R = 0$ for $a = 0, 1$ and for an $\bar{a}$ such that $0 < \bar{a} < 1$.

We go now to contrast these estimates E_H , E_R with their exact counterparts ε_H , ε_R . By a simple calculation we find that

$$y_H(2dx) = y(dx)(1 + 3/4\, a + 1/4\, a^2) = \frac{dx^a}{4}(a^2+3a+4) = y(2dx)\left(\frac{a^2+3a+4}{4\cdot 2^a}\right)$$

and that

$$y_R(2dx) = \frac{y(dx)}{108}\left(a^4+6a^3+26a^2+75a+108\right) =$$

$$= y(2dx)\left(\frac{a^4+6a^3+26a^2+75a+108}{108\cdot 2^a}\right).$$

Thus

$$\varepsilon_H = (2dx)^a\left(\frac{a^2+3a+4}{4\cdot 2^a} - 1\right) \quad , \quad \varepsilon_R = (2dx)^a\left(\frac{a^4+6a^3+26a^2+75a+108}{108\cdot 2^a} - 1\right).$$

We see that

$$\varepsilon_H = 0 \quad \text{for} \quad a = 0, 1, -1 \qquad \text{and} \qquad \varepsilon_R = 0 \quad \text{for} \quad a = 0, 1, -1 .$$

In passing we note that on the interval (0,1) the expression $\left|(a^2+3a+4)/4\cdot 2^a - 1\right|$ has its maximum near $a = .6$ where its value is $\lesssim .017$. Thus the true relative error in the Heun method near $x = 0$ is

$$\frac{\varepsilon_H}{y(2dx)} \lesssim .017 ;$$

the comparable expression in ε_R has its maximum near $a = .7$ where its value is $\lesssim .00052$. Thus the true relative error in the Runge-Kutta method near $x = 0$ is

$$\frac{\varepsilon_R}{y(2dx)} \leqq .00052 .$$

From what has been said we observe first that the classical estimates must be properly interpreted to be meaningful in certain, quite reasonable cases, and that the presence of a singularity in

may lower the expected precision of either method.

We turn attention now to various estimates of comparative sizes of integration intervals. We require that the true absolute error should be $\nu 2^{-15\alpha}$ ($\alpha = 1$ or 2 are about the precision levels most commonly of interest). Then for $a = 1/2$ we have

$$dx_H \sim 2.10^{3-9\alpha} \quad , \quad dx_A \sim 2.10^{-6-9\alpha}$$

showing an advantage of three powers of 10 for the Runge-Kutta over the Heun method in this case.

We now turn to the third method under consideration, that of Milne. Per time step this method only involves about half the work of that required for the Runge-Kutta.

The Milne method presupposes as known the following data:

$$y'_{i,s-3} \; , \; y'_{i,s-2} \; , \; y'_{i,s-1} \; , \; o_{i,s-4} \; , \; y_{i,s-2} \; .$$

It then proceeds to determine a first estimate ${}'y_{is}$ for $y_i(x_s + dx)$ by means of the formula

$${}'y_{is} = y_{i,s-4} + \frac{4\,dx}{3}\left(2\,y'_{i,s-3} - y'_{i,s-2} + 2\,y'_{i,s-1}\right).$$

With this an estimate for y'_{is} is found by means of

$${}'y'_{is} = f(x_s, {}'y_s)$$

and then a revised value $y_{is} = {}^2y_{is}$ is calculated as

$${}^2y_{is} = y_{i,s-2} + \frac{dx}{3}\left({}'y'_{i,s} + 4\,y'_{i,s-1} + y'_{i,s-2}\right) \; ;$$

finally a revised ${}^2y_{is}$ is obtained.

We see at once that if the functions f_i are independent of y_j then the value ${}'y_{is}$ is formed by Simpson's rule and the error will be

32 times the corresponding one for the Runge-Kutta method. I.e., if dx_M is the length of a step in the Milne method, then for comparable accuracy

$$dx_R = 4\, dx_M$$

The amount of computation involved in this method is therefore comparable to that for the Runge-Kutta for that method since no revision of $'y_s$ is needed.

We go now to give a more precise estimate of the error when f involves y. We do this in the case $I = 1$ by parity with the comparable estimate above for the Runge-Kutta one. We assume the data y'_{s-3}, y'_{s-2}, y'_{s-1}, y_{s-4}, y_{s-2} are exact. Then the error in $'y_s$ is this:

$$|y_s - {}'y_s| \sim \frac{14}{45} |y^v_{s-2}|\, dx^5 \;;$$

the error in $'y'_s$ is

$$|y'_s - {}'y'_s| \nu |f_y(x_s, y_s)| \cdot |y^v_{s-2}|\, dx^5 \;;$$

and finally the error in ${}^2y_s$ is

$$|y_s - {}^2y_s| \nu \tfrac{1}{90} dx^5 |y^v_{s-1}| + \tfrac{14}{135} |f_y(x_s, y_s)| \cdot |y^v_{s-1}|\, dx^6 =$$

$$= \frac{dx^5}{45} |y^v_{s-1}| \left(\tfrac{1}{2} + \tfrac{14}{3} dx |f_y(x_s, y_s)| \right).$$

We would therefore expect this method to be more precise than the Runge-Kutta since the dependence of the error on f_y is less stringent, in that in this estimate f_y is multiplied by dx.

Let us apply the Milne method to the equation

$$y' = \frac{a}{x}\, y$$

and compare the result with those obtained earlier for our other two methods.

We assume the values at $s = 0, 1, 2, 3$ to be given and find

$$y_4 = (dx)^a \left\{ 2^a + \frac{a}{3}\left[\frac{a}{18}(12 - 3\cdot 2^a + 4\cdot 3^a) + 4\cdot 3^{a-1} + 2^{a-1} \right] \right\}$$

We compare this to $y_4 = 4^a (dx)^a$, i.e. we examine the difference

$$\varphi(a) = 4^a - \left\{ 2^a + \frac{a}{3}\left[\frac{a}{18}(12 - 3\cdot 2^a + 4\cdot 3^a) + 4\cdot 3^{a-1} + 2^{a-1} \right] \right\}$$

We see that φ vanishes for $a = 0, 1, 2, 3, 4$ -- recall that the comparable expression for the Runge-Kutta case vanished for 0, 1, -1. Further the maximum of $\varphi(a)/4^a$ on the inverval (0, 1) is about .0096. Thus

$$\left| \frac{y_M(4\,dx) - y(4\,dx)}{y(4\,dx)} \right| \lesssim .0096 .$$

Comparing this to our comparable estimate for the Runge-Kutta scheme we see that the bound of the relative error of the Milne method is about 20 times as large as that of the Runge-Kutta method. We go now to compare in a crude fashion the length of the fundamental step for each of these methods. Near $a = 0$ it is easy to see that

$$dx_R \sim dx_M \left| \frac{\ln 2 - 75/108}{\ln 2 - 11/18} \right|^{-1/a} \sim (63)^{1/a}\, dx_M .$$

(This estimate is valid for $a < .08$.) Next for $a = 1/2$ we have

$$dx_R \sim (21)^2\, dx_M$$

Moreover, if we wish the error to be the order of 2^{-15}, then for $a = 1/2$

$$dx_R \sim 2\times 10^{-3}, \quad dx_M \sim 5\times 10^{-4}, \quad dx_R \approx 4\,dx_M = 2(2\,dx_M)$$

Thus for $0 < a < 1$ the Runge-Kutta method would seem to be superior to that of Milne. However, the situation for a in the range from 1 to 5 is as indicated below where we give a table showing the

relative errors δ_R , δ_M for the two methods as a function of a .

a	δ_R	δ_M	$(\delta_M/\delta_R)^{1/a}$
1.0	0	0	—
1.5	.0038	.0015	.116
2.0	.0139	0	0
2.5	.0329	.0008	.226
3.0	.0625	0	0
3.5	.1028	.0009	.498
4.0	.1528	0	0
4.5	.2107	.0045	.709
5.0	.2743	.0139	.552

The ratio $(\delta_M / \delta_R)^{1/a}$ is a measure of the comparative interval lengths for the two methods. In fact

$$dx_R \sim (\delta_M / \delta_R)^{1/4} (2\,dx_M)$$

Thus, in the range $1 < a \leq 5$ the Milne method seems superior to that of Runge-Kutta since our figure of merit $(\delta_M / \delta_R)^{1/a}$ is inversely proportional to the amount of labor needed in the latter scheme as against the former.

1.1 Stellar interiors.

Professor M. Schwarzschild of the Department of Astronomy, Princeton University has for some time interested himself in the internal structure of stars, particularly red-giants. To this end several sets of computations were performed to integrate the equilibrium equations describing the stellar interior.

These equations may be expressed in at least one important case in the form

$$\frac{dp}{dx} = -l\,\frac{pq}{t\,x^2}\,\beta\ ,\quad \frac{dq}{dx} = l\,\frac{px^2}{t}\,\beta\ ,\quad \frac{dt}{dx} = -lj\,c\,\frac{p^2}{t^{6.5}\,x^2}\cdot\alpha\beta^2,$$

where p is pressure, t temperature, q mass and x radial distance; β measures the ratio of radiation pressure to total pressure and α the ratio of electron to ionization opacity.

With the help of the transformations

$$\lambda = \log p/p_0\ ,\quad \tau = \log t/t_0\ ,\quad \psi = \log q/q_0\ ,\quad y = \log x/x_0$$

equations (1) become

$$\log\left(-\frac{d\lambda}{dy}\right) = -\tau - (1-\gamma_1)\psi - y + \log\beta\ ,$$

$$\log\left(-\frac{d\tau}{dy}\right) = n_1\tau + n_2\lambda + 1.3\,\gamma_2\,\psi - y + 2\log\beta + \log\alpha + \log\left(l_0 r_0 c\,\frac{p_0}{t_0^{4.5}}\,x_0\right),$$

$$\log\left(-\frac{dx}{dy}\right) = -\tau + \lambda + (1+\gamma_3)\,x + 3y + \log\beta,$$

with

$$\log(1-\beta) = 4\tau - \lambda - \log(B\,t_0^4/p_0)$$

$$\log(\alpha-1) = \tfrac{1}{2}\tau - \gamma\psi + \log(1-\beta) - \log\beta + \log(A\,t_0^{1/2}/l_0)$$

$$\beta = 1 - B\,t^4/p\ ,\qquad \alpha = 1 + A\cdot\frac{1-\beta}{\beta}\cdot t^{1/2}\ ,$$

$$l = l_0\,(q/q_0)^{-\gamma}\ ,\qquad j = j_0\,(q/q_0)^{-\delta}\ .$$

and with certain relations between p_0, l_0, q_0, t_0, x_0. (From equations (1) we have $n_1 = 9.5$, $n_2 = 2$; however, in some problems n_1, n_2 were assigned different values. Again $\gamma_1 = \gamma_2 = \gamma_3 = \gamma$; but in some problems they were assigned other values.

For purposes of keeping all quantities in the interval (-1, +1) λ, τ, ψ, y, β, α were replaced by

$$\lambda^* = \tfrac{1}{10}\lambda \;,\quad \tau = \tfrac{1}{10}\tau \;,\quad \Psi^* = \tfrac{1}{10}\Psi \;,\quad y^* = \tfrac{1}{10}y \;,\quad \beta^* = \tfrac{1}{10}\beta \;,\quad \alpha^* = \tfrac{1}{100}\alpha$$

It is convenient to let

$$a = -k\left(\tfrac{1}{16}\tau^* + \tfrac{1-\nu_1}{16}\Psi^* + \tfrac{1}{16}y^*\right) \;,\qquad b = -k\left(\tfrac{\nu_1}{16}\tau^* - \tfrac{\nu_2}{16}\lambda^* - \tfrac{1+3\nu_2}{16}\Psi^* + \tfrac{1}{16}y^*\right)$$

$$c = -k\left(\tfrac{1}{16}\tau^* - \tfrac{1}{16}\lambda^* - \tfrac{1+\nu_3}{16} - \tfrac{3}{16}y^*\right) \;,\qquad d = -k\left(-\tfrac{7}{16}\tau^* + \tfrac{1}{16}\lambda^*\right),$$

$$e = -k\left(-\tfrac{1}{32}\tau^* + \tfrac{\nu_3}{16}\Psi^*\right);$$

$$r = -\tfrac{1}{800}\cdot e^{2q a} \;,\qquad s = -\tfrac{1}{800}\cdot\tfrac{125}{128}\cdot\left(l_0 j_0 C\,\frac{p_0^2}{t_0^{4.5}\,x_0}\right)e^{2q b} \;,\qquad t = -\tfrac{1}{800}\,e^{2q c}$$

$$u = \tfrac{1}{10}\left(B\,\frac{t_0^{\,4}}{p_0}\right)e^{2q d} \;,\qquad v = \tfrac{1}{800}\left(A\,\frac{t_0^{1/2}}{l_0}\right)e^{2q e}$$

In these terms the equations handled by the machine were

$$\frac{1}{10^3}\frac{d\lambda_i^*}{dy^*} = r_{i-1}\cdot\tfrac{1}{2}\,e^{2q(a_i - a_{i-1})}\cdot 8\beta_i^*$$

$$\frac{1}{10^3}\frac{d\tau_i^*}{dy^*} = s_{i-1}\cdot\tfrac{1}{2}\,e^{2q(b_i - b_{i-1})}\,(8\beta_i^*)^2\cdot\alpha_i^*\cdot 2^7$$

$$\frac{1}{10^3}\frac{dx_i^*}{dy^*} = t_{i-1}\cdot\tfrac{1}{2}\,e^{2q(c_i - c_{i-1})}\cdot 2\cdot 8\beta_i^*$$

$$\frac{1}{10} - \beta_i^* = u_{i-1}\cdot\tfrac{1}{2}\cdot e^{2q(d_i - d_{i-1})}\cdot 2$$

$$\alpha^* - \tfrac{1}{100} = v_{i-1}\cdot\tfrac{1}{2}\,e^{2q(e_i - e_{i-1})}\cdot 2\cdot\frac{\left(\frac{1}{10} - \beta_i^*\right)2^{-1}}{8\beta_i^*}\cdot 2^7$$

The exponentials we handled in each case as series expansions through terms of order dx^7. The integration method was that of Heun. In all about 90 trajectories were calculated involving about 35,000 steps of the Heun method. Thus in all about 70,000 evaluations fo the right-hand sides of the equations (4) was involved. Each such evaluation,

however, involves the order of 30 multiplications and so the group of integrations involved the order of 2×10^6 multiplications, irrespective of those performed in the conversion of the results into decimal form.

The first group of about 40 integrations were concerned with those intermediate zones of stars where electron scattering provides the main opacity. These trajectories are continuations of solutions for the outer zones were photo-ionization is the principal factor in opacity. These solutions for the outer and intermediate regions are then filled to a group of contracting cores. This gives a number of complete stellar models characterized by a large range of physical conditions for the study of the evolution of bright stars.

Some graphs showing the nature of the solutions are included. In these graphs the parameter

$$V = lq / tx$$

is plotted as a function of $\bar{u} = lpx^3 / tq$.

1.2 Bessel and Cylinder Functions.

At the request of Professor S. Chandrasekhar of Yerkes Observatory, extensive preliminary tabulations were made, at convenient intervals of y, of the functions

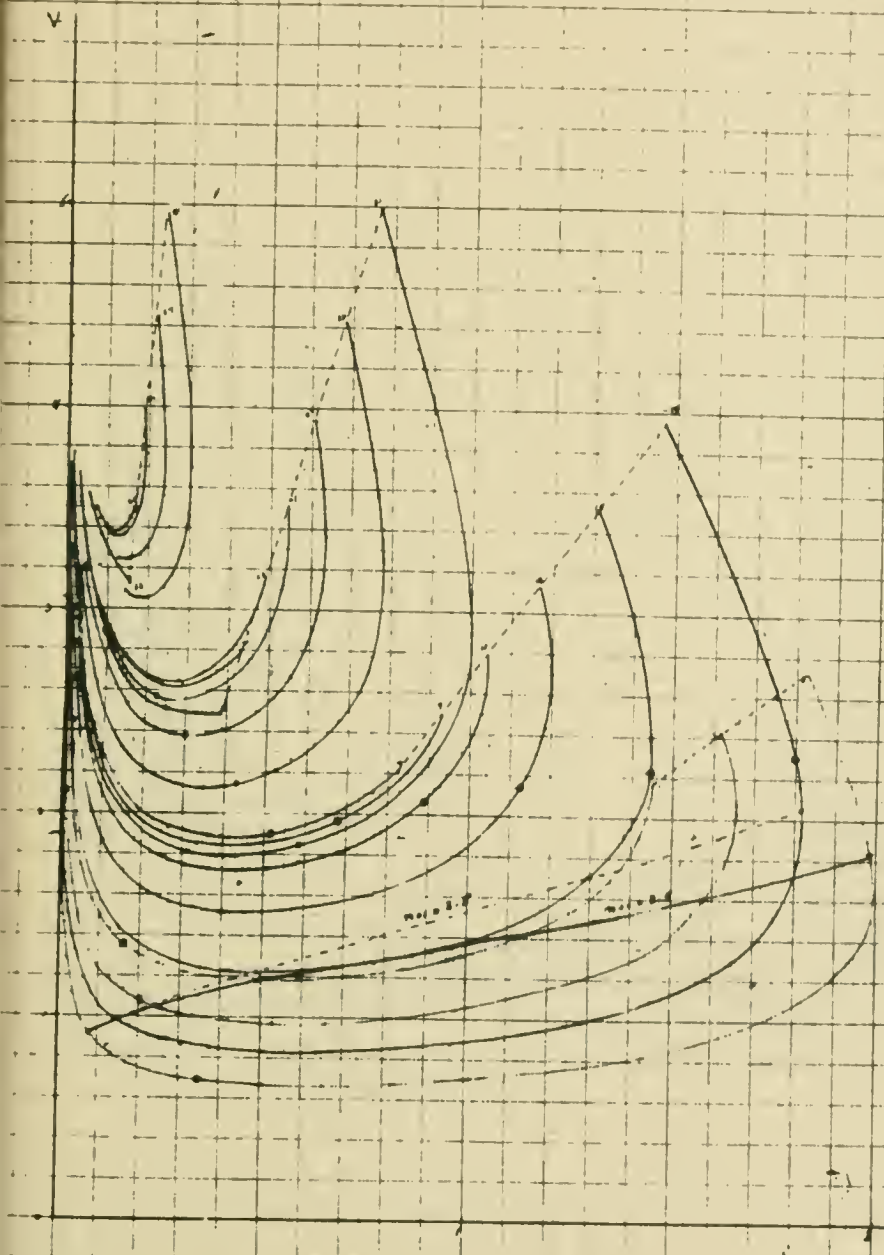
$$J_n(\alpha_{nj}\, y) \equiv J_n(x)$$

$$C_{njk}(\alpha_{njk}\, y) \equiv C_{njk}(x) \equiv Y_n(\alpha_{njk}\, \eta)\, J_n(x) - J_n(\alpha_{nji}\, \eta)\, Y_n(x$$

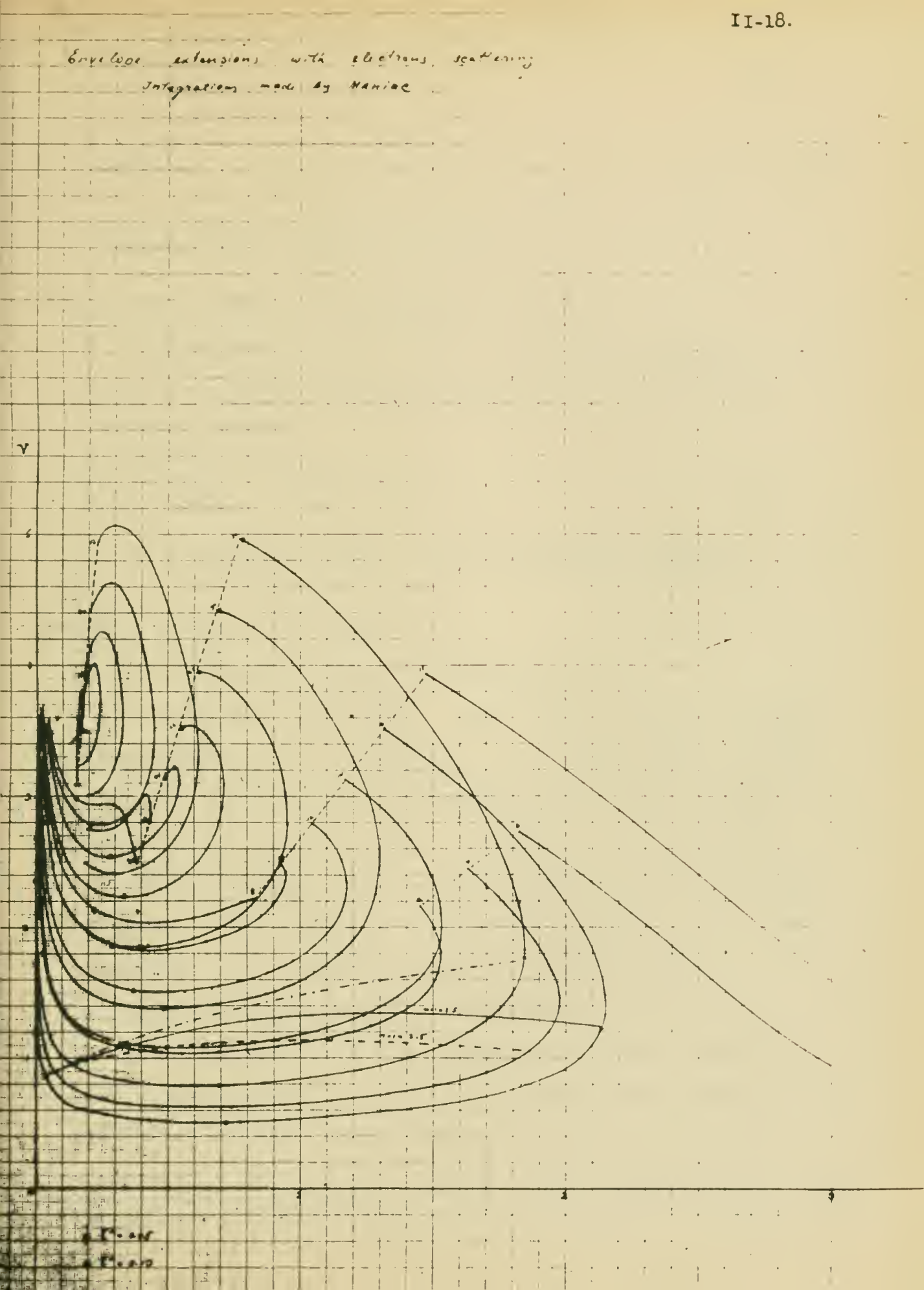
where $J_n(x)$ and $Y_n(x)$ are the Bessel functions of the first and second kinds of integral order $n > 0$, and α_{nj}, α_{nj} -- briefly, α -- are the j-th zeros of

Envelope extensions with electrons scattered

Integration made by MANIAC



Envelope extensions with electrons scattering
Integrations made by MANIAC



$J_n(\alpha)$

$Y_n(\eta\alpha)\,J_n(\alpha) - J_n(\eta\alpha)\,Y_n(\alpha)$

respectively, where η is a parameter, $0 < \eta < 1$

The variables J_n , Y_n , C_n , are cylinder functions of x , of order n , that is, they are solutions of the n-th order Bessel differential equation,

$$\frac{d^2u}{dx^2} + \frac{1}{x}\frac{du}{dx} + \left(1 - \frac{n^2}{x^2}\right)u = 0$$

in x , or letting $x = \alpha y$, of the differential equation

$$\frac{d^2u}{dy^2} + \frac{1}{y}\frac{du}{dy} + \left(\alpha^2 - \frac{n^2}{y^2}\right)u = 0$$

in y (this equation involving α as well as n).

Specifically, $J_n(\alpha_{nj}\, y)$ and $C_{nj}(\alpha_{nj}, y)$ (or their constant multiples) are, for integral $n > 0$, those cylinder functions of order n which vanish at the initial point $y = 0$ for J, $y = \eta$ for C , at the final point $y = 1$, and at $j - 1$ points between.

The preliminary computations were made just before the Institute for Advanced Study computer was dismantled for relocation. More refined computations will be made when the computer is recommissioned.

The function $J_n(\alpha_{nj}\, y)$ was tabulated at intervals $\Delta y = .01$ for $.0 \leqq y \leqq 1.04$, for every combination of parameter values $n = 1, 2, 3, 4, 5, 6$ with $j = 1, 2, 3, 4, 5$. (These tables differ from the usual tables of Bessel functions in being tabulated at equal decimal arguments of y rather than of $x = \alpha y$.) The function $C_{nj\eta}(\alpha_{nj}, y)$ was tabulated at intervals $\Delta y = .01$ for $\eta \leqq y \leqq 1.04$, for most

combinations of parameter-values n = 1, 2, 3, 4, 5, 6 with η = .2, .3, .4, .5, .6, .8 for j = 1 (and in some cases for other j). The results were given to eight figures, but are not necessarily that exact.

The roots α , used as input data, were available to us partly from Watson's "Bessel Functions", and partly in manuscript. They were given variously to about seven decimals (providing a natural limit of $\gtrsim 10^{-8}$ on the accuracy to be expected in the results).

METHOD OF COMPUTATION. The method choosen to compute these functions was to integrate the Bessel differential equation (5) by the Runge-Kutta process for second order differential equations.

The major points of interest in planning the integration are, first, the singularity of the differential equation at $y = o = x$, the lower limit of the desired range of y for $J(\alpha y)$, and second, the consequence that the solutions of (5) which are bounded near zero vanish there to order n , in fact

$$J_n(x) \upsilon \left(\frac{x}{2}\right)^n / n! \qquad \text{near} \quad x = o$$

We must consider the accumulation and propagation of truncation and rounding errors in the light of these peculiarities.

The main emphasis will be on the function J ; the requirements on C turn out to be less stringent.

RELATIVE ACCURACY. First we note that when a value $u(x_1$ is to be found by integrating a homogeneous linear differential equation $L[u] = 0$ from x_0 to x_1 , to attain a given accuracy in $u(x_1$ it is, in general, necessary to maintain <u>at least</u> the same <u>relative</u>

accuracy in · throughout the range ' ' - ' '

For suppose that errors are introduced into the values ,

$u'(\bar{x})$ at a certain , and that thereafter the integration is exact; then for a solution of the differential equation is produced instead of . If the errors amount to a scale error, so that

$$\tilde{u}(x) = (1+\epsilon)\, u(x)$$

then the relative error remains the same, ϵ, for $x > \bar{x}$ as at . But generally the solution is contaminated by other solutions, so we may write (not uniquely)

$$\tilde{u}(x) = (1+\epsilon_1)\, u(x) + \epsilon_2 \frac{u(\bar{x})}{v(\bar{x})} v(x)$$

where $v(x)$ is an independent solution, and ϵ_1, ϵ_2 are small.

The relative error at $x = \bar{x}$ is

$$\epsilon(\bar{x}) = \epsilon_1 + \epsilon_2 = \epsilon$$

Since

$$\tilde{u}'(x) = (1+\epsilon_1+\epsilon_2 \frac{v(\bar{x})}{v(\bar{x})}) \frac{u'(x)}{u(\bar{x})}$$

the relative error at x is

$$\epsilon(x) = \epsilon_1 + \epsilon_2 \frac{\left(\frac{v(x)}{v(\bar{x})}\right)}{\left(\frac{u(x)}{u(\bar{x})}\right)} = \epsilon_1 + \epsilon_2 \frac{\left(\frac{v(x)}{u(x)}\right)}{\left(\frac{v(\bar{x})}{u(\bar{x})}\right)}$$

Clearly if $\frac{v(x)}{u(x)}$ is of the same order as $\frac{v(\bar{x})}{u(\bar{x})}$, or smaller, then

$$\frac{\epsilon(x)}{\epsilon} \lesssim 1$$

In the contrary unfavorable case that $v(x)$ grows relatively to $u(x)$,

$$\frac{\epsilon(x)}{\epsilon} \sim$$

Thus an introduced relative error will generally not decrease, and may be amplified. (This amplification could be serious, for example, if $J_n(x)$ were computed by integration toward $x=0$ where the contaminating solution $Y_n(x)$ has a singularity; it is safer to integrate away from 0.)

In fact, of course, truncation and rounding errors arise at each integration step, and the total error, which is to be kept small, arises from their combined effects. Hence the relative errors at each step should be kept much smaller than the desired final relative accuracy.

FLOATING DECIMAL POINT. The Institute for Advanced Study computer is a 39-binary digit fixed binary-point machine, with fixed and relative accuracy of 2^{-39}, or about $10^{-11.7}$.

Since $J_n(x)$ is small for x near 0 (for example, $J_{10}(2) \doteq 2\ 10^{-10}$ it is clear that maintenance of any appreciable relative accuracy requires either multiple-precision or floating-point arithmetic. A floating-decimal-point operation was adopted, each "floated" number Z being represented as $Z = a \cdot 10^{p}$, where $|a| < 1$ and p is an integer, and a, x are represented in storage by the machine numbers $a, x \cdot 10^{-p}$

A complete floating point operation is uneconomical because of the considerable machine time needed for the logical manipulations of floating, relative to that used for useful computation -- especially in the case of additions, which are numerous in the Runge-Kutta process. This situation was mitigated by "common scaling" whereby variables

z_i to be added are scaled with a common exponent p, and by "critical scaling", whereby exponents p are adjusted only occasionally; the bounds for the a_i and the sequence of arithmetic must be designed so that neither overflow beyond the machine capacity, nor undue loss of significant figures, occurs between adjustments.

In this problem the quantities were generally scaled so that $1 > |a| \geq 1/10$ before multiplications, or $1/10 > \max|a_i| \geq 1/100$ before additions, so that the relative rounding error is $< 10^{-9}$. With this scheme (and proper starting values and integration steps) it was necessary to test and adjust the scaling only once during each integration step.

For the functions C, the floating point operation is not essential.

THE RUNGE-KUTTA METHOD. For a second-order differential equation $u'' = f(u, u', x)$ this method consists (cf. Collatz, Numerische Behandlung von Differentialylichungen, p. 33) in advancing from x to $x + dx$ (knowing $u = u(x)$ and $du = u'(x)dx$ by the following algorithm:

Let

$$k_1 = \tfrac{1}{2} dx^2 \, f(u, \tfrac{1}{dx}(du), x)$$

$$k_2 = \tfrac{1}{2} dx^2 \, f(u + \tfrac{1}{2} du + \tfrac{1}{4} k_1, \tfrac{1}{dx}(du + k_1), x + \tfrac{1}{2} dx)$$

$$k_3 = \tfrac{1}{2} dx^2 \, f(u + \tfrac{1}{2} du + \tfrac{1}{4} k_1, \tfrac{1}{dx}(du + k_2), x + \tfrac{1}{2} dx)$$

$$k_4 = \tfrac{1}{2} dx^2 \, f(u + du + k_3, \tfrac{1}{dx}(du + k_3), x + dx)$$

$$k = \tfrac{1}{3}(k_1 + k_2 + \ldots)$$

$$k' = \tfrac{1}{3}(k_1 + 2k_2 + 2k_3 + 2y)$$

Then

$$u(x+dx) \simeq u + du + k$$
$$du(x+dx) \simeq du + k'$$

Each integration step requires four substitutions into the differential equation.

TRUNCATION ERROR. The classical estimate for the error of the above method is

$$O\left[(u^{v} + \varphi)\, dx^{5}\right]$$

where φ is a polynomial in lower derivatives of u and in partial derivatives of f. (The corresponding error estimate for a first-order equation is $\frac{1}{2880}(u^{v} + \psi)\, dx^{5} = O\left[(u^{v} + \psi)\, dx^{5}\right]$.) Hence except for fortuitous cancellation, the absolute error is at least

$$O(u^{v}\, dx^{5})$$

so that the relative error, which must be made small, is at least

$$O\left(\frac{u^{v}\, dx^{5}}{u}\right)$$

The error is more critical in the case of J than in that of C. We know that $J_n(x)$ has the power series expansion

$$J_n(x) = \sum_{k=0}^{\infty} \frac{x^{n+2k}}{2^{n+2k}\, k!\, (n+k)!}$$

so that

$$J_n^{v}(x) = \begin{cases} O(x) & \text{for } n = (0), 2, 4, 6 \\ O(1) & \text{for } n = 1, 3, 5 \\ O(x^{n-5}) & \text{for } n \geq 5 \end{cases}$$

and the relative error is at least

$$\mathcal{O}\begin{pmatrix} J_n^{\vee}(x)\,dx^5 \\ J_n(x) \end{pmatrix} = \begin{cases} \mathcal{O}(x^{1-n}\,dx^5) & \text{for } n = (0), 2, 4, 6 \\ \mathcal{O}(x^{-n}\,dx^5) & \text{for } n = 1, 3, 5 \\ \mathcal{O}(x^{-5}\,dx^5) & \text{for } n \geq 5 \end{cases}$$

$$\begin{cases} \mathcal{O}(x\,dx^5) & \text{for } n = 0 \\ \mathcal{O}(x^{-1}\,dx^5) = \mathcal{O}\left[\left(\frac{dx}{x}\right)^1 dx^4\right] & \text{for } n = 1, 2 \\ \mathcal{O}(x^{-3}\,dx^5) = \mathcal{O}\left[\left(\frac{dx}{x}\right)^3 dx^2\right] & \text{for } n = 3, 4 \\ \mathcal{O}(x^{-5}\,dx^5) = \mathcal{O}\left[\left(\frac{dx}{x}\right)^5\right] & \text{for } n \geq 5 \end{cases}$$

It is seen, especially in the cases $n \geq 5$, that $\frac{dx}{x}$ must be fairly small in order ther the relative truncation error be negligible. This means, first, that the integration cannot be started at $x = 0 = y$ (even though $J_n(x)$ is not singular there) but must be started away from 0; and second, that the integration step dx must be small compared to this starting value:

$$0 < dx << x^{\circ}$$

To minimize the total number of integration steps, the starting value should be as large as possible, namely $x^{\circ} = \alpha y^{\circ}$ where y° is the first tabular argument after 0 (e.g. $y^{\circ} = .01$).

Since the desired range of x for $C(x)$ avoids the singularity of the Bessel equation at $x = 0$, the error integration procedure can be expected to be much less critical than in the case of $J(x)$, so that an interval dx which is adequate for J should suffice for C.

INITIAL VALUES. To start the Runge-Kutta process at $x = x^{\circ} = \alpha y^{\circ}$ it is necessary to know $u(x^{\circ})$ and $du(x^{\circ}, dx) = u'(x^{\circ})\,dx$.

For J the code computes these from the first three terms of the

power series expansions (20) and

$$J_n(x) = \sum_{k=0}^{\infty} \frac{(-1)^k x^{n+2k}}{2^{n+2k}\, k!\, (n+k)!}$$

$$J'_n(x) = \sum_{k=0}^{\infty} \frac{(-1)^k (n+2k)\, x^{n+2k-1}}{2^{n+2k}\, k!\, (n+k)!}$$

for a relative accuracy of 10^{-8} over the relevant ranges of n and of $x = \alpha y$; the floating point operation is needed to preserve this relative accuracy in computing J and J' .

Although the expression for C is more complicated than that for J , the integration is started more easily since we can use the natural initial argument $y = \eta > 0$ $(x = \alpha y = \alpha\eta)$ rather than avoiding the natural argument $y = 0$ as in the case of J . Then

$$C_{njn}(\alpha_{njn}\eta) \equiv C_{njn}(\alpha\eta) = Y_n(\alpha\eta)\, J_n(\alpha\eta) - J_n(\alpha\eta)\, Y_n(\alpha\eta) = 0$$

$$\left[\frac{d}{dx} C_{njn}(\alpha_{njn}\eta)\right]_{y=\eta} \equiv C'_{njn}(\alpha\eta) =$$
$$= Y_n(\alpha\eta)\, J'_n(\alpha\eta) - J_n(\alpha\eta)\, Y'_n(\alpha\eta) =$$
$$= W[Y_n(\alpha\eta), J_n(\alpha\eta)] = -\frac{2}{\pi(\alpha\eta)}$$

where $W[u(x), v(x)]$ is the Wronskian determinant of u and v with respect to x (cf. Watson, "Bessel Functions" (1948) p. 76.).

Then the code for C will differ from the code for J principally in having a different starting value y_0 , and in having a subroutine which supplies $C(\alpha\eta)$ and $C'(\alpha\eta)$ -- instead of $J(\alpha y_0)$ and $J'(\alpha y_0)$ -- as initial values of u and u' .

Since $C'(\alpha\eta) \neq 0$, the zero of C at $y = \eta$ is only of first order (instead of the n -th order as for J) so that the "floating" of u could have been omitted in this case if desired.

CODING FEATURES. A basic flow diagram of the code follows. Four features of the code are described below.

Subroutines. To avoid later duplication of effort, and for convenience in checking and modification, the code was written chiefly in terms of closed subroutines as follows:

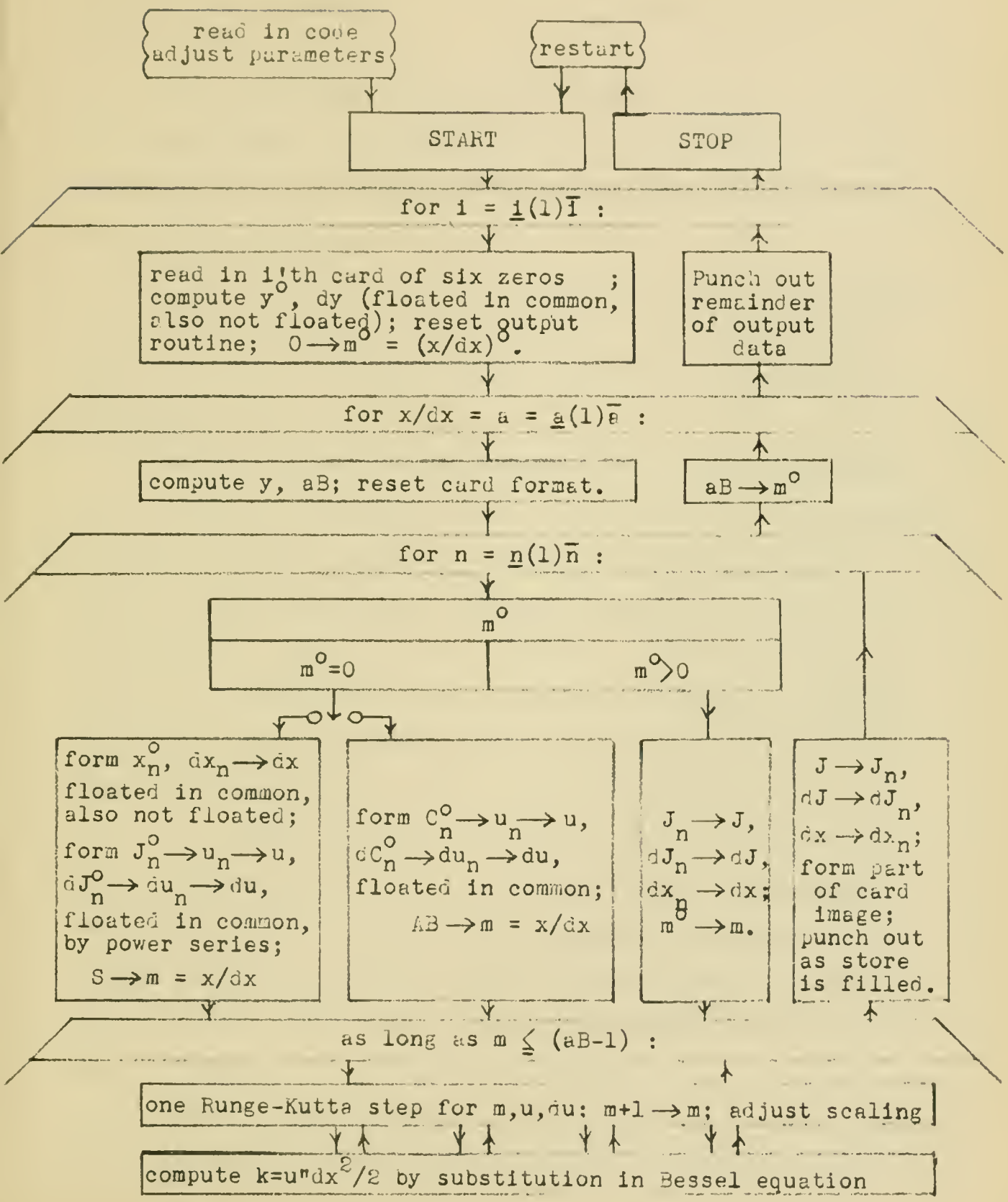
General purpose routines: second-order Runge-Kutta integration; decimal scaling and shifting of several quantities in common; binary-to-decimal conversion of integers and machine numbers; formation of "card-images" and punch-outs of blocks of output data. These routines are adaptable for re-use in other problems with various amounts of modification.

Special purpose routines: main control code; starting values for J and C ; substitution in Bessel differential equations; routine for format of output cards.

Subroutine format. The subroutines were written in a special format to facilitate re-use. It is desirable to write subroutines in some "relative" notation so that self-references in a subroutine may be properly adjusted depending on the differing locations of the subroutine in storage during uses in different codes. It is also convenient during code checking, and also in case relativization is done manually rather than automatically, that the actual address references in a subroutine shall be readily relatable to the relative form in which the subroutine was written.

At the Institute for Advanced Study computer it is customary to refer to each of the $2^{10} = 1024$ storage locations by a pair of "pentads", where a pentad is a "digit" in the base $2^5 = 32$, but is commonly written

BESSEL FUNCTIONS



decimally, i.e. 00 through 31. Thus (29,07) represents $29 \cdot 32 + 7 = 935$.

A convenient relativizing notation is then obtained by using the first pentad symbol as a relativizing symbol -- e.g. addresses of (0,15) or (1,07) relative to the start of the subroutine will be written (R , 15) or (R + 1, 07).

If the subroutine is now put into the store starting at an address N , then in customary fashion the symbol R in an address may be used to discriminate on whether or not to add N to the remainder of the address.

The special feature of the format is that if the starting address N is chosen as a multiple of 32: $N = n \cdot 32 = (n,00)$ -- as is possible if there is adequate intermediate storage, or if the code is not too large -- then the relativization consists in replacing the symbol R by the number n . Thus if $n = 17$, from (R , 15) we get (17,15) where the 17 identifies the current location of the subroutine, and the 15 is still a relative address within it. For code checking, and for manual relativization, this feature is quite a convenience.

Output and input. The output is on decimal IBM cards each containing three values of w , in floating decimal form (an exponent and eight figures, more figures than the expected accuracy), as well as identifications (y , j , etc.). For tabulation convenience these three values are for a common argument, and various parameter values, rather than vice-versa.

This means that the several integrations for the different parameter values must be carried forward in parallel; in fact, for chosen j

and the code carries forward the integration from $y - \Delta y$ to y, in steps of dy, for each of $n = 1, 2, \ldots, 6$ in succession; that is, six different equations are integrated at a time.

The zeros α were supplied as intermittent input (as called for by the code) on IBM cards each containing up to six zeros α for fixed j and η (identified on the card) and various n, and having sum 0 for check purposes.

Parameters. It is desirable that the tabular interval Δy, starting value $\dot{y}$, and integration interval dy by readily adjustable. Rather than being set as decimal or binary fractions which would have to be known or computed, they were defined in terms of integer ratios A, B, S, from which the code computes, in floating form,

$$\Delta y = 1/A$$
$$dy = \Delta y / B$$
$$\dot{y} = S\, dy$$

Thus for $\Delta y = .01$, $dy = .0005$, $\dot{y} = .01$, we would set $A = 100$, $B = S = 20$.

These parameters, and others defining the number of input cards to be processed and the ranges of n and x for which computations are to be made, are contained in the code pre-set to convenient values, but may be altered as desired before the computations. The code adjusts for the current values of the parameters as necessary, each time it is run from the starting point.

RESULTS. The preliminary results obtained were briefly catalogued earlier. The integration steps $dx = \alpha\, dy$ are defined by a common dy,

which was tentatively chosen as .001. (The computation time is inversely proportional to dy .) From the previous error estimates it may be expected that the resulting accuracy for C will be good, and for J will range from good for $n = 1$ to fair for $n = 6$.

The first and easiest test on the output results, aside from their general character and plausibility, is how well their values approximate the expected value 0 at $y = 1$.

The greatest errors at $y = 1$ were 4.4×10^{-7} for J , at $J_{3,4}$, and 6.1×10^{-7} for C , at $C_{6,1,.5}$. But these figures are deceptive in two ways. On the one hand, these errors are principally ascribeable to slight errors in the available values of the zeros α which were used as input data; hence the integration error at $y = 1$ is considerably smaller. On the other hand, the accuracy at $y = 1$ is only a partial measure of the integration accuracy, and at other values of y the errors are several orders greater. These matters are discussed in greater detail below.

Suspicion about the accuracy of the available values of the roots α had been aroused by the erratic nature of the small errors at $y = 1$. Desk-calculator interpolation in standard tables of Bessel functions (where the arguments are decimal fractions of x rather than of y) confirmed that some are in error. The following values in Watson's "Bessel Functions" (1948), pp. 749-750, need correction:

for 16.2234640 read 16.2234662

for 19.4094148 read 19.4094152

for 7.5883427 read 7.5883424

To test the accuracy of the values of J and C for arguments

$y \neq 1$, a number of values were computed to 8 or 10 decimals on a desk-calculator either by interpolating in published tables or by using the power series expansion for J . These laborious processes could be carried out for only a selected few of the thousands of combinations of parameters and arguments.

In the case of C , the values checked (near maxima of C) had relative errors of less than 10^{-7}. Since the largest computed C was less than 10, it seems likely that the results for C have absolute errors of less than 10^{-6}.

In the case of J the errors are somewhat larger. As expected, they are largest for $n = 6$ where over most of the range of x , the relative error is nearly constant at about -4×10^{-4} (giving a maximum absolute error of about -1.5×10^{-4}). This is explained as follows:

Well away from $y = 0$ the Runge-Kutta method is adequately accurate with the chosen interval $dx = .001$, as testified by the accuracy of C . But over a limited range near $y = \bar{y} = .01$, it is not so accurate, as noted earlier. Imagine that the integration method is subject to small relative errors for $y \leq \bar{y}$, not too far from 0, and is exact for $y \geq \bar{y}$. Then the function obtained for $y \geq \bar{y}$, instead of being $J_n(\alpha y)$, is that exact solution $\tilde{u}$ (of the differential equation) for which $\tilde{u}(\alpha \bar{y})$ and $\tilde{u}'(\alpha \bar{y})$ agree with the results of the approximate integration to $\bar{y}$. Then, as a solution of the differential equation,

$$\tilde{u}(\alpha y) = (1+\epsilon_1) J(\alpha y) + \epsilon_2 \frac{J(\alpha \bar{y})}{Y(\alpha \bar{y})} \cdot Y(\alpha y)$$

where $\epsilon(\alpha \bar{y}) = \epsilon_1 + \epsilon_2$, so that ϵ_1 and ϵ_2 are small. Now

$Y(x)$ is very large for small x, and moderate for larger x; so that for larger x, and J not near zero,

$$\tilde{u}(\alpha y) = \left(1+\epsilon_1+\epsilon_2 \cdot \frac{Y(\alpha y)}{Y(\alpha \bar{y})} \Big/ \frac{J(\alpha y)}{J(\alpha \bar{y})}\right) J(\alpha y) \cong (1+\epsilon_1)\, J(\alpha y)$$

That is, if the integration becomes sufficiently exact before Y becomes too small, then when Y does become small (which happens rapidly for $n = 6$, more slowly for $n = 1$), the error in $\tilde{u}$ relative to J amounts principally to a scale error. This explains how the accuracy of the results at $y = \bar{y} = .01$ and at $y = 1$ are an incomplete test of the over-all accuracy; the accuracy of the vanishing at $y = 1$ is, in this problem, rather a measure of the eventual accuracy of the Runge-Kutta method for y away from 0, and of the accuracy of the zeros α.

It is hoped that in the near future the values of J will be recomputed with smaller integration intervals as necessary to secure greater accuracy. To avoid too great an increase in the total number of integration steps and consequently in machine time and in accumulation of rounding errors, the code may be modified so that dy may be varied, depending on the parameters α, n, j and possibly on x, being chosen very small only where necessary.

1.3 Travelling Wave Amplifier.

The extensive calculation described below was carried out for Dr. W. R. Beam of the Radio Corporation of America's David Sarnoff Research Laboratories in Princeton.

When an electron beam of high space charge density is collimated by a large magnetic field, the velocities of all electrons within the beam are not equal; for a beam of circular cross section, the centermost

electrons travel at the lowest velocity since the electron potential there is lower than at larger radii.

Any electron beam of moderately constant cross section, uniform charge density, and uniform velocity will propagate waves of space charge, at speeds nearly that of the beam. Thses solutions are well known. In the case of high space charge and resulting non-constant velocity over the cross section the waves will be of considerably different form. This was the case treated by the code and the purpose was to examine the space charge wave propagation in detail.

Since an electron beam consisting of two discrete velocity classes of electrons is capable of amplifying any disturbance placed upon it (signal or noise), it is only natural to question whether it is possible, in a beam carrying a continuum of velocity classes, to produce amplification. Analysis of beams of this nature, where every velocity class exists at any point on the cross section, have shown that no amplification is expected. This code was designed to determine if amplification occurred in the important case where the velocity is given as a single-valued function of the radial distance from the center of the beam.

If we assume that the perturbations propagate as $e^{i(\omega t + \eta z)}$, then the perturbations equation (holding only for small perturbations in the electron beam) is

$$\frac{d^2 E_z}{dr} + \frac{1}{r}\frac{dE_z}{dr} + (k^2 - \eta^2)\left[1 - \frac{\omega_p^2}{(\omega + \eta v_0(r))^2}\right] E_z = 0$$

where

E_z = Perturbation of the electric field in the direction of

motion of the beam (z -axis)

r = radius from center of beam

k = ω/c

ω = angular frequency of disturbance

c = velocity of light in a vacuum

η = propagation constant of the beam (to be determined)

$v_0(r)$ = velocity of the electrons as a function of radius:

$$v_0(r) = v_{00} + c_1 r^2$$

$\omega_0^2 = \frac{\rho_0 e}{m\epsilon}$, where ρ_0 is the unperturbed space charge density, e and m are charge and mass of the electron, and ϵ the permittivity of free space.

The values of η are found by unpressing <u>two</u> boundary conditions:

1. $E_z' = 0$ at $r = 0$, required by the absence of a line charge at $r = 0$.
2. $E_z = 0$ at $r = r_0$, the requirement that the field reduce to zero at the radius of a metallic cylinder surrounding the beam at $r = r_0$.

The method of solution chosen is a straightforward one. Choose an arbitrary η , set up the boundary condition $E_z'(0) = 0$, and integrate in a forward direction until r_0 is reached. The values $E_0(r_0, \eta)$ can be considered as values of a complex function of a complex variable, η , and its zeros in the η-plane are the values sought.

It can be easily demonstrated that $E_z(r_0, \eta)$ is an analytic function of η in the upper half plane. By taking a rectangular contour C , in some part of the η -plane, it is possible to determine the number of zeros of $E_z(r_0, \eta)$ within the contour, C , simply by counting the net number of times $E_z(r_0, \eta)]_C$ crosses any radial line in the complex η -plane as η goes around C , for the

function will encircle the axis once for each zero contained within the contour. A four-way check is possible by counting the number of times the four major axes are crossed. This is done simply by comparing signs of the real and imaginary parts of $E_z(\rho_0, \eta)$ for successive values of η.

This routine was also altered so that it could record and print $E_z(\rho_0, \eta)$ for values of η lying along horizontal lines in the η-plane. By study of these it is possible to determine all the values of η which satisfied the boundary conditions, and to catalog them.

The problem was run with a large selection of typical constants, and all tests to determine whether amplification exists were negative. In addition, sets of values of $E_z(\rho_0, \eta)$ were calculated for a large number of η's, and the results showed accurately the position of the various modes of propagation of the electron stream. A graphical display of one of these outputs appears (together with explanatory remarks) at the conclusion of the discussion of this problem.

We now briefly describe the computational procedure. For writing convenience let us denote E_z by y. In the forward integration of the equation (for some fixed η) we employ the Milne method:

$$y'_{n+1} \cong y'_{n-3} + \frac{4h}{3}\left(2y''_n - y''_{n-1} + 2y''_{n-2}\right)$$

to find y'_{n+1}, then Simpson's rule

$$y_{n+1} \cong y_{n-1} + \frac{h}{3}\left(y'_{n+1} + 4y'_n + y'_{n-1}\right)$$

to find y_{n+1}, having done this, y''_{n+1} is found by substituting the differential equation. We see that the first three values of y,

y', and y'' are required in order to start the Milne method. These are provided by using the approximations

$$y_{n+1} \simeq y'_n + \bar{h}\, y''_n \qquad \text{and}$$

$$y_{n+1} \simeq y_n + \bar{h}\, y'_n + \frac{\bar{h}^2}{2}\, y''_n \qquad , \text{where} \quad \bar{h} = \frac{h}{8}$$

for 24 steps of $\bar{h}$.

The errors in this process are now discussed. The error in (4) is $\simeq \left(\frac{h}{8}\right)^3 \frac{d^3y}{dr^3}\Big|_{r=\xi}$ where $\frac{d^3y}{dr^3}\Big|_{r=\xi}$ is a mean value of $\frac{d^3y}{dr^3}$.

The error in the Milne method (3) is $\simeq (4h)^5 \frac{d^5y}{dr^5}\Big|_{r=\xi'}$. The interval $\frac{h}{8}$ in (4) was chosen so that these two error expressions should be in rought agreement. It may be remarked that since all the accuracy required was that the computed value of $y(r_0)$ have its proper sign, the matter of digital accuracy was not overly important.

We now discuss the methods that were employed to check the code before the main calculations were made. Note, that upon setting $\eta = \omega_0^2 = 0$ in (1), the equation reduces to

$$\frac{d^2E_2}{dr} + \frac{1}{r}\frac{dE_2}{dr} + k^2 E_2 = 0$$

This has as solution $J_0(kx)$, where $J_0(x)$ is the (Bessel) solution of the zero order Bessel equation $y'' + \frac{1}{x}y' + y = 0$. This reduction was made in the code and the machine results compared with the W. P. A. Bessel Function Tables to five decimal places. The root count was checked by inserting constant values of v_0 which gave predictable zeros in the η-plane.

The details of the code are now given by the schematic diagrams

(Figures 1 and 2) and the explanatory remarks which follow. The numbers in the remarks correspond to the numbers of boxes in the figure.

EXPLANATORY REMARKS.

Figure 1. Boxes 1 - 9 set up the initial values of all of the quantities and forms $T \cdot 2^{-t}$ where $T = 1 - \frac{\omega_0^2}{(\omega + \eta v_0(x))^2}$ and $t = t$ is the scale factor required to make $|T \cdot 2^{-t}| < 1$.

Boxes 11 - 19 substitute the values of y and y' into the differential equation, yielding y''.

Boxes 20 - 25 are used in the starting calculation to obtain y_1, y_2, y_3; y_1', y_2', y_3' for use in the main calculation. The interval in the starting calculation is $h/4$ where h is the interval in the main calculation.

Boxes 26 - 37 carry out the main calculation to the point $x = x_m$.

Figure 2. In Box 1 P, Q, U, and V are the net number of times $y(x_m, \eta)$ crosses the positive real η-axis, negative real η-axis, positive imaginary η-axis and negative imaginary η-axis, respectively.

By net number of crossings we mean that 1 is added if the crossing is counter-clockwise, and subtracted if the crossing is clockwise.

Boxes 2 and 3 set up the new value of $\eta = \eta_{min} + \Delta R_e \eta + \sqrt{-1} \cdot i \Delta I_m \eta$. Figure 3. In figure 3 are shown modes of propagation of an electron beam. These represent different waves which can be excited by a resonant cavity at the input end of a drift region. The waves are characterized by different wave velocities. In the case represented in (A), the electron velocity at the center of the beam was about 1.4% less than that at the outer radius. The resultant waves cannot have velocities

within the range of electron velocities. In the case of a uniform electron velocity, as in (B), no such forbidden region exists. The differences in wave velocities are thus the only effect of the electron velocity gradient.

The points shown were obtained from solutions of the equation for completely real values of η , by determining values of η for which the boundary conditions were satisfied. This was done following the conclusion that no such solutions existed for complex η . This conclusion was verified by extensive experimental work with demountable electron beam apparatus.

1.4 Accelerating gradient accelerator.

A group at Princeton University under the direction of Professor M. White has conducted an investigation into the design characteristics of an accelerating gradient accelerator. An extensive series of numerical calculations was carried out to investigate the stability of particle orbits under various sets of physical conditions. Of principal interest are those conditions under which the particle returns in the same phase after each revolution around the machine. Most of the numerical work was concerned with investigating the effects of small errors in the gradients and positions of the focussing magnets on the stability of the orbits. In addition, some calculations were performed to gain some insight into the effects of non-linearities in the focussing fields.

In what follows r is the distance measured outward from the center of symmetry of a magnet and z is the vertical distance from this point. The machine itself consists of alternate sections of

FIGURE NO. 1. (INTEGRATION OF DIFFERENTIAL EQUATION)

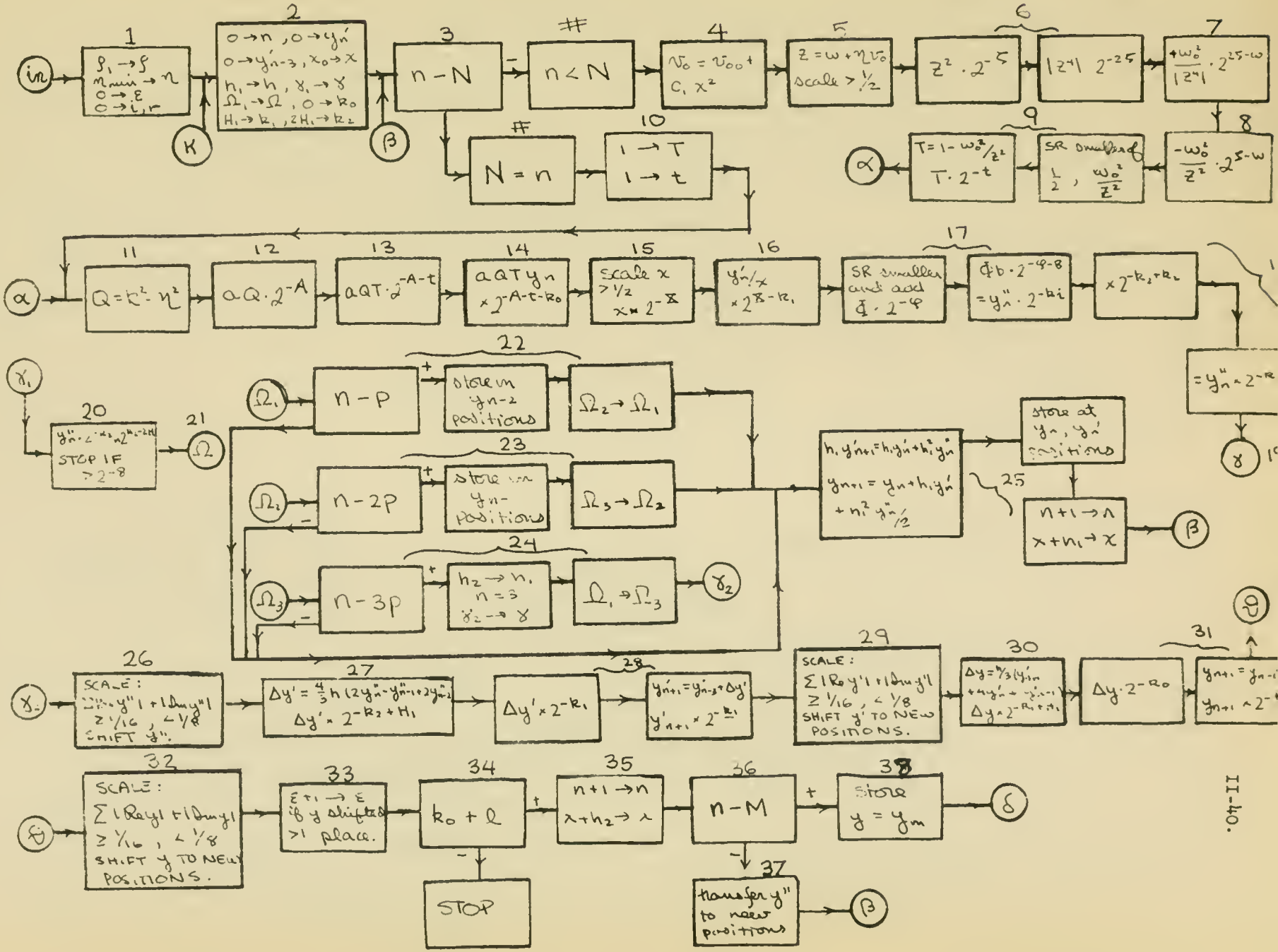
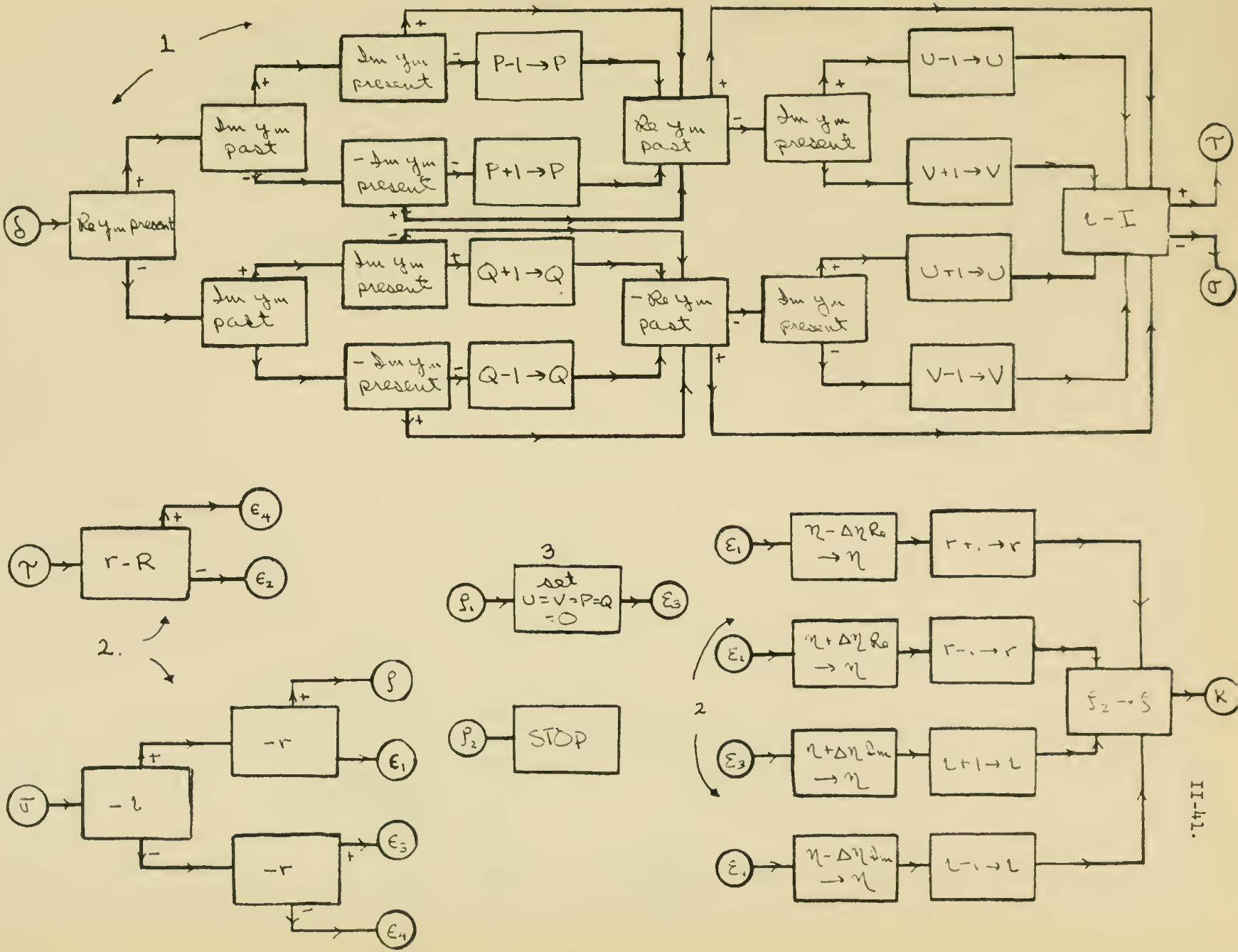
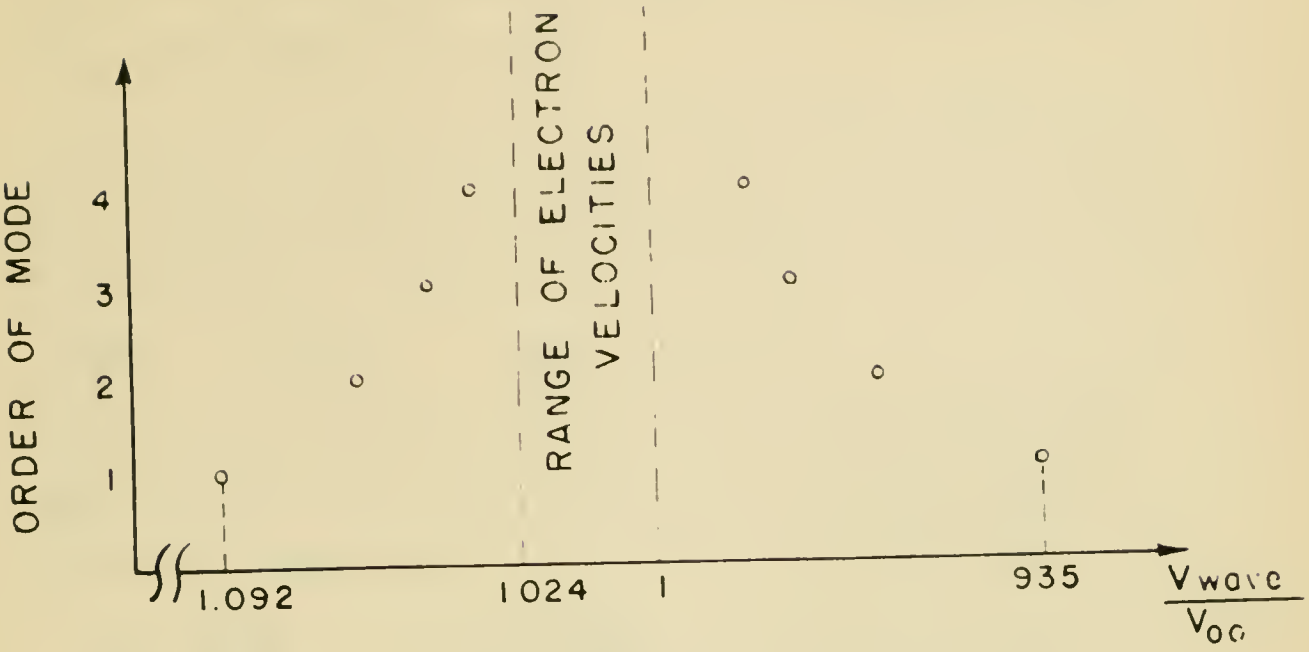


FIGURE NO. 2 (ROOT COUNT)

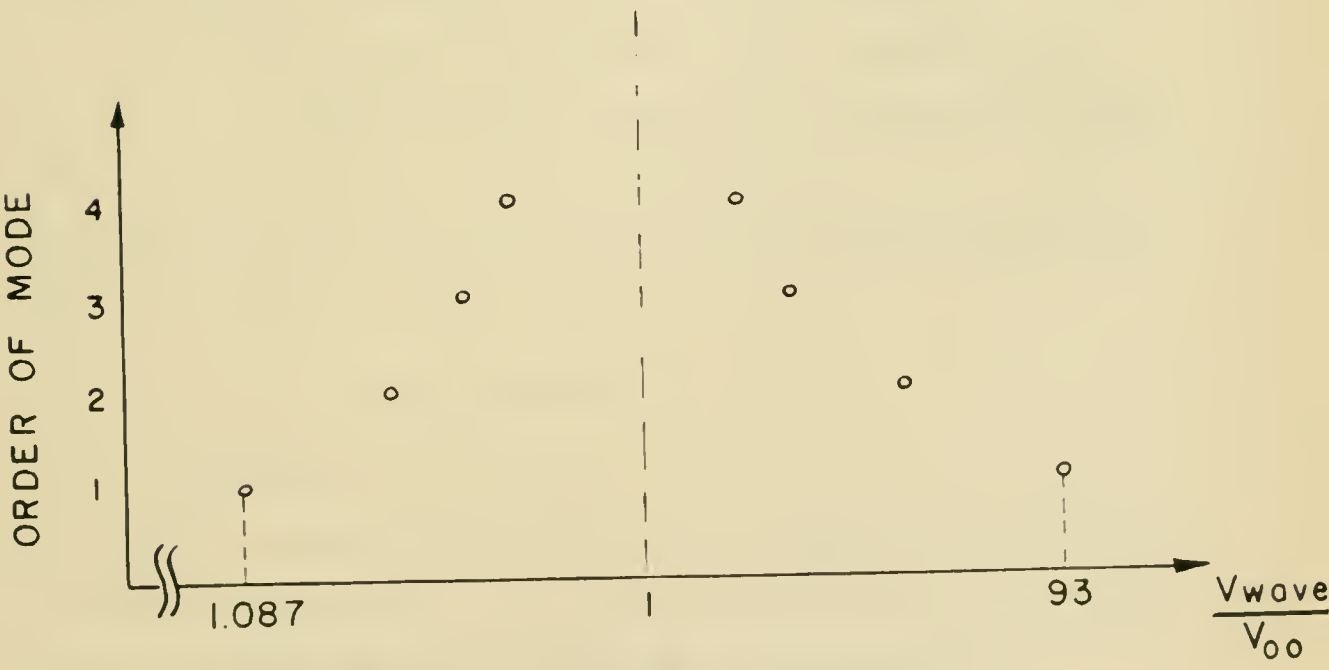




(A) BEAM WITH APPROXIMATELY 2.4 % SLIP

$\frac{\omega}{\omega_0} = 10$; $V_0 = 100^{v.}$; BEAM RADIUS = 1 mm. ;

f = 3000 mc. ; I ≃ 6 ma. ; BEAM FILLS DRIFT TUBE



(B) BEAM WITH UNIFORM VELOCITY Voo

SAME CONSTANTS

four-pole focussing magnets and guide magnets. The components of magnetic field strength are of the form

$$H_r = \beta + \gamma z - \delta r + 2\epsilon rz + \xi(r^2 - z^2) + \eta(r^3 - 3rz^2) + \vartheta(3r^2 z - z^3)$$

$$H_z = \alpha + \gamma r + \delta z + \epsilon(r^2 - z^2) - 2\xi rz - \eta(3r^2 z - z^3) + \vartheta(r^3 - 3rz^2)$$

and the equations of motion of a particle are then

$$\frac{dp_r}{d\tau} = -e\frac{v}{c}H_z \quad , \quad \frac{dp_z}{d\tau} = e\frac{v}{c}H_r$$

where p is the momentum of the particle, τ the time, e/c is the particle charge, and v is the velocity. If t measures distance along the line perpendicular to the r and z directions, then

$$p_r = p_t \frac{dr}{dt} \quad , \quad p_z = p_t \frac{dz}{dt} \quad , \quad t \sim v\tau$$

Thus we have approximately

$$\frac{d^2 r}{dt^2} = -\frac{e}{c p_t} H_z \quad , \quad \frac{d^2 z}{dt^2} = \frac{e}{c p_t} H_r \quad ;$$

in these equations the dependence of p_t on t is neglected.

In a focussing sector the values H_r , H_z appearing in (2) are given by (1) whereas in a guide sector they are considerably simpler. In fact, if r_0 , r_0' , z_0 , z_0' are the values of r , r' , z , z', at the beginning of such a sector and if the corresponding values at the end are r , r' , z , z' , then

$$r_1 = r_0 \cos\alpha\omega + r_0' R \sin\alpha\omega$$

$$r_1' = -\frac{r_0}{R}\sin\alpha\omega + r_0' \cos\alpha\omega$$

with a pair of similar equations for z_1 , z_1' , where R is the radius of curvature of the orbit, α is a certain constant and ω is the angle swept out in moving from one end of the sector to the other.

The calculations themselves consisted of following the path of a particle through alternate focussing and guide sectors with alternating signs for the gradients in alternate focussing sectors. Equations (2) were integrated by the Runge-Kutta method. The coefficients α , , ..., ν were supposed to be subject to small random errors due to gradient and alignment discrepancies. To simulate this condition each coefficient was subjected to a small perturbation chosen from a uniform distribution. For a discussion of how such numbers may be generated see the section on random numbers. Thus each coefficient a in (1) was of the form

$$a = a_0 + \delta_a R$$

where R is the random number corresponding to the given magnet, δ' is a factor determined by the tolerances set for the machine and a_0 is the unperturbed value of the coefficient.

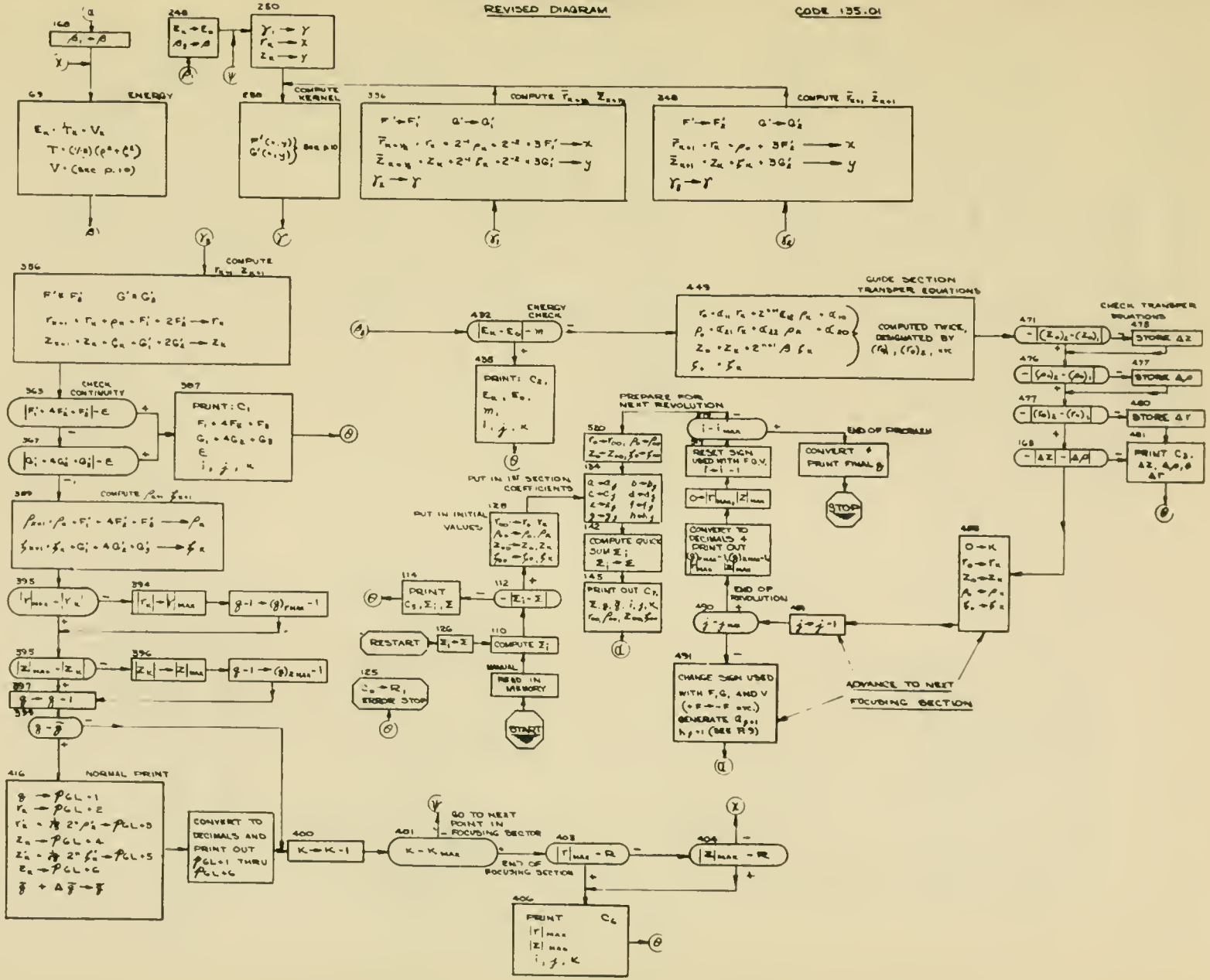
To ensure the accuracy of the results several checks were introduced in the form of redundancies. The most important of these was an energy check. The force functions are derivable from a potential function V and thus the total energy could be expressed as V + the kinetic energy $(r'^2 + z'^2)/2$, and this must be a constant over each focussing sector. The energy was calculated at the beginning and at the end of each sector and required to agree to a pre-assigned tolerance.

About 400 integrations were done in all. Most of these were concerned with the linear case, i.e. that one where H_r , H_z contain no terms higher than the first in r and z . These results are now being analyzed. Some work was also done with the non-linear case but this was not as complete as the corresponding set for the linear one.

ORBIT STABILITY

REVISED DIAGRAM

CODE 135.01



II. OTHER MATHEMATICAL PROBLEMS

2.1 Flows past curved obstacles.

In the description of plane cavity flows past curved obstacles there is a highly non-linear integral equation which G. Birkhoff and E. Zarantonello showed plays a central role. To describe this equation suppose $\lambda(\sigma)$ is a real-valued function on the interval $(0, \pi)$ and let

$$\vartheta(\sigma) = \int_{\pi/2}^{\sigma} \lambda(\sigma')\, d\sigma' \quad , \quad \tau(\sigma) = \int_{0}^{\pi} D(\sigma, \sigma')\, \lambda(\sigma')\, d\sigma' \quad ,$$

with

$$D(\sigma, \sigma') = \frac{1}{2\pi} \ln \left\{ \frac{\tan \sigma/2 + \tan \sigma'/2}{\tan \sigma/2 - \tan \sigma'/2} \right\}$$

Then the equation in question is this:

$$\lambda(\sigma) = M K[\vartheta(\sigma)]\, \nu(\sigma)\, e^{-\tau(\sigma)} \quad ;$$

more precisely given the functions K , ν we wish to find a function $\lambda(\sigma)$ and a constant M such that relations (1), (2), (3) are satisfied as well as a condition on M of the form

$$f(M, \lambda) = 0$$

In solving this system (1) - (4) it is convenient to resort to a Fourier transformation. The functions ϑ and τ are simply expressible in terms of the Fourier coefficients of λ . In fact, if

$$\lambda(\sigma) = \sum_{n=1}^{m} n\, a_n \sin n\sigma \quad ,$$

then

$$\vartheta(\sigma) = \sum_{n=1}^{m} a_n \cos n\sigma \quad , \qquad \tau(\sigma) = \sum_{n=1}^{m} a_n \sin n\sigma$$

The method of solution of the integral equation is inductive and will be described below.

The continuous variable σ is replaced by the discrete variables σ_k ($k = 1, 2, \ldots, 24$), $\lambda(\sigma)$ by $\lambda_k = \lambda(\sigma_k)$, $\theta_k = \theta(\sigma_k)$ $r_k = r(\sigma_k)$, where $\sigma_k = 3.75^\circ + 7.50^\circ (k-1)$. Then the Fourier coefficients a_n in (5) above are obtainable from the relations

$$a_n = \frac{2}{mn} \sum_{k=1}^{m} \lambda_k \sin n\sigma_k \qquad (n \neq m)$$

$$= \frac{1}{m^2} \sum_{k=1}^{m} \lambda_k \sin n\sigma_k \qquad (n = m)$$

We go to show the validity of these relations. Suppose that

$$\lambda(\sigma) = \sum_{n=1}^{m} c_n \sin n\sigma$$

is a trigonometric polynomial which takes on the value λ_k at $\sigma = \sigma_k$ (cf. above) ($k = 1, 2, \ldots, m$). Then

$$d_l = \sum_{k=1}^{m} \lambda_k \sin l\sigma_k = \sum_{n=1}^{m} c_n \sum_{k=1}^{m} \sin n\sigma_k \sin l\sigma_k$$

$$= 2 \sum_{n=1}^{m} c_n \sum_{k=1}^{m} (\cos(n-l)\sigma_k - \cos(n+l)\sigma_k) = \begin{cases} \frac{1}{2} c_\lambda m & l \neq m \\ c_m m & l = m \end{cases}$$

Thus

$$c_n = \frac{2}{m} d_n \qquad n \neq m,$$

$$= \frac{1}{m} d_m \qquad n = m,$$

and we then have established the relationship between (5) and (7).

It is convenient in what follows to define μ_k to be

$$\mu_k = K(\theta_k) \nu(\sigma_k) e^{-\sigma_k}$$

and its Fourier coefficients to be b_n. Thus

$$b_n = \frac{2}{m} \sum_{k=1}^{m} \mu_k \sin k\sigma_k \qquad k \neq m$$

$$= \frac{1}{m} \sum_{k=1}^{m} \mu_k \sin k\sigma_k \qquad k = m$$

In terms of the b_n the condition on M mentioned above assumes the form

$$M = g(b)$$

Our numerical problem is now this: For a given K, ν and g to find a λ_k such that

$$\lambda_k = M \mu_k$$

where M is determined by (10). Suppose $\lambda_k^{(n)}$ is an approximant to λ_k and M_n to M. We then find the Fourier coefficients $a_n^{(n)}$ of $\lambda_k^{(n)}$ with the help of the definitions (7) and with these we construct $\tau_k^{(n)}$, $\sigma_k^{(n)}$ from (6). We are now able to form $\mu_k^{(n)}$ from (8) and also $b_n^{(n)}$ from (9). In terms of these $b_k^{(n)}$ we use (10) to form an error correction quantity

$$\Delta_n = g(b^{(n)}) - M_n$$

and a new approximant M_{n+1} to M by means of the relation

$$M_{n+1} = M_n + (\operatorname{sgn} \Delta_n) c \epsilon \operatorname{Min.}(|\Delta_n|, \eta),$$

where c, ϵ and η are pre-assigned by a human judgment as to the rate of convergence of M_n and $\lambda_k^{(n)}$. They may be varied as the computation proceeds if it seems desirable. In practice this was seldom done.

We are now in a position to form $\lambda_k^{(n+1)}$

$$\lambda_k^{(n+1)} = \epsilon M_{n+1} \mu_k^{(n)} + (1 - \epsilon) \lambda_k^{(n)}$$

as a weighted average of M_{n+1}, $\mu_k^{(n)}$ and of $\lambda_k^{(n)}$. This completes the induction.

At this point we remark that one can calculate $a_k^{(n+1)}$ in two different ways: given $\lambda_1^{(n+1)}$ we can use the equations (7) and obtain one estimate $a_k^{(n+1)}$ and also

$$\bar{a}_k^{(n+1)} = \epsilon\, M_{n+1}\, \lambda_k^{(n)} + (1-\epsilon)\, a_k^{(n)}$$

to obtain another one. Thus we are able at this point to check against possible machine errors by forming the difference

$$\left| a_k^{(n+1)} - \bar{a}_k^{(n+1)} \right|$$

Two criteria were established for determining when the inductive process would terminate. The first of these was concerned with the nearness of $\lambda_k^{(n+1)}$ to $\lambda_k^{(n)}$ and consisted of measuring the difference

$$\left| \lambda_k^{(n+1)} - \lambda_k^{(n)} \right|;$$

the second was concerned with the nearness of M_{n+1} to M_n and consisted of forming the difference

$$\left| M_{n+1} - M_n \right|.$$

The computation was concluded when the former error was less than 10^{-4} and the latter less than about $10^{-5.4}$. This latter condition turned out to be the more stringent.

A number of checking procedures in addition to the one mentioned earlier were incorporated into the code for the problem. Due to the symmetry introduced into the problem all a_{2n} must vanish. This was checked by comparing $|a_{2n}|$ at each stage to 2^{-28} to verify that it did vanish at least to this precision. Also since reasonable bounds could be given for allrelevant quantities - λ, τ, θ, K, M,

γ' -- their sizes were monitored at each step. In all cases the calculations were stopped when any of these "error checks" was violated.

The various kernels K (cf. below) were constructed, as needed, by the code itself with the help of a few parameters which were fixed on special input cards for each case. These functions K as well as the $\sin h\sigma_k$, $\cos h\sigma_k$ were formed with the help of a table of $\sin \sigma_k$, $\cos \sigma_k$ stored for the purpose in the memory.

All quantities were handled with a precision of 39 binary places throughout the calculation and the relevant results were converted to 9 place decimal numbers at the end of each special case.

There were seven functions $\gamma(\sigma)$ and nine obstacle functions $K(\theta)$ in addition a choice of five side-conditions was used to determine

2.2 Eigenvalues of symmetric matrices.

The proper values and the associated vectors for several symmetric matrices of order 16 and for several of lower order were obtained by means of a procedure due to Jacobi. This method, which is a highly stable one (this has been shown in an as yet unpublished paper by von Neumann, Murray and Goldstine) brings all proper values into evidence at the same rate. It is inductive in character. We shall now describe it in some detail.

In what follows we shall have occasion frequently to deal with several quantities associated with symmetric matrices. If $B=(b_{ij})$ is a symmetric matrix of order , we define certain quantities with the help of the relations

$$N^2(B) = \sum_{i,j} b_{ij}^2 \quad , \quad \mathcal{C}^2(B) = \sum_{i \neq j} b_{ij} \quad , \quad T(B) = \sum_i b_{ii}$$

We formulate our problem in this fashion: Given a symmetric matrix A of order n we seek a unitary matrix U such that

$$U^* A U = \mathfrak{D}$$

where U^* is the transpose of U and $\mathfrak{D}$ is a diagonal matrix, i.e.

$$d_{ij} = \begin{cases} 0 & i \neq j \\ \lambda_j & i = j \end{cases}$$

We note at once that

$$\tau^2(\mathfrak{D}) = 0$$

Thus $\tau^2 \geqq 0$ and is zero if and only if $\mathfrak{D}$ is of the form given by (3). Our procedure will be to define a sequence of "simple" unitary matrices U_k and transforms of A

$$A_0 = A \ , \quad A_k = U_k^* A_{k-1} U_k \qquad (k = 1, 2, \cdots)$$

such that

$$\tau^2(A_k) < \tau^2(A_{k-1}) \qquad (k = 1, 2, \cdots)$$

We say that a unitary matrix $U = (u_{ij})$ is simple in case there are two integers i', j' between 1 and n with $i' \neq j'$ and a number φ such that

$$u_{ij} = \begin{cases} \delta_{ij} & i \neq i' \text{ or } j \neq j' \\ \cos\varphi & i = i', \ j = i' \\ \sin\varphi & i = i', \ j = j' \\ -\sin\varphi & i = j', \ j = i' \\ \cos\varphi & i = j', \ j = j' \end{cases}$$

If U is simple and φ is so chosen that

$$\bar{A} = U^* A U$$

has the element $\bar{a}_{i'j'} = 0$, then it is easy to see that

$$\tau^2(\bar{A}) = \tau^2(A) - a^2_{i'j'}$$

Thus simple unitary matrices of the sort just mentioned reduce the value of τ^2 .

If in particular we choose i' , j' so that

$$a^2_{i'j'} \geq 2\tau^2(A)/n(n-1)$$

i.e. if $a^2_{i'j'}$ is at least average in size, then (6) implies that

$$\tau^2(\bar{A}) \leq \tau^2(A)\left(1 - \frac{2}{n(n-1)}\right) < e^{-2/n(n-1)}\ \tau^2(A) ,$$

or

$$\tau(\bar{A}) < e^{-1/n(n-1)}\ \tau(A)$$

We see now how our inductive process proceeds. At a given step say that A_{k-1} has been defined, and that $(a^{k-1}_{i'j'})^2$ is an at least average-sized element -- the larger it is the more quickly does the process proceed. Then a simple unitary matrix is chosen so that the element $a^{(k)}_{i'j'}$ in A_k defined by (4) is null. We then have

$$\tau(A_k) < e^{-1/n(n-1)}\ \tau(A_{k-1}) < \cdots < e^{-k/n(n-1)} \cdot \tau(A)$$

Thus at a cost of $k \sim n^2$ operations we reduce $\tau(A)$ by a factor $1/e$. Each operation consists of altering only two rows and two columns of a matrix of order n by forming simple linear combinations. In all about $4n$ multiplications are involved in each operation and hence each reduce of τ by a factor $1/e$ requires about $4n^3$ multiplications.

It is important to see how the diagonal elements $a_{ii}^{(k)}$ are related to the proper values λ_i of A. To this end we note first that the proper values of A_k are the same as those of A and second that we can decompose A_k into

$$A_k = B_k + C_k$$

where B_k is the diagonal matrix having $a_{ii}^{(k)}$ in the i-th position on the diagonal. Then by a well-known result of Hilbert and Courant

$$|\lambda_i - a_{ii}^{(k)}| \leq N C_k = 2^{1/2} \tau(A_k)$$

Thus τ is a valid measure of the rate of convergence of the diagonal elements of A_k to the proper values.

We remark in passing that the unitary matrix U in (2) will be approximated by the product $U_1, U_2, \ldots$ of the simple unitary matrices. Further we observe that this method is not an unreasonable one to use for inverting a matrix since the inversion of U is automatic and that of D requires exactly n divisions. With this method we see at once the loss in accuracy which will occur since we can exactly determine how much scaling must be performed to invert the diagonal elements.

The estimate given in (8) is apparently not a sufficiently sharp inequality to give an accurate picture of the number of steps needed to ensure a given precision. Some experimental work in this connection is being done at the University of Illinois under A. H. Taub.

Two versions of the code were prepared. In one the numerically largest off-diagonal element was located at each step and in the other only one larger than average was located. In each code several checks

were performed. The traces and norms of the A_n were evaluated and compared to the starting values of these quantities and the calculation stopped if either exceeded a pre-assigned tolerance. Also after the calculation was completed and the matrices U and D, i.e. B_k for some k (cf. (9) above) obtained the expression

$$N(A - U B_k U^*)$$

was evaluated to ensure that this was zero to the requisite precision.

2.3 Spherical blast wave.

Some exploratory calculations were made on a formulation of a spherical blast wave problem. Since these indicated the desirability of using different variables we give below an account of the problem in terms of the new quantities. We interest ourselves in the shock wave started by a very short duration impulse of finite energy acting in an ideal gas in three dimensions with adiabatic constant γ. We assume that all motions take place in a direction we choose as the r-axis and we can treat our problem as if it had but one spatial dimension. We further assume that at time $t = 0$ and at position $r = 0$ the impulse acts. We wish then to study the motion of the shock wave as it travels into the undisturbed gas. In the case where the pressure in this undisturbed gas is zero there is an exact theory for the motion given by von Neumann. We shall make use of his results for certain estimates we make below.

At each point r, t we denote by u, p, ρ the velocity, pressure and density of the gas and by 0, p_0, ρ_0 their values in the undisturbed gas. We shall adopt a Lagrangean point of view and shall study the history of that particle which at $t = 0$ was at r;

we denote its location at t by $R(r,t)$. In the undisturbed gas $R = r$. The equation of the shock wave we give as $r = \mathcal{R}(t)$

The differential equation of motion in the region enclosed by the shock wave is given by the well-known laws of hydro-dynamics to be

$$\rho \frac{\partial^2 R}{\partial t^2} + \frac{\partial p}{\partial R} = 0 ,$$

i.e.

$$\frac{\partial^2 R}{\partial t^2} = -\frac{1}{\rho}\left(\frac{\partial R}{\partial r}\right)^{-1} \cdot \frac{\partial p}{\partial r}$$

Conservation of mass assures us that $R^2(\partial R/\partial r)\rho$ is constant in t along each particle path, i.e.

$$\rho = \rho_0 r^2 R^{-2} (\partial R/\partial r)^{-1}$$

Hence (2) becomes

$$\frac{\partial^2 R}{\partial t^2} = -\frac{1}{\rho_0} \cdot \frac{R^2}{r^2} \cdot \frac{\partial p}{\partial r}$$

We find it convenient now to change variables from r to $s = r^2/2$. In these terms (4) becomes

$$\frac{\partial^2 R}{\partial t^2} = -\frac{1}{\rho_0} \frac{R^2}{(2s)^{1/2}} \frac{\partial p}{\partial s}$$

We next give the shock conditions, i.e. those conditions on u , p , ρ , which are dictated by the requirements of conservation of mass, momentum and energy across the shock. Let the values of u , p , ρ on the disturbed side of the wave be u' , p' , ρ' and let the value of the shock velocity be v . Then

$$\frac{\rho'}{\rho_0} = \frac{(\gamma+1)p + (\gamma-1)p_0}{(\gamma-1)p + (\gamma+1)p_0}, \qquad \mu' = \sqrt{\frac{2}{\rho_0}} \frac{|p - p_0|}{(\gamma+1)p + (\gamma-1)p_0}$$

$$v = \pm\sqrt{\frac{1}{2\rho_0}} \cdot \sqrt{(\gamma+1)p + (\gamma-1)p_0}$$

Next we recall that

$$p = A\rho^{\gamma}$$

where A is independent of t along each particle path as long as it does not corss the shock, i.e. it is independent in the entire region behind the shock. Thus

$$p(s,t)\rho^{-\gamma}(s,t) = p(s,t')\rho^{-\gamma}(s,t') = A(s)$$

provided both (s,t) and (s,t') are on the same side of the shock. We therefore have

$$p(s,t) = p(s,t')\left(\frac{\partial R^3(s,t)}{\partial s^3}\right)^{\gamma}\left(\frac{\partial R^3(s,t')}{\partial s^3}\right)^{-\gamma} =$$

$$= p(s,t')\left(\frac{\partial R^3(s,t)}{\partial s^3}\right)^{\gamma}\left(\frac{\partial R^3(s,t')}{\partial s^3}\right)^{-\gamma}$$

We now formulate our difference equation system. We divide the s-axis from 0 to shock $s(t) = R^2(t)$ into 100 equal intervals, each of length $s(t)/100$ and the t-axis into intervals whose length will depend on t. This point is discussed in more detail below.

We now introduce some notations that will be advantageous in what follows. Suppose that $dt_{n+1/2}$ (n = 0, 1, 2, ...) are the intervals in t. Then

$$t_0 = 0, \quad t_{n+1} = t_n + dt_{n+1/2} \qquad (n = 0, 1, 2, \ldots),$$

$$s_n = s(t_n), \quad ds_n = s_n/100, \quad R_n = (2s_n)^{1/2}$$

$$R_j^n = R(j\,ds_n, t_n), \quad \tilde{p}_{j-1/2}^{n} = p((j-1/2)\,ds_n, t_n),$$

(ontd.) $\bar{R}_j^{n+1} = R(j\,ds_n,\ t_{n+1})$, $\bar{p}_{j-1/2}^{n+1} = p((j-1/2)\,ds_n,\ t_{n+1})$,

$\bar{\bar{R}}_j^{n-1} = R(j\,ds_n,\ t_{n-1})$, $\hat{R}_j^{n+1} = R_j^n + \frac{dt_{n+1/2}}{dt_{n-1/2}}\left(R_j^n - \bar{\bar{R}}_j^{n-1}\right)$

In terms of these quantities equation (5) has been written as

$$\bar{R}_j^{n+1} = \hat{R}_j^{n+1} + \frac{1}{\rho_0}\left(p_{j-1/2}^n - p_{j+1/2}^n\right)\frac{\left(R_j^n\right)^2}{j^{1/2}}\,\frac{dt_{n+1/2}}{ds_n}\left(\frac{dt_{n+1/2} + dt_{n-1/2}}{2\,ds_n}\right)\frac{\mathcal{R}_n}{20}$$

We go to show that (9) is an estimate for (5). Consider first the expression

$$\bar{R}_j^{n+1} - \hat{R}_j^{n+1} = R(j\,ds_n,\ t_{n+1}) - \hat{R}(j\,ds_n,\ t_{n+1}) =$$

$$= R(j\,ds_n,\ t_{n+1}) - R(j\,ds_n,\ t_n) - \frac{dt_{n+1/2}}{dt_{n-1/2}}\left(R(j\,ds_n,\ t_n) - R(j\,ds_n,\ t_{n-1})\right)$$

$$\sim \frac{\partial R}{\partial t}\Big|^{s=j\,ds_n,\ t=t_{n+1/2}} dt_{n+1/2} - \frac{\partial R}{\partial t}\Big|^{s=j\,ds_n,\ t=t_{n-1/2}} dt_{n+1/2} \sim \frac{\partial^2 R}{\partial t^2}\Big|^{s=j\,ds_n,\ t=t_n} dt_{n+1/2}\left(\frac{dt_{n+1/2} + dt_{n-1/2}}{2}\right)$$

Next

$$p_{j-1/2}^n - p_{j+1/2}^n = p((j-1/2)\,ds_n,\ t_n) - p((j+1/2)\,ds_n,\ t_n) \sim -\frac{\partial p}{\partial s}\Big|^{s=j\,ds_n,\ t=t_n} ds_n$$

Finally we note that

$$\frac{\left(R_j^n\right)^2}{j^{1/2}\,ds_n}\cdot\frac{\mathcal{R}_n}{20} = \frac{\left(R_j^n\right)^2}{(2j\,ds_n)^{1/2}}\cdot\frac{\mathcal{R}_n}{(2ds_n)^{1/2}}\cdot\frac{1}{10} = \left[\frac{R^2}{(2s)^{1/2}}\right]^{s=j\,ds_n,\ t=t_n}\cdot\frac{\mathcal{R}_n}{10}\cdot\frac{1}{(2ds_n)^{1/2}} = \frac{R^2}{(2s)^{1/2}}\Big|^{s=j\,ds_n,\ t=t_n}$$

This completes our justification for the estimate (9) to the implicit equation (5).

Now we describe how we find $\bar{p}_{j-1/2}^{n+1}$ given p_j^n , R_{j-1}^n , R_j^n , $\bar{R}_{j-1}^{n+1}$, $\bar{R}_j^{n+1}$. We make use of the adiabatic relation given in equation (7) above as follows:

$$\bar{p}_{j-1/2}^{n+1} = p_{j-1/2}^n\left[\frac{\left(\bar{R}_j^{n+1}\right)^3 - \left(\bar{R}_{j-1}^{n+1}\right)^3}{\left(R_j^n\right)^3 - \left(R_{j-1}^n\right)^3}\right]^{\gamma}$$

To justify this we note that

$$\left(\bar{R}_j^{\nu+1}\right)^3 - \left(\bar{R}_{j-1}^{\nu+1}\right)^3 \sim \frac{\partial R^3}{\partial s^3}\Bigg|^{s=(j-1/2)\,ds_\nu,\ t=t_{\nu+1}} \left[(j\,ds_\nu)^3 - ((j-1)\,ds_\nu)^3\right],$$

$$\left(R_j^{\nu}\right)^3 - \left(R_{j-1}^{\nu}\right)^3 \sim \frac{\partial R^3}{\partial s^3}\Bigg|^{s=(j-1/2)\,ds_\nu,\ t=t_{\nu}} \left[(j\,ds_\nu)^3 - ((j-1)\,ds_\nu)^3\right].$$

We next explain the determination of $dt_{\nu+1/2}$. It has been shown by Courant, Friedrichs and Levy that the ratio of the interval in t to that in s must be less than the reciprocal of the velocity of sound in order to ensure that the calculation will be stable, i.e. that rounding errors will not increase rapidly. In the present case this means that

$$\left(\frac{dt}{ds}\right)^2 < \frac{\rho}{\gamma p}\left(\frac{\partial R}{\partial s}\right)^2 = c^{-2}$$

It is with the help of this formula that we calculate $dt_{\nu+1/2}$. We evaluate the right-hand side of this inequality at the shock, i.e. at $s = s_\nu$. There $R = r$ and hence relation (3) tells us that

$$\frac{\partial R}{\partial s} = \frac{\rho_0}{R^2 \rho};$$

Further the first of relations (6) gives us the value of ρ/ρ_0 at the shock. Thus

$$\frac{\rho}{\gamma p}\left(\frac{\partial R}{\partial s}\right)^2 = \frac{(\gamma-1)p + (\gamma+1)p_0}{(\gamma+1)p + (\gamma-1)p_0} \cdot \frac{1}{\gamma p} \cdot \frac{1}{2s}.$$

Now we define $dt_{\nu+1/2}$ with the help of the relation

$$dt_{\nu+1/2} = \Gamma \sqrt{\frac{1}{\gamma p}} \sqrt{\frac{(\gamma-1)p + (\gamma+1)p_0}{(\gamma+1)p + (\gamma-1)p_0}} \cdot \frac{1}{R_\nu}\, ds_\nu,$$

We describe now the modus procedendi of the calculation. To this end we first describe shock treatment. This is carried out by a

method due to von Neumann. Let us assume that we have obtained the values of R and p at time t_n together with the shock position $R_{n,0}$ and the shock velocities $D_{n-1/2}$, $D_{n-3/2}$. Given these data we make a first estimate to $D_{n+1/2}$ by a linear extrapolation and obtain a value $\tilde{D}^{(k)}$; we define $\tilde{D}^{(0)}$ to be $D_{n-1/2}$. With the help of this $\tilde{D}$ we find at $t_{n+1/2}$ the mass-velocity $\tilde{u}$, the shock pressure $\tilde{p}$ and the shock density $\tilde{\rho}$. We are also able to find $\bar{R}_{99}^{n+1}$, $\bar{R}_{98}^{n+1}$ from equation (9) and $\bar{p}_{98\,1/2}^{\,n+1}$, $\bar{p}_{97\,1/2}^{\,n+1}$ from (10). From $\bar{R}_{99}^{n+1} - \bar{R}_{99}^{n}$, $\bar{R}_{98}^{n+1} - \bar{R}_{98}^{n}$ we can estimate the mass-velocities $\bar{u}_{99}^{n+1/2}$, $\bar{u}_{98}^{n+1/2}$ and from these together with $\tilde{u}$ we can estimate $\bar{u}_{100}^{n+1/2}$. This latter quantity may now be used to obtain a value for $\bar{R}_{100}^{n+1}$ and hence for $\bar{p}_{99\,1/2}^{\,n+1/2}$. Given $\bar{p}_{97\,1/2}^{\,n+1}$, $\bar{p}_{98\,1/2}^{\,n+1}$, $\bar{p}_{99\,1/2}^{\,n+1}$, we can extrapolate a value $p_{(n+1/2)}$ for $\bar{p}^{\,n+1}$ at $R_n + \tilde{D}\,dt_{n+1/2}/2$. Note that this point is in the s-plane directly above the point on the shock at which $\tilde{D}$ was estimated. We have values $\tilde{p}$ for p and $\tilde{\rho}$ for ρ at this point and can therefore calculate a value $p_{[n+1/2]}$ for $\bar{p}^{\,n+1}$ at this s with the help of the adiabatic relation. We now have two different estimates $p_{(n+1/2)}$ and $p_{[n+1/2]}$ for the value of p at this point, and thus an error indicator $\Delta = p_{[n+1/2]} - p_{(n+1/2)}$. We can then use these to obtain a new, improved value for $\tilde{D}$ in this fashion: If $\tilde{D}^{(k)}$, $\tilde{D}^{(k-1)}$ are known estimates for $D_{n+1/2}$ and if $\Delta^{(k)}$, $\Delta^{(k-1)}$ are the corresponding values of Δ, then

$$D^{(k+1)} = D^{(k)} + \frac{\Delta^{(k)}}{\Delta^{(k-1)} - \Delta^{(k)}} \cdot \left(D^{(k)} - D^{(k-1)} \right)$$

is a presumably improved value for $D_{n+1/2}$

This iteration is continued until $D_{n+1/2}$ is obtained to desired precision. In terms of this quantity one then computes

$$\mathcal{R}_{n+1} = \mathcal{R}_n + D_{n+1/2}\, dt_{n+1/2},$$

and next $\bar{R}_j^{n+1}$, $\bar{\rho}_{j-1/2}^{n+1}$ ($j = 100, 99, \ldots, 1$). We note that $R(0,t) \equiv 0$. It remains only to discuss the method whereby we go from the barred to the unbarred quantities. We do this by a simple quadratic interpolation. I.e.

$$f_j^{n+1} = f(j\,ds_{n+1}, t_{n+1}) = f(j\,ds_n + j(ds_{n+1} - ds_n), t_{n+1}) \sim$$
$$u\bar{f}_j^{n+1} + \frac{j\nu}{2}\left(\bar{f}_{j+1}^{n+1} - \bar{f}_{j-1}^{n+1}\right) - \frac{j^2\nu^2}{2}\left(\bar{f}_{j+1}^{n+1} - 2\bar{f}_j^{n+1} + \bar{f}_{j-1}^{n+1}\right),$$

where

$$\nu = \mu(2+\mu), \qquad \mu = D_{n+1/2}\, dt_{n+1/2} / \mathcal{R}_n$$

It is the intention to start the calculation with values of R , ρ prescribed by von Neumann's exact theory for the infinitely strong shock. These values will be taken as an approximation to the situation that obtains when the ratio of the shock pressure to that of the undisturbed gas is 100.

Von Neumann showed that these values could be obtained simply by integrating a first order ordinary differential equation. If

$$z = r/\mathcal{R}, \qquad f(z) = R/\mathcal{R}, \qquad c = 2\left((\gamma-1)/(\gamma+1)\right)^{\gamma}/(\gamma+1)$$

then f satisfies

$$\frac{c}{\gamma-1}\cdot z^{2(\gamma-1)}\cdot f(z)^{-2(\gamma-1)}\cdot f_2^{-(\gamma-1)} + \tfrac{1}{2} z^3 \left(f(z) - z f(z)\right)^2 =$$

$$= c\, z^{-3+2\gamma}\cdot f(z)^{-2\gamma}\cdot f_2(z)^{-\gamma}\left(f(z) - z f(z)\right).$$

He also showed that this equation can be expressed in the following parametric form

$$\frac{df}{dz} = -\frac{f}{z}\left(\frac{1-\gamma}{\vartheta+\gamma}\right), \quad \frac{d\vartheta}{dz} = \frac{\vartheta(\vartheta+1)(3(z-\gamma)\vartheta+1+2\gamma)}{z(\gamma\vartheta^2+2\vartheta+\gamma)},$$

and then that

$$\frac{dp}{dz} = \frac{p}{\vartheta+\gamma}\left(3(\gamma-1)\vartheta + \gamma z \frac{d\vartheta}{dz}\right)\cdot\frac{1}{z}$$

We wish, however, to find the values of $R = \mathcal{R} f$, p as functions of our variable $s = \lambda^2/2$. To this end we introduce the new variable

$$u = z^2$$

Then

$$\frac{df}{du} = -\frac{f}{2u}\left(\frac{1-\gamma}{\vartheta+\gamma}\right), \quad \frac{d\vartheta}{du} = \frac{\vartheta(\vartheta+1)(3(2-\gamma)\vartheta+1+2\gamma)}{2u(\gamma\vartheta^2+2\vartheta+\gamma)}$$

$$\frac{dp}{du} = \frac{p}{\vartheta+\gamma}\left(3(\gamma-1)\vartheta + 2\gamma u \frac{d\vartheta}{du}\right)\cdot\frac{1}{2u}$$

Finally, since we are interested in R , p only for one value of t we may express these equations as

$$\frac{dR}{ds} = -\frac{R}{2s}\left(\frac{1-\gamma}{\vartheta+\gamma}\right), \quad \frac{d\vartheta}{ds} = \frac{\vartheta(\vartheta+1)(3(2-\gamma)\vartheta+1+2\gamma}{2s(\gamma\vartheta^2+2\vartheta+\gamma)}$$

$$\frac{dp}{ds} = \frac{p}{\vartheta+\gamma}\left(3(\gamma-1)\vartheta + 2\gamma s \frac{d\vartheta}{ds}\right)\cdot\frac{1}{2s},$$

the integration to be carried out from $s = \frac{1}{2}\mathcal{R}^2$ to $s = 0$ with the starting values at $s = \frac{1}{2}\mathcal{R}^2$ of $R = \lambda = \mathcal{R}$, $p = 100$.

The calculation is to proceed until the shock pressure is about 1/10.

2.4 Random numbers.

In the numerical investigation of several types of physical and mathematical problems it is necessary to have available in a code, a sub-routine which will produce, upon the demand of the main routine in

which it is embedded, a number such that long sequences of these numbers have many of the statistical properties usually ascribed to a table of random numbers.

As examples one has the class of physical problems in which it is necessary to numerically simulate the movement of a particle in a random path, and as another instance there is the differential equation solution scheme which is commonly called the "Monte Carlo" method.

We first discuss the numerical scheme of "random-number" production which was the central object of our investigations.

The calculation starts with some given forty-digit binary number a_1. The sequence $a_1, a_2, \ldots, a_m$ is then calculated inductively by the procedure now described: assume we are at the i-th stage (i.e. $a_1, \ldots, a_i$ have been computed) then a_{i+1} is obtained from a_i by squaring a_i (a_i^2 is thus an eighty digit number) and extracting the middle forty digits of a_i^2. This new number (namely the middle forty digits of a_i^2) is a_{i+1}. Thus if

$$a_i = \alpha_i^1 \alpha_i^2 \cdots \alpha_i^{40}$$

then

$$a_i^2 = \beta_{a_i^1}^1 \beta_{a_i^2}^2 \cdots \beta_{a_i^{80}}^{80}$$

and

$$a_{i+1} = \beta_{a_i^{21}}^{21} \beta_{a_i^{22}}^{22} \cdots \beta_{a_i^{60}}^{60} = \alpha_{i+1}^1 \alpha_{i+1}^2 \cdots \alpha_{i+1}^{40}$$

The code does the following: starting with a_1, it computes the sequence $a_1, a_2, \ldots, a_{2^{15}}$ then it recomputes this sequence checking whether for any i $(= 1, 2, \ldots, 2^{15})$, a_i is equal to $a_{2^{15}}$, if there is such an i, (and there need by no means be one since there are 2^{40} distinct forty-digit binary numbers) we call it i^* and proceed

to look for another value of i (which we call $\bar{i}$) such that $a_{i^*} = a_{\bar{i}} = a_{2^{15}}$ $(\bar{i} > i^*)$. This determines the length of the repeating block in the sequence $a_1, a_2, \ldots, a_{2^{15}}$; namely $\bar{i} - i^*$ and moreover guarantees that $a_1, a_2, \ldots, a_K$ (where $K = \max\{i^*, (\bar{i} - i^*)\}$) are distinct. This sequence ($a_1, a_2, \ldots, a_K$) is the object of interest and the code gathers the following information about it.

Each $a_j = \alpha_j^{1} \alpha_j^{2} \ldots \alpha_j^{40}$ is broken up into four decades $d_K = \alpha_j^{10K+1} \alpha_j^{10K+2} \ldots \alpha_j^{10(K+1)}$, $K = 0, 1, 2, 3$ and a count is made of the number of d_K 's having the value $d = 0, 1, 2, 3, \ldots, 2^{10} - 1$. Call the d -count N_d , and let $n_d = 2^{-4} N_d$ and $\bar{n}_d = \min(2^7, n_d)$. (Clearly for $K = 2^{15}$, n_d has the expected value 2^3 and the quantity $\bar{n}_d$ is introduced simply on the grounds that $\bar{n}_d$ should usually equal n_d and $\bar{n}_d$ definitely has the bound 2^7 which simplifies its handling.) The code now makes a count of the $\bar{n}_d$'s.

The procedure in the running was to get a starting number a_1 which produced a distinct sequence $a_1, a_2, \ldots, a_{2^{15}}$, then gather the above information and then start again using $a_{2^{15}}$ as a_1 in the next run. In this way a distinct sequence of 2^{18} a 's was obtained starting from the number 010101...01. Tables of the step-wise and cumulative $\bar{n}_d$ counts (Figures 1 and 2) are given at the end of this section. The tables seem to exhibit the expected distribution convergence from binomial to normal.

Some statistics gathered by the computing group at the Los Alamos Scientific Laboratory are enclosed (Figure No. 3) as being of interest. The statistics are based on the method of random-number production

described above. A total of 718,726 distinct sequence elements was generated before the repeating block was encountered. The table listings are as follows: The first entry in each column is the number of counters with readings less than ten half-probable error deviations, the second column entry the number between -10 and -9, etc. The last two column entries are respectively the iteration numbers and the integer ε which is related to the χ^2 by

$$\chi^2 = 2^{10} N \varepsilon^2 \cdot 2^{-39}$$

where N is the number of decades. No analysis of these statistics was provided.

2.5 Solid diffusion in fixed beds.

The problem discussed below was done for Dr. J. B. Rosen of the Forrestal Research Center of Princeton University.

The mathematical problem and resulting computation arise in connection with the performance of a fixed bed chemical system such as an ion exchange column or an adsorption column. In such a system a fluid flows at constant velocity through a fixed column packed with solid spherical particles. The fluid contains a concentration C of active material which is transferred reversibly between the fluid and the solid bed particles. The rate at which this transfer takes place is determined by the combined effect of a liquid film at the surface of a particle, characterized by a film resistance parameter ν, and diffusion into the interior of the particles, characterized by a solid diffusion coefficient D.

The normal operation of a fixed bed system consists of feeding an initially empty column with a fluid containing a constant concentration

I^{-4} Freq (Freq of occurrence of values between 0 and 1023)

$\alpha_0 = 55555\ 55555$ code #122

	α_0 to $\alpha_{2^{15}}$	$\alpha_{2^{15}}$ to $\alpha_{2^{16}}$	$\alpha_{2^{16}}$ to $\alpha_{2^{16}+2^{15}}$	$\alpha_{2^{16}+2^{15}}$ to $\alpha_{2^{17}}$	$\alpha_{2^{17}}$ to $\alpha_{2^{17}+2^{15}}$	$\alpha_{2^{17}+2^{15}}$ to $\alpha_{2^{17}+2^{16}}$	$\alpha_{2^{17}+2^{16}}$ to $\alpha_{2^{17}+2^{16}+2^{15}}$	$\alpha_{2^{17}+2^{16}+2^{15}}$ to $\alpha_{2^{18}}$
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	2		2		4	3	2	0
6	66	66	66	75	76	73	70	74
7	436	438	434	452	412	432	434	439
8	433	434	433	393	430	425	428	423
9	83	81	85	99	102	82	81	81
10	4	4	4	4	0	9	8	7
11	0	0	0	0	0	0	1	0

Figure 1.

I^4 Cumulative Freq. (Freq. of occurrence of values between 0 and 1023)

$\alpha_0 = 55555\ 55555$ code # 122

i is index of α

	$i=2^{15}$	$i=2^{16}$	$i=2^{16}+2^{15}$	$i=2^{17}$	$i=2^{17}+2^5$	$i=2^{17}+2^{16}$	$i=2^{17}+2^{16}+2^{15}$	$i=2^8$
0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	2	3	5	6	10	13	15	15
6	66	132	198	273	349	422	492	566
7	436	874	1,308	1,760	2,172	2,604	3,038	3,477
8	433	867	1,300	1,693	2,123	2,548	2,976	3,399
9	83	164	249	348	450	532	613	694
10	4	8	12	16	16	25	33	40
11	0	0	0	0	0	0	1	1

Figure 2.

Figure no. 3.

NO OF HALF-PROB ERROR DEVIATIONS (WITH RESPECT TO MEAN VALUE)										
< -10	0	0	0	0	0	0	0	0	0	0
-10	0	0	0	0	0	0	0	0	0	0
-9	0	0	0	1	1	0	0	0	0	0
-8	0	0	0	0	0	1	1	0	0	1
-7	1	7	3	2	0	0	1	2	3	3
-6	4	7	7	5	8	7	6	8	4	2
-5	15	7	11	22	19	16	13	15	18	15
-4	50	42	50	40	42	48	42	45	51	50
-3	83	105	77	84	83	96	106	92	90	94
-2	158	150	158	150	154	146	144	155	141	135
-1	15	204	180	218	192	186	193	192	194	212
0	184	190	234	190	209	209	202	192	265	199
+1	151	152	147	150	163	158	151	165	140	151
2	2[illegible]	101	100	97	89	94	101	95	97	88
3	[illegible]0	39	41	38	37	36	37	33	40	47
4	20	16	10	21	16	16	18	18	17	12
5	5	8	2	3	9	8	6	9	11	10
6	2	1	4	2	1	3	3	3	0	1
7	0	0	0	1	1	0	0	0	1	1
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
≥ +10	0	0	0	0	0	0	0	0	0	0
RELATIVE	0 to 100,000	100,000 to 200,000	200,000 to 300,000	300,000 to 400,000	400,000 to 500,000	500,000 to 550,000	550,000 to 600,000	600,000 to 650,000	650,000 to 700,000	650,000 to 1,716,726
$(\bar{X}_i - \bar{N}_i)^2$	1,432,644	718,402		363,713	297,008	267,171	245,069	226,829	215,475	210,320
NUMBER										

C_o (influent concentration). The quantity of interest is then the concentration flowing out of the other end of the bed as a function of time (effluent concentration). This type of operation is known as saturation. The reverse process to this, elution, is also of importance. Because of the linearity of the system, the solution of the saturation problem also gives the solution of the elution problem.

The desired mathematical solution gives the normalized effluent $u = c/c_o$ in terms of three dimensionless parameters: ν the film resistance parameter, x the bed length parameter, and y the time parameter. All three parameters are proportional to the diffusion coefficient D. The most convenient way of presenting the values of $u(\nu, x, y)$ is to plot curves of u vs. y/x for a range of values of x. One of the five graphs thus obtained (covering the useful range of parameters) is appended to this report together with the tables of values from which it was drawn (Figure 4). The 280 values of $u(\nu, x, y)$ computed were chosen with this method of presentation in mind. The parameter values chosen were ν/x = 0, .025, .05, 0.1, 0.2; x = 0.2, 0.5, 1, 2, 5, 10, 20, 40; and seven appropriate values of y/x for each of the preceding pairs of values.

The original mathematical formulation of the problem* leads to the following partial integro-differential equation for

$$\frac{\partial c}{\partial x} = -2 \sum_{n=1}^{\infty} \int_0^y \left[\frac{\partial c}{\partial \lambda} + \nu \frac{\partial^2 c}{\partial x \partial \lambda} \right] e^{-\frac{n^2 \pi^2}{4}(\lambda - y)} \, d\lambda$$

Its solution by the Laplace transform gives

---

* J. B. Rosen, J. Chem. Phys., 20, 387-94 (1952).

$$\mu(\nu, x, y) = \frac{1}{2} + \frac{1}{\pi}\,\Psi(\nu, x, y)$$

where

$$\Psi(\nu, x, y) = \int_0^\infty e^{-x H_1(\sqrt{\beta}, \nu)} \sin\left(y\beta - x H_2(\sqrt{\beta}, \nu)\right) \frac{d\beta}{\beta},$$

$$H_1(\lambda, \nu) = \frac{H_{D_1}(\lambda) + \nu\left[H_{D_1}^2(\lambda) + H_{D_2}^2(\lambda)\right]}{\mathcal{D}(\lambda)},$$

$$H_2(\lambda, \nu) = \frac{H_{D_2}(\lambda)}{\mathcal{D}(\lambda)},$$

$$H_{D_1}(\lambda) = \frac{\lambda\left[\sinh 2\lambda + \sin 2\lambda\right]}{\cosh 2\lambda - \cos 2\lambda},$$

$$H_{D_2}(\lambda) = \frac{\lambda\left[\sinh 2\lambda - \sin 2\lambda\right]}{\cosh 2\lambda - \cos 2\lambda},$$

$$\mathcal{D}(\lambda) = \left(1 + \nu H_{D_1}(\lambda)\right)^2 + \left(\nu H_{D_2}(\lambda)\right)^2,$$

and

$$\lambda = \sqrt{\beta}$$

It can be demonstrated that $-\pi/2 \leq \Psi \leq \pi/2$, so that $0 \leq \mu \leq 1$ as it should be on the basis of physical reasoning.

The numerical representation is effected in two basic steps:

1) Replacing the infinite integral $\Psi = \int_0^\infty F(\beta)\, d\beta$ by the finite one $\int_0^{\beta_m} F(\beta)\, d\beta$, and then

2) Replacing $\int_0^{\beta_m} F(\beta)\, d\beta$ by an approximating sum S. Here, of course, $F(\beta)$ is the integrand.

$$\frac{e^{-x H_1(\sqrt{\beta}, \gamma)} \sin(y\beta - x H_2(\sqrt{\beta}, \gamma))}{\beta},$$

and β_m is the finite upper limit of integration.

We thus have

$$\Psi = S + \epsilon_T + \epsilon_S$$

where ϵ_T is the error in truncating the upper limit Ψ and ϵ_S is the summation error. These errors will be discussed and bounds derived for them following the description of the integration technique.

The motivation for the rather complicated numerical integration procedure (described below) employed is as follows. In numerically evaluating a sinusoidal integrand the usual procedures dictate the necessity of maintaining a minimum number of points of evaluation of the integrand per cycle (of the integrand). In the integrand (4) it is clearly seen that (4) oscillates infinitely often as y/β increases. Thus if one were to determine a fixed number of evaluation points so as to guarantee a pre-specified minimum number of points per cycle, the total number of points to β_m would be prohibitively large (e.g. some of the integrals would take over an hour to calculate).

This difficulty is avoided by employing an integration procedure in which the interval of integration is a variable which increases as permitted by decreasing summation errors for increasing β, and in which the integration formula is Weddle's rule (6) at first, but changes

to a series (7) of formulas which are more efficient when $x H_2(\beta^{1/2}, \nu)$ is small relative to y/β . The particulars now follow.

The range of integration is broken up into intervals, each of six finite difference steps of h . For each value of the parameter y the formula (6) is employed (see below) during the range for which $y/\beta \leq 4 x H_2(\sqrt{\beta}, \nu)$, i.e. while the oscillations of (4) are still being influenced heavily by the term $x H_2$. On y/β becoming greater than $4 x H_2$ the formula (7) replaces formula (6) to the conclusion of the integration.

$$\int_{\beta}^{\beta+6h} F(\beta)\, d\beta = \frac{h}{140} \sum_{j=0}^{6} A_j F(\beta + jh)$$

$$\int_{\beta}^{\beta+6h} F(\beta)\, d\beta = \frac{1}{y} \sum_{j=0}^{6} [C_j^p P(\beta + jh) - B_j^p Q(\beta + jh)]$$

where

a) $$P(\beta) = \frac{1}{\beta} e^{-x H_1(\sqrt{\beta}, \nu)} \cdot \sin(x H_2(\sqrt{\beta}, \nu))$$

b) $$Q(\beta) = \frac{1}{\beta} e^{-x H_1(\sqrt{\beta}, \nu)} \cos(x H_2(\sqrt{\beta}, \nu))$$

The A_j are stored constants and the C_j^p and B_j^p are sets of constants which are different for each value of $p = 0, 1, 2, 3, 4, 5$. The role of p and the method for determining h are now given. The following definitions will be needed.

Let K be the integer such that

$$2^K \leq 3/2\ y/x < 2^{K+1}$$

and let K^* be

$$K^* = \begin{cases} 0 & , K \geq 0 \\ K & K < 0 \end{cases}$$

Thus K (and K^*) are determined by the choice of parameters y and x.

Now let m be the greatest integer such that

$$\frac{3H_2(\sqrt{\beta}, \nu)}{2\beta} \leq 2^{-m} \qquad \text{and} \qquad h < \pi/6$$

where h and the formula to be used are determined by the table (10).

		value of h	formula
(10.a)	$m+K^* < 0$	$2^{m+K^*}\ \pi/12y$	(6)
(10.b)	$0 \leq m+K^* < 2$	$\pi/12y$	(6)
(10.c)	$2 \leq m+K^* < 6$	$2^{m+K}\ \pi/12y$	(7)
(10.d)	$6 \leq m+K^* < \infty$	$2^7 \cdot \pi/12y$	(7)

Now, p is defined as $p \begin{cases} 2^{m+K-2} & , 0 \leq m+K-2 \leq 5 \\ 2^5 & , 6 \leq m+K-2 < \infty \end{cases}$

thus the cases (10.c) and (10.d) have $h = p\,\frac{\pi}{3y}$. The procedure is then to determine K, K^*, and m, this determines (using the above scheme) the initial value of h (in the actual computation, the initial value -3 is forced upon m in order to guarantee that the integration starts with formula (6), $P(\beta)$ (7.a) being singular at $\beta = 0$). The integral is then calculated over the interval $(0, 6h)$. After this is done, m is tentatively increased by unity and a new calculated using $m+1$ (thus h is doubled, effectively). It is then checked whether the new h is consistent with the conditions (9) and

4) $\beta/2\gamma h$

is an integer.

If it is so, then the new h is allowed to stand and is used in computing the next contribution (i.e. for $(\beta, \beta+6h)$), if not, m is restored to its previous value and the next step proceeds using the previous h. This process is then continued after each integration interval $(\beta, \beta+6h)$ up to an upper limit β_m. (The value of β_m is motivated by error considerations and will be discussed in that section of this report.)

Formula (7) is derived using the identity

$$F(\beta) = P(\beta) \sin\gamma\beta - Q(\beta) \cos\gamma\beta$$

and takes advantage of the periodic oscillations of the $\sin\gamma\beta$ and $\cos\gamma\beta$ terms to increase the interval $6h$ to include from one up to as many as 32 complete cycles of these oscillations: these correspond to $P = 0, 1, \ldots, 5$. For each value of P, six each of the coefficients B_i^P and C_i^P are used, thus requiring a total of 72 coefficients.

It is tedious but quite easy to show that the scheme described above guarantees that

a) While using formula (6) there are six evaluation points per cycle of the integrand.

b) The transfer from formula (6) to formula (7) occurs at the end of a cycle.

c) P is related to the basic period $(2\pi/\gamma)$ by a power of 2.

d) The interval h is doubled as frequently as possible consistent with the above, and

e) $h < 1/6$, finally

f) The error in Ψ is less than .005 (see below).

We now proceed with the discussion of the error terms ϵ_T and ϵ_S (5). First, we define β_m :

$$\beta_m = \min\{144, \beta \text{ such that } x H_1(\sqrt{\beta}, \nu) \geq 10\}.$$

The desired accuracy was to have Ψ (2) computed with error less than .005, thus the required bounds on ϵ_s and ϵ_T are

$$|\epsilon_s|, |\epsilon_T| < .0025.$$

Now,

$$|\epsilon_T| = \left| \int_0^\infty F(\beta)\, d\beta - \int_0^{\beta_m} F(\beta)\, d\beta \right| = \left| \int_{\beta_m}^\infty F(\beta)\, d\beta \right|$$

A bound for this latter integral was obtained by an asymptotic expansion of $F(\beta)$ (4) in inverse powers of β (for certain cases it was found necessary to actually compute the first term of this expansion as a correction term in order to keep β_m from becoming too large) and it was demonstrated that the choice (12) of β_m kept $\epsilon_T < .0025$. In some cases it was possible to use β_m as small as ~ 1.

Now,

$$|\epsilon_c| = \left| \int_0^{\beta_m} F(\beta)\, d\beta - S \right|$$

and clearly the magnitude of $|\epsilon_s|$ depends on the formula used ((6) or (7)), the values of the higher derivatives of $F(\beta)$ and the size of the step length h (10). The error per $6h$ interval for formula (6) is $< 9/1400\, h^9 F^{(8)}$ and for formula (7) is $< (1/10) h^7 (|P^{(7)}| + |Q^{(7)}|)$, where the quantities $F^{(8)}$, $P^{(7)}$, $Q^{(7)}$ are the evaluations of the

derivative of the order of the superscript at some intermediate point. An estimate of these derivatives together with the requirement $|\varepsilon_s| < .0025$ determine a bound for the maximum step length h (determined by (10)). The values of h actually had the range $.0041 \leq h \leq 3/6$.

Before the problem was run, a great many check-runs were made to determine the correctness and accuracy of the code; principally these consisted in the following: Ψ (2) was calculated with the imposed conditions $H_1(\sqrt{\beta}, \nu) \equiv 0$, $H_2(\sqrt{\beta}, \nu) = 0$, thus yielding $\int_0^{\beta_m} \frac{\sin y\beta}{\beta} d\beta$. The machine-computed values agreed (to well past the desired accuracy) with the W. P. A. tables of this integral. Values of H_{D_1}, H_{D_2}, H_1 and H_2 were computed by hand and compared with the machine results. In addition, three non-trivial values of Ψ were hand computed and checked against the machine. For several values of Ψ the integral was re-run with the step length halved, with good agreement.

The full details of the code are now given in the following diagrams (Figures 1, 2, and 3) and the explanatory remarks below:

EXPLANATORY REMARKS.

Figure 1.

1.A. To achieve maximum flexibility in the use of the code it was decided to make each computation of $\Psi(x, y, \nu)$ independent. A given set (x, y, ν) is punched on a card which the code automatically reads in. $\Psi(x, y, \nu)$ is then computed and its value punched out on a card. The process is then repeated until all the cards with the triples (x, y, ν) punched on them have been

read in and the corresponding Ψ 's computed and punched out.

1.B. The indicator σ is set equal to 1 if formula (6) is to be used in the next step of 6ℓ .

1.C. Similarly σ is set equal to zero if formula (7) is to be used.

1.D. Here the discrimination on σ is made.

Figure 2.

2.A. $$F(\beta) = \frac{e^{-x H(\sqrt{\beta},\nu)} \sin(\gamma\beta - x H_2(\sqrt{\beta},\nu))}{\beta}$$

and a ready computation yields

$$F(0) = \gamma - \tfrac{2}{3}\cdot x$$

2.B. $$A_0 = A_6 \ , \quad F_{\ell-1}^{6} = F_\ell^{0}$$

2.C. The symbol $\langle S \rangle$ means that the subroutine S (Figure 3) is entered and returned from. This subroutine is entered in the usual fashion from any memory position $\varkappa$, it then computes $H_1(\sqrt{\beta},\nu)$, $H_2(\sqrt{\beta},\nu)$, and $e^{-x H_1(\sqrt{\beta},\nu)}$ and returns the control to memory position $\varkappa+1$

2.D. $\langle \sin \rangle$ is a subroutine which is entered at memory position $\varkappa$, computes the sine of the angle stored at memory position $\varkappa+1$ (by the multiplicative form of the power series for sin) and returns the control to memory position $\varkappa+2$).

2.E. The indicator ρ is defined in Figure 3.

Figure 3.

3.A. For $\beta < .01$ the asymptotic expressions

$$H_{D1} = \frac{4}{45}\beta^2 \ , \quad H_{D2} = \tfrac{2}{3}\cdot\beta$$

are accurate to six decimals.

3.B. $\langle e^{-\xi} \rangle$ is a subroutine which is entered at α , computes where $e^{-\xi}$ is the quantity stored at $\alpha+1$ (by multiplicative power series expansion) and returns the control to $\alpha+2$.

3.C. For $\beta > 64$ the asymptotic formulas $H_{D1} = \lambda - 1$, $H_{D2} = \lambda$ are accurate to six decimals.

3.D. For $\nu = 0$, $H_1 = H_{D1}$ and $H_2 = H_{D2}$.

3.E. The indicator ρ is set equal to 1 if $\alpha H_1 \geq 10$ and equal to zero if $\alpha H_1 < 10$.

2.6 Bionumeric evolution.

Extensive experiments in a numerical analogue of genetic and evolutionary processes were run for and by Dr. Nils Aall Barricelli of the Mathematics Institute, University of Oslo.

Dr. Barricelli is interested in bionumeric evolution of numerical organisms composed of numerical genes. By this is meant the experimental study of a class of logical-arithmetical systems constructed to suggestively simulate the corresponding biological system.

We give an introductory description of the model studied. (See his earlier report, "Numerical models of evolutionary organisms".) The universe is a linear array of N locations a , studied during successive generations g . Each content $X_{a,g}$, of location a at generation g , is either 0 or a numerical gene, a small positive or negative integer.

The contents of the various locations at generation g determine those at generation $g+1$ by reproduction and mutation rules. By the reproduction rules, a gene $X = X_{a,g}$ in location a would be reproduced in one or more locations of the next generation: first, at

FIGURE NO. 1

DETERMINATION OF h AND FORMULA DECISION

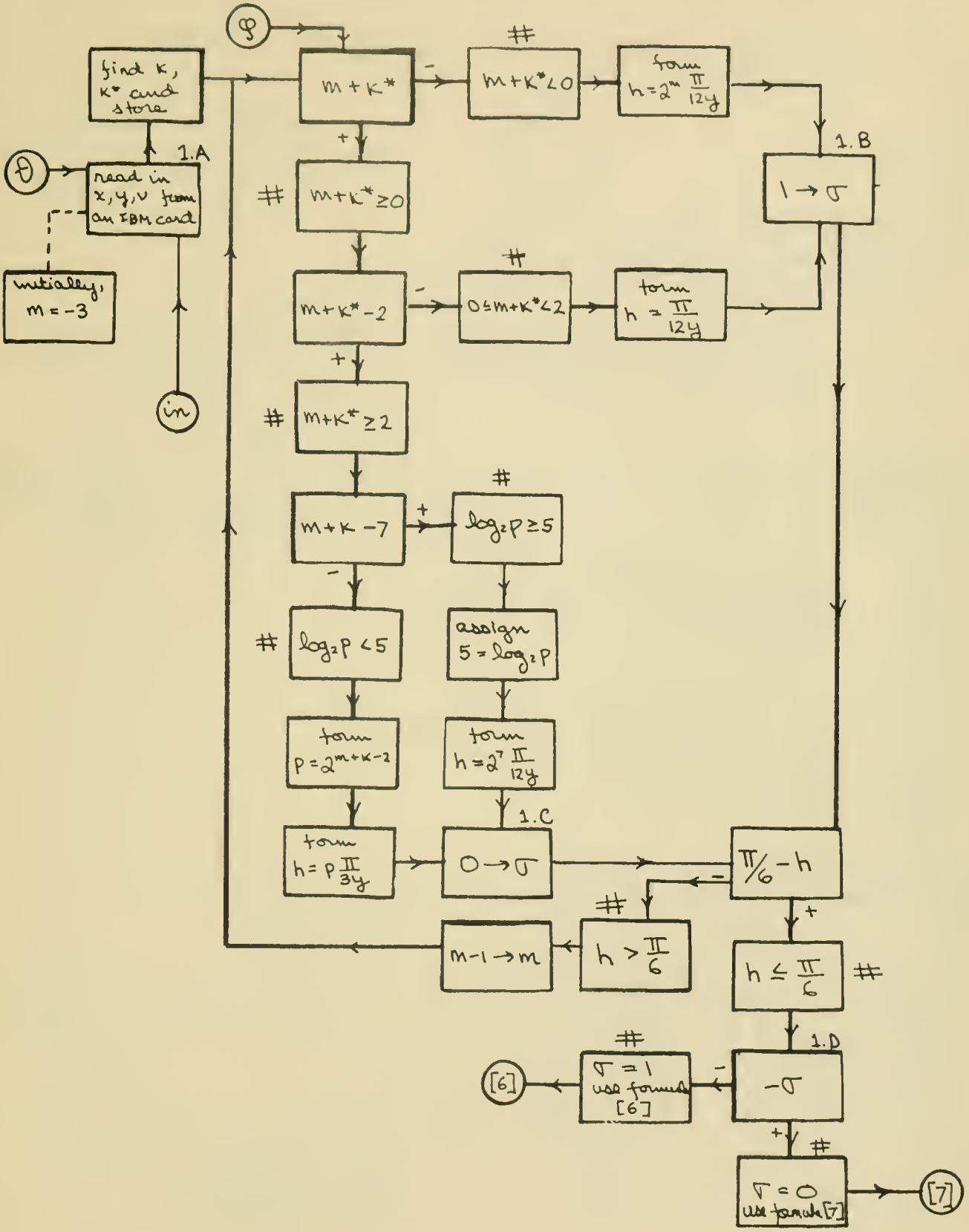
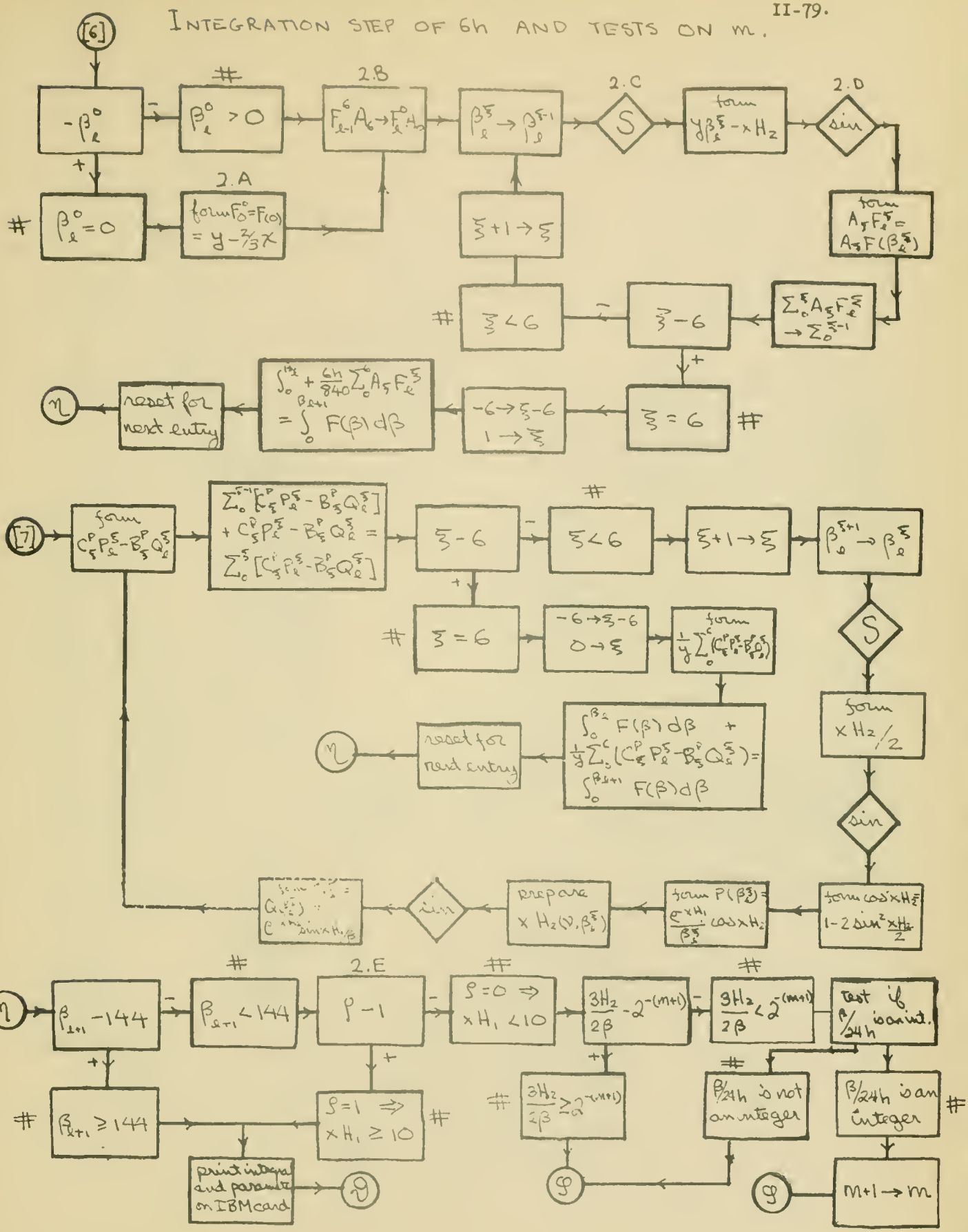


FIGURE N°. 2.

INTEGRATION STEP OF 6h AND TESTS ON m.



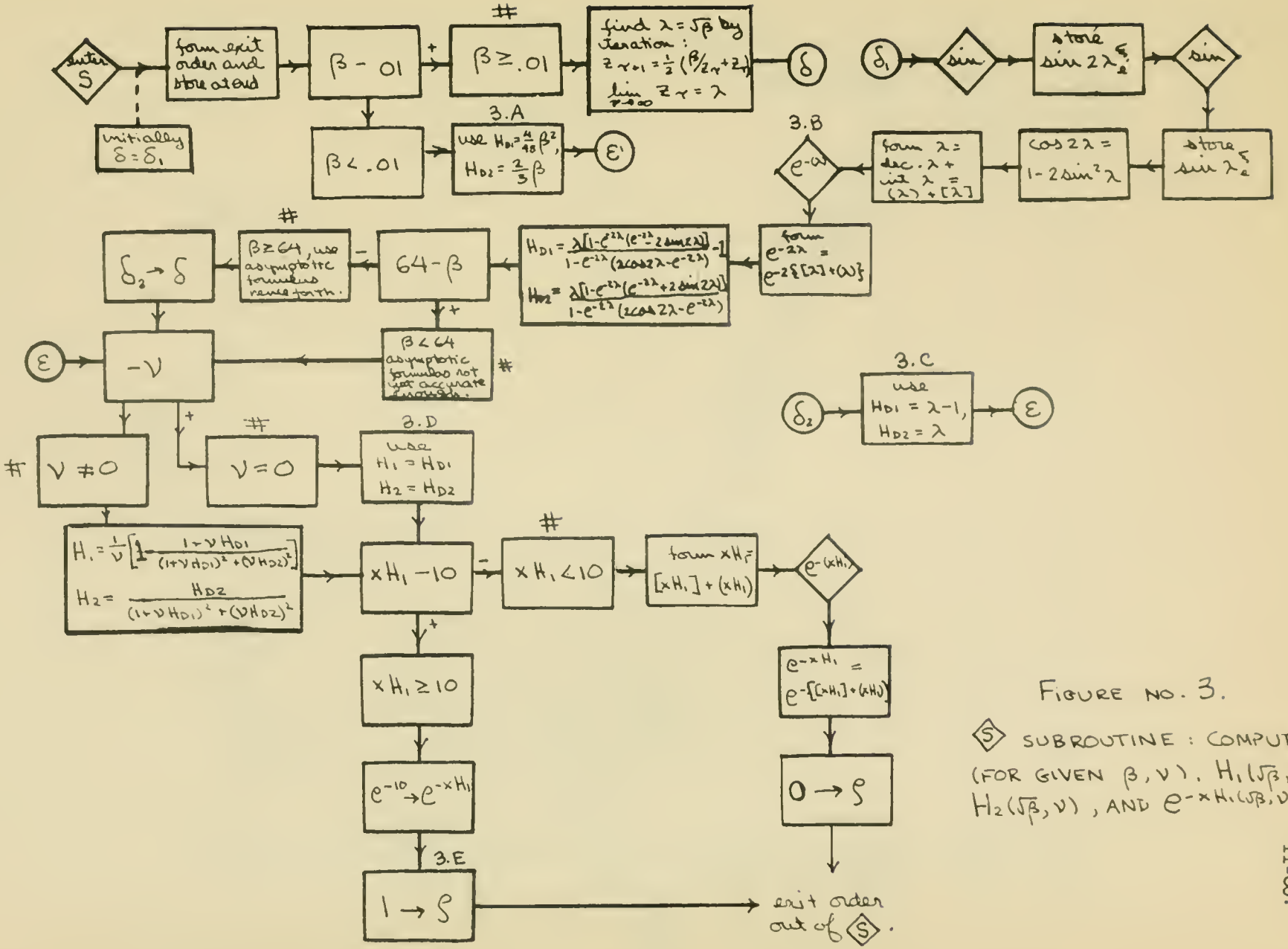


FIGURE NO. 3.

(S) SUBROUTINE : COMPUTES (FOR GIVEN β, ν), $H_1(\sqrt{\beta}, \nu)$, $H_2(\sqrt{\beta}, \nu)$, AND $e^{-x H_1(\sqrt{\beta}, \nu)}$.

II-80.

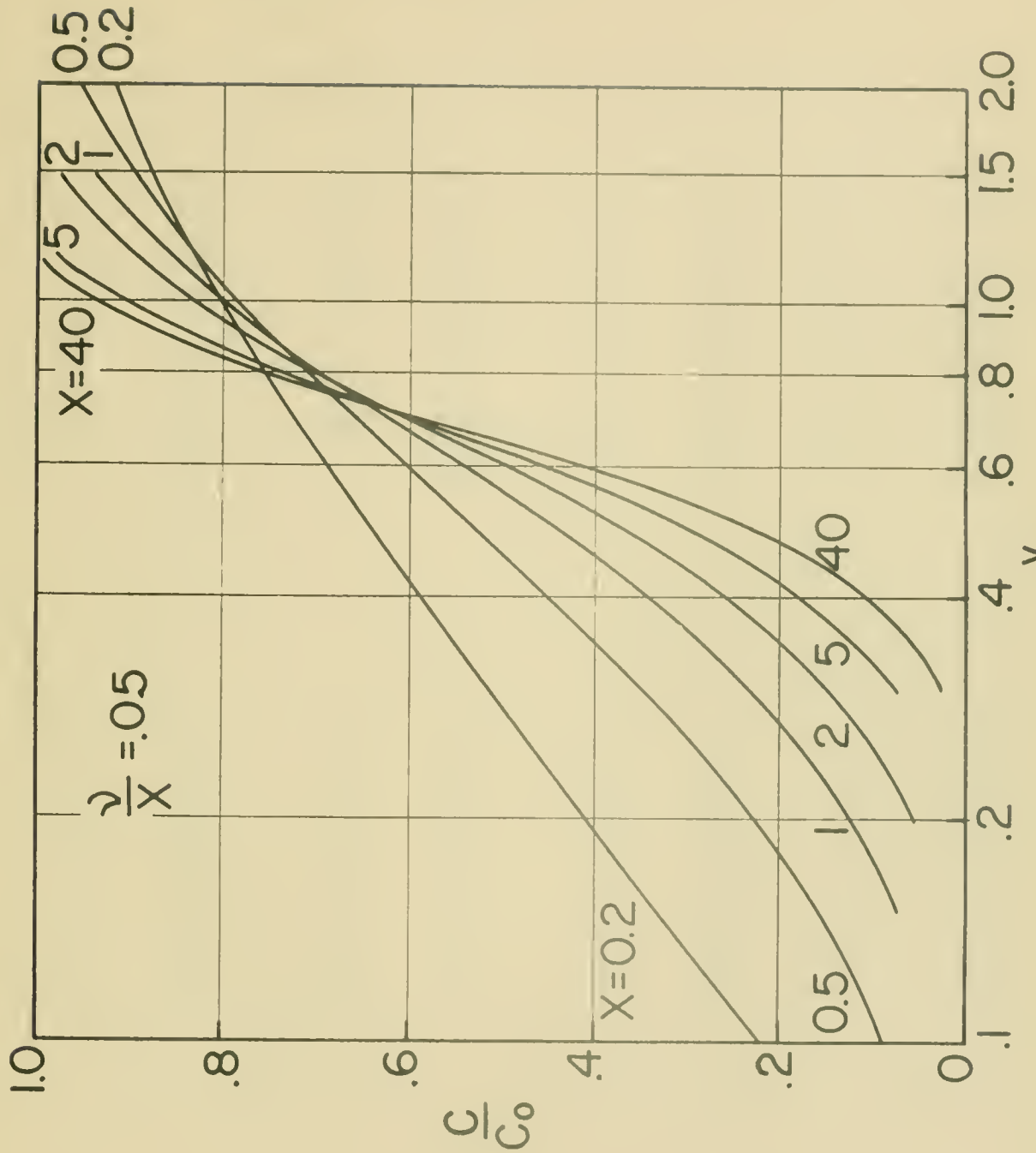

0.5
0.2
2
1
5
X=40
$\frac{\lambda}{X}$=.05
1.0
.8
.6
.4
.2
0
.1
.2
.4
.6
.8
1.0
1.5
2.0
$\frac{y}{X}$
C/C_0
X=0.2
0.5
1
2
5
40

FIG. 4

Tabulation of Computed Values

of Effluent Concentration =

= .05

= 0.2		= 0.5		= 1.0		= 2.0	
.10	.219	.10	.088	.15	.074	.20	.059
.1475	.327	.20	.229	.35	.290	.40	.255
.20	.411	.35	.403	.45	.397	.50	.375
.30	.511	.47	.508	.57	.512	.62	.516
.50	.643	.70	.654	.70	.619	.70	.600
1.00	.798	1.00	.780	.90	.747	.90	.770
2.00	.914	2.00	.954	1.50	.938	1.50	.971

= 5.0		= 10.0		= 20.0		= 40.0	
.30	.075	.30	.049	.30	.036	.30	.030
.45	.236	.45	.198	.45	.175	.45	.163
.55	.377	.55	.349	.55	.331	.55	.321
.65	.523	.65	.512	.65	.507	.65	.503
.75	.655	.75	.662	.75	.669	.75	.673
.85	.763	.85	.782	.85	.796	.85	.804
1.20	.954	1.20	.971	1.20	.979	1.20	.983

location $b_0 = a + X$; and also, if at location b_i , then also at location $b_{i+1} = a + X_{b_i, g}$, provided $X_{b_i, g} \neq 0$. However, whenever two or more genes would "collide" in a location a' under this rule, one of various mutation norms is invoked to define a single $X_{a', g+1}$ (or 0), depending on the colliding genes and on some of the $X_{a'', g}$ for a'' near a' .

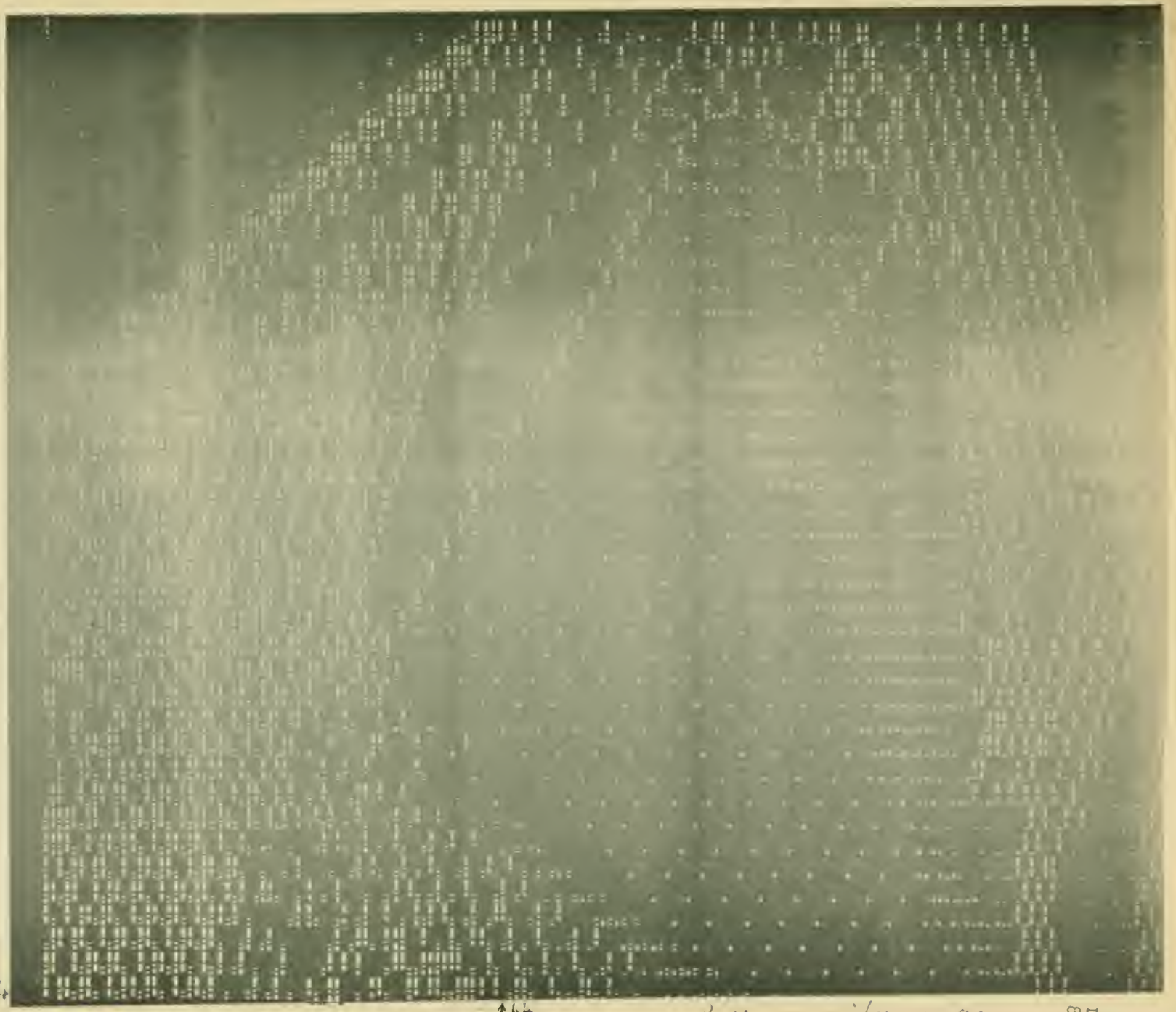
Under such rules Dr. Barricelli showed, by hand methods, the formation and growth of numerical organisms -- linear sequences of genes, which reproduce themselves from one generation to the next -- and further biological analogies such as parasites.

He then turned to the Institute for Advanced Study computer in order to perform more extensive experiments, with varied mutation norms, a larger universe, and especially, many more generations.

As coded for the computer, the universe was cyclic with 512 generations, and each gene X , restricted so that $|X| < 40$, required eight binary digits so that five generations of a location could be packed into a single 40-binary-digit storage location.

The code was written so that various mutation norms could be employed in selected regions of the universe. Special attention was paid to coding for maximum speed of operation, and for the convenient re-use of output data as input after interrupted operation.

The output cards, punched with the contents of half the memory, when abutted top-to-bottom, present five generations of the 512 locations, in proper array. Such arrays were reproduced photographically and further assembled. A small sample of the results is reproduced in the attached figure. The binary representation of the genes proved


485
524
0
164

visually convenient, as well as more economical of output cards and machine time than decimal output. For further economy in the considerable card output, only five out of each 100 generations were recorded during reconnaissance. Interesting phenomena were then reinvestigated in more detail.

Dr. Barricelli will present his results in detail elsewhere. In the present report we wish only to indicate the trend of ideas, and for this purpose quote some passages selected from his preliminary report, interspersed with summaries of the omitted material.

Experiments in bionumeric evolution executed by the electronic computer at Princeton, N. J.

Aims of the experiments.

The aims of the experiments are of two kinds:

1) To find analogies or, possibly, essential discrepancies between bionumerical and biological pehnomena.

2) To observe how the evolution of numeric organisms takes place by hereditary changes and selection and to verify whether some of the organisms are able to speed up their evolution by gene replacements or by acquiring new genes or by any other primitive form of sexual reproduction.

The latter aim can only be reached if we are able to keep one or more species alive for a large number of generations under conditions producing hereditary changes and evolution in the species. But we must avoid producing such conditions by changing the character of the experiment after the experiment has started.

Methods.

1. Employed norms: The mutation and reproduction rules (norms) employed are all of the kind which in the preliminary communication "Numerical models of evolutionary organisms", are called shift norms. The laws of reproduction are the same in all these norms. The only distinction among them concerns the law of mutation.

There follows a description of the nine mutation norms (designated either by a description such as "addition norm" or by an arbitrary name such as "blue norm"), with a summary of their observed evolutionary properties.

2. Size and inputs: In the experiments just executed the number N of cells used for the recording of genes has been 512 numbered from 0 to 511. This is the size of our gene universe. All rules are applied regarding position 0 as identical with position 512. That means the gene universe is cyclic. The input, used as generation 0, have been random numbers obtained by different methods using a set of playing cards.

. . . . .

Results.

1. Experiments by single norms. Most norms have been tested alone -- as single norm in the whole gene universe -- some of them (red, yellow, green and purple modified) by the computer, some others (addition, exclusion and blue norm) were tested before.

. . . . .

All experiments have shown that the use of a single norm in the whole gene universe leads -- in most cases in less than 500 generations -- to uniform conditions in the whole universe, for instance, by progressive disorganization or by a single species expanding to the whole universe.

Where uniform conditions are established every further evolution stops.

2. The combination of different norms. A way to prevent uniformity is to use different norms in the different parts of the gene universe, each of them favoring a different kind of organization or disorganization.

Several experiments with combinations of different norms have been performed. . . . . .

These experiments are then described, and the detailed outputs are reproduced.

3. Some resulting organisms and their properties. One of the aims of this research has been to find possible analogies

between bionumerical and biological phenomena. This can be done by studying the properties of the resulting numerical organisms.

In this and the succeeding two sections, a few of the resulting organisms are discussed as to their properties, especially as to being independent or dependent ("parasites").

6. Vitality maxima. It may be expected that some combinations of genes will produce organisms which are stronger and more fitted to the conditions prevailing in the gene universe than others. In organisms with few genes, this circumstance may easily lead to relative maxima of fitness; that means combinations of genes in which it is impossible to change only one gene without getting weaker organisms. . . . . .

The evolution of an organism may for a long period of time stop in a relative maximum of fitness. But a change in the conditions of the universe, e.g. in the kind of concurrent organisms may sooner or later level the maximum making further evolution possible.

7. Observed evolution processes. Before the Princeton experiments executed by the electronic computer, some experiments with numerical organisms were executed manually. Many of the bionumeric phenomena hitherto described were already observed in these experiments (cf. "Numerical models of evolutionary organisms"). But no evolution process had been observed in these preliminary experiments.

The most important questions to be decided by the Princeton experiments were therefore:

I. Whether or not the numerical organisms are able to undergo some evolutionary processes by hereditary changes and selection.

II. If evolution takes place, what kind of hereditary changes are working in the numerical organisms in order to make evolution possible?

Some short evolutionary processes have been observed during the Princeton experiments. In most of the evolutionary processes observed only one or two hereditary changes have been able to expand; but in some cases even four or five changes have had the time to expand before the species was destroyed by concurrents or parasites or by some kind of disorganization process.

In no case has the evolution led to a degree of fitness which could make the species safe from complete destruction and insure an unlimited evolution process like that which has taken place in the earth and led to higher and higher organisms.

The Princeton experiments have, however, shown that evolution is possible in the numerical organisms and has given some information concerning the kind of hereditary changes which make evolution possible.

It will be one of the most important aims of the next bionumerical experiments to find the way to start an unlimited evolution.

There follows a description of some of the evolutionary processes observed in the numerical organisms.

When a relative maximum of fitness is reached, the evolution process may stop for a long time.

. . . . .

8. The hereditary changes and the transfer of hereditary characters in numeric organisms. In the higher organisms we are in the habit of finding hereditary differences between progeny and parent organisms only as a consequence of crossing or mutation. The crossing process -- sexual reproduction -- is highly specialized. The cases in which hereditary changes may be transferred by virus-molecules able to leave a cell and invade another cell are rare and may be considered as exceptional.

In bacteria and lower organisms the situation is completely different. Mutations produced, for example, by irradiation in bacteria are normally transferred to the bacterial genes by surrounding molecules affected by the radiation. The transfer of hereditary characters from bacterium to bacterium by virus-like molecules is very frequent. Cellular fusion is never observed in bacteria and it is likely that all hereditary interchanges between bacteria are mediated by virus-like particles of different size and with different tasks.

The surrounding material apparently plays an important part in hereditary changes and transfer of hereditary matter in the most primitive organisms.

We find a similar situation in the numeric organisms. The hereditary changes and the transfer of hereditary characters from organisms to organism are prevalently produced or

mediated by surrounding numerical genes. There are also, however, many cases in which a mutation is not transferred by external genes but appears in the organism, for instance, by collision between two different genes -- when the purple modified norm is used -- or by loss of a gene, or by re-arrangement of the existing genes.

The report closes with detailed discussions of the observed mutations (classified as "internal" and "carried") and of crossings between organisms (classified as "transferred gene (induced mutation)", "regular cross" and "transferring cross"). These two types of hereditary changes are compared in a transitional discussion:

All the mutations here adduced have been able to expand and are most likely useful or in any case not injurious. The reason is that we have chosen our examples from the mutations which have played a part in some evolution process. Injurious mutations rapdily disappear and cannot aid evolution. But is is ture also for numeric organisms that most mutations are injurious. Every change in a numeric organism may be considered as a mutation; but only a few of them show the ability to expand. Also in numeric organisms therefore it will be convenient to get hereditary changes which in other organisms of the same species already have shown the ability to expand.

We cannot expect to find a specialized crossing mechanism or sexual reproduction in the primitive numeric organisms which hitherto appeared. But the ability of the numeric organisms to receive external genes, able to enter and to reproduce between their genes, makes it possible to transfer hereditary characters from one organism to another and to obtain crossing results.

As may be expected, the crossing and gene transfer between numeric organisms play an important part in the evolution. The majority of the new variants which have shown the ability to expand are a result of crossing-phenomena and not of mutations, although mutations (especially injurious mutations) have been much more frequent than hereditary changes by crossing in the experiments performed.

2.7 Miscellaneous codes.

For efficient operation of the computing machine it is

necessary to have in convenient form several "service" codes. These perform such operations as the conversion of addresses and symbolic orders to their binary forms as used by the computer and the arrangements of decimal results on output cards so that these may be used directly on an IBM tabulator to give special arrangements of the data on the printed page. The two most important arrangements used were a tabulation on a rectangular grid and the production of a graph. The preparation of the former arrangement is given below.

The graphical output code converts numbers in the computer into suitably punched cards so that the numbers can be plotted as absicissas by an IBM tabulator, thus providing points which may be joined to form a graph of the data.

The code has four parameters: integers N , A and machine numbers C , m . For the contents x_i of each of the N storage locations starting at A , the routine computes $y_i = (x_i - c)/m$, rounded to the nearest tenth (where $-1000 \leq y_i < 1000$), and punches a card.

The punches are such that the tabulator can be wired to print, on successive lines, not only y_i in conventional fashion, but also an entry whose horizontal location (column of the tabulator) depends on y_i . To achieve ultimate resolution finer than that of the solumn spacing, the tenths digit of y_i is made the entry, while the integral part determines the column. It is thus easy, if desired, to interpolate by hand on the tabulator sheet to a tenth of a column, at the location of each entry. Because of peculiarities of the tabulator, a tenths digit 0 was made to print as 1 in the graph, with the correct 0 indicated

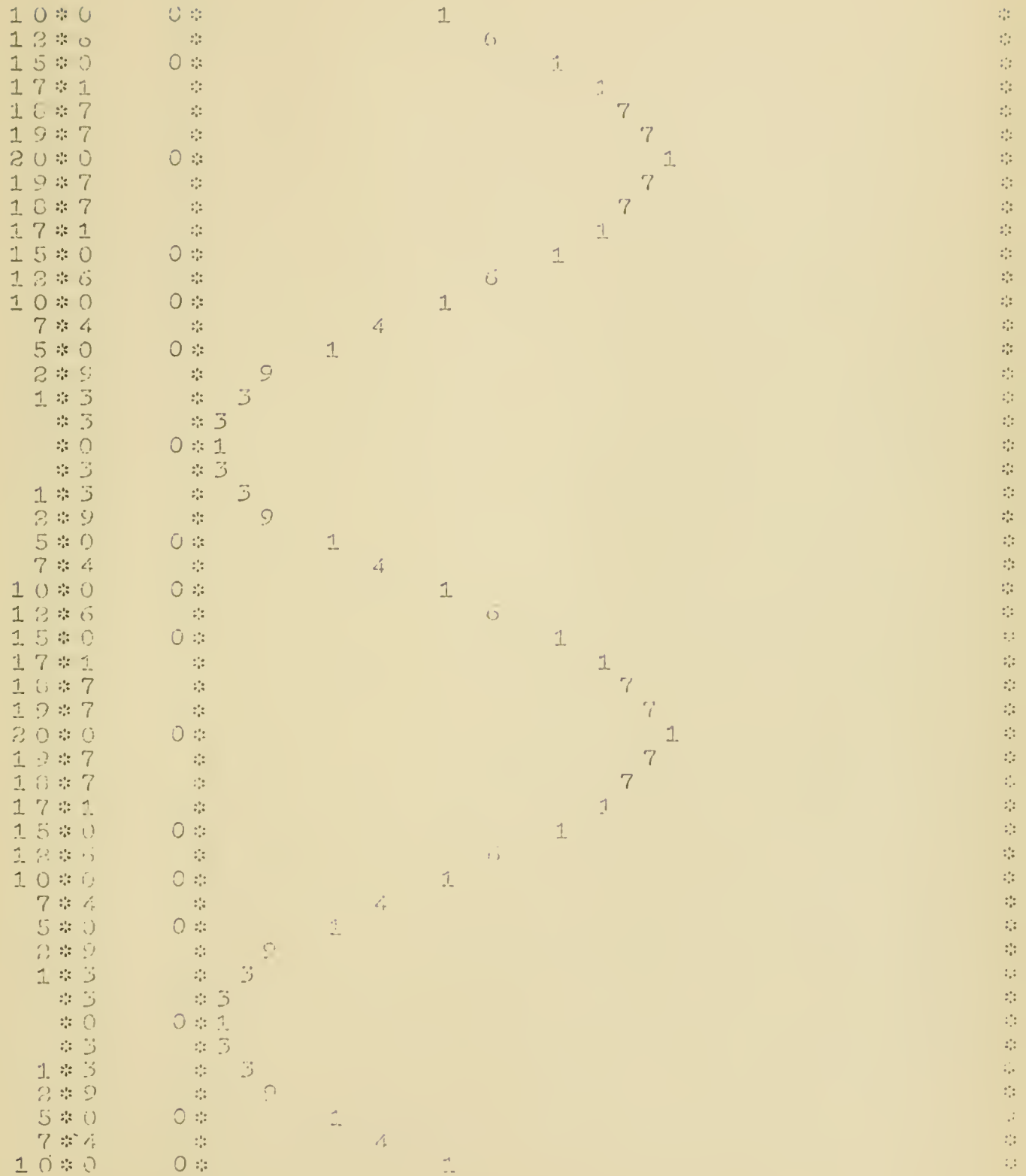
at the side.

The display is 36 columns wide, corresponding to the range $0 \leq y_i < 36$. If y_i is outside this range, the routine translates the entry right or left by a multiple of 36 columns to bring it within range (without affecting the conventional numerical display of y_i).

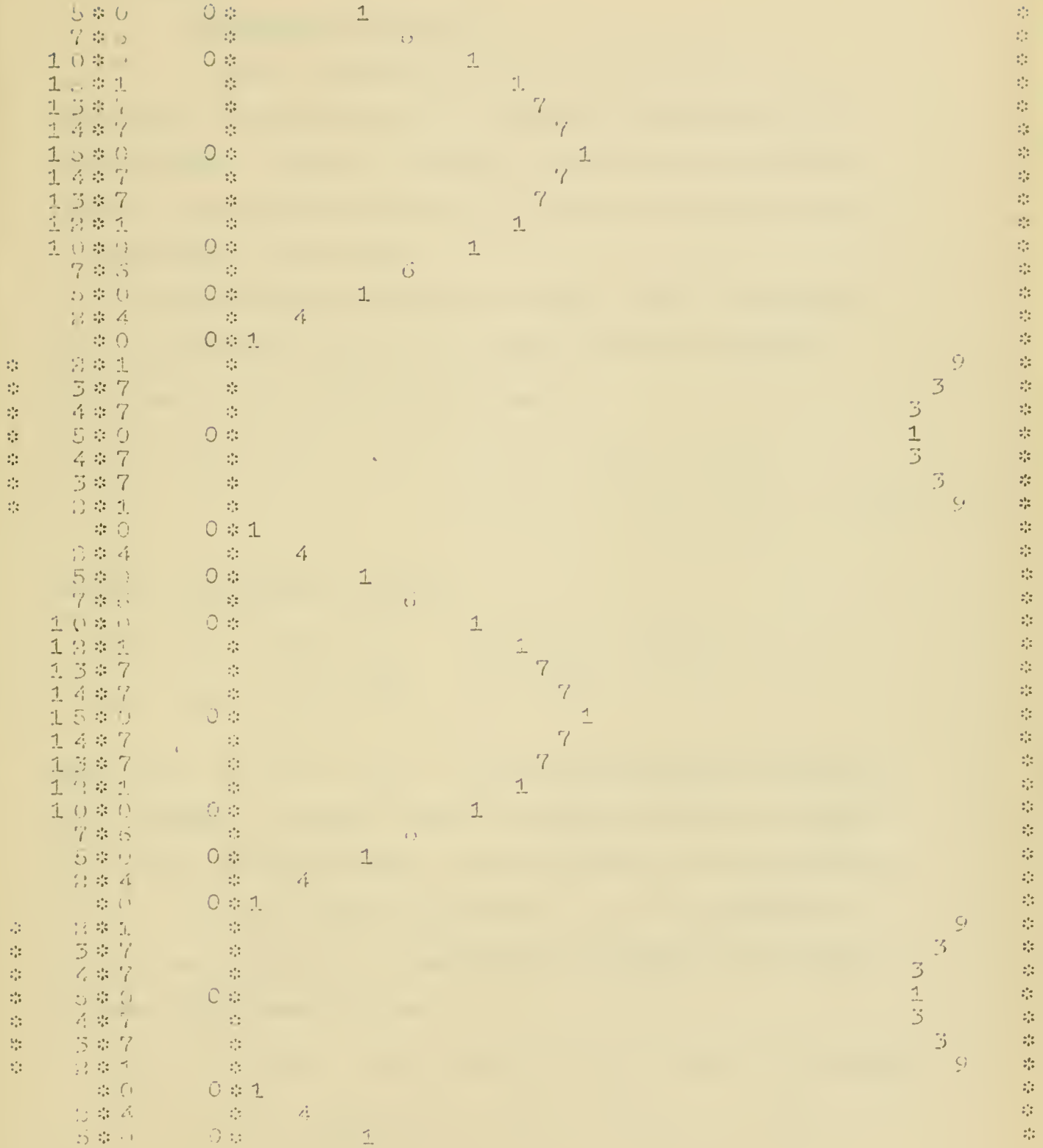
If the code is started at an earlier point, it first finds and displays the maximum and minimum of the N numbers x_i , for use in choosing suitable values of x_0 and m .

Graphs of $10(1 + \sin x)$ and of $10(1/2 + \sin x)$, produced using this code are attached.

GRAPH OF 10(1 + sin)



GRAPH OF 10(1/2 + sin)



III. METEOROLOGY

3.1 Mathematical Introduction.

Before proceeding to a detailed examination of the meteorological problems that have been handled on the machine let us consider briefly a method of solution of elliptic partial difference equations. This is desirable since the solution of such systems plays a dominant role in what follows.

We describe first the work of S. Frankel (1950)* in connection with the difference analogues of the Laplace and Poisson equations. Suppose that the difference system at hand is

$$x_{i+1,j} - 2x_{ij} + x_{i-1,j} + x_{i,j+1} - 2x_{ij} + x_{i,j-1} = f_{ij}$$
$$(i = 1, 2, \ldots I-1,\ j = 1, 2, \ldots J-1)$$

subject to the boundary conditions

$$x_{0j} = a_j \ , \ x_{I,j} = A_j \qquad \begin{pmatrix} i = 1, 2, \ldots I-1 \\ j = 1, 2, \ldots J-1 \end{pmatrix}$$
$$x_{i0} = b_i \ , \ x_{i,J} = B_i$$

where f_{ij} , a_j , A_j , b_i , B_i are known functions of the variables indicated. In order to remove the implicit nature of the system (1) we write, following Frankel, a new system which is an approximation to (1). In this new system we assume that y_{ij} is an approximant to x_{ij} , the solution of (1), satisfying the conditions (2) and define a "better" approximant z_{ij} as

$$z_{ij} = \omega\left(z_{i-1,j} + y_{i+1,j} + z_{i,j-1} + y_{i,j+1} - f_{ij}\right) - (4\omega - 1)y_{ij}$$
$$(i = 1, 2, \ldots I-1\ ;\ j = 1, 2, \ldots J-1)$$

---

* References for Section III will be found at the end of the section.

where

$$z_{0j} = a_j \ , \ z_{I,j} = A_j \ , \ z_{i0} = b_i \ , \ z_{i,J} = B_i$$
$$(i = 1, 2, \ldots\ I-1 \ ; \ j = 1, 2, \ldots\ J-1)$$

and ω is an as yet unspecified parameter. We shall see that by suitably changing ω we can materially alter the size of the deviation

$$| y_{ij} - z_{ij} |$$

We note that the choice $\omega = 1/2$ corresponds to the method of Liebmann (1918). We now discuss the situation for a general ω . We therefore consider the deviations

$$v_{ij} = y_{ij} - x_{ij} \ , \ r_{ij} = z_{ij} - x_{ij}$$

of y_{ij} , z_{ij} from x_{ij} the solution of the system (1). These deviations are related by the equations

$$r_{ij} = \omega (v_{i-1,j} + v_{i+1,j} + v_{i,j-1} + v_{i,j+1}) - (4\omega - 1) v_{ij}$$
$$(i = 1, 2, \ldots\ I-1, \ j = 1, 2, \ldots\ J-1)$$

$$r_{0j} = r_{Ij} = r_{i0} = r_{iJ} = 0 \ , \ v_{0j} = v_{Ij} = v_{i0} = v_{iJ} = 0$$

We may consider this system (4) as being of the form

$$r_{ij} = L_{ij}(v)$$

where L is a linear operator. If we change notation for the moment and let $w_{ij}^{(n)} = v_{ij}$, $w_{ij}^{n+1} = r_{ij}$. Then (5) states that

$$w^{(n+1)} = L(w^{(n)}) = L^{n+1}(w^{(0)})$$

From this it is clear that to discuss the convergence of $w^{(n)}$ to the zero function we must consider the behavior of the proper values and vectors of the operator L . We therefore consider the proper value problem

$$\lambda w_{ij} = L_{ij}(w)$$

or equivalently

$$\lambda w_{ij} = \omega (\lambda w_{i-1,j} + w_{i+1,j} + \lambda w_{i,j-1} + w_{i,j+1}) - (4\omega - 1) w_{ij}$$
$$(i = 1, 2, \ldots I-1, \quad j = 1, 2, \ldots J-1)$$

$$w_{0j} = w_{Ij} = w_{i0} = w_{iJ} = 0$$

Let w_{ij} be given by

$$w_{ij} = \lambda^{(i+j)/2} \sin \frac{\pi i p}{I} \sin \frac{\pi i q}{J} \quad , \quad (i, p = 1, 2, \ldots I-1 \; ; \; j, q = 1, 2, \ldots J-$$

These $(I-1)(J-1)$ functions of i , j satisfy the boundary conditions (8). We shall now see that the parameter ω can be so chosen as a function of p , q that the w_{ij} also satisfy (7), i.e. that they are the proper functions.

It is not difficult to see that

$$\lambda - 2\omega \left(\cos \frac{\pi p}{I} + \cos \frac{\pi q}{J} \right) \lambda^{1/2} - (4\omega - 1) = 0$$

Thus we have an equation which will determine λ_{pq} . This equation will apparently yield two roots for each p , q which would make it appear a priori that there were not $(I-1)(J-1)$ proper values but actually twice as many! However, we see with the help of the transformations $p' = I - p$, $q' = J - q$ that the proper value

$$\lambda_{p'q'} = -\lambda_{pq}$$

and thus there are only $(I-1)(J-1)$ proper functions. Thus the numbers λ_{pq} determined by (10) and the functions $w_{ij}^{(pq)}$ determined by (9) and (10) are the proper values and associated functions sought.

We see from (6) that the magnitude of each proper value must be

less than unity to ensure the convergence of $\omega^{(n)}$ to the zero function. We shall therefore concern ourselves with the sizes of these quantities viewed as functions of the parameter ω . It is not difficult to see that the smaller are the sizes of these proper values the more rapidly $\omega^{(n)}$ converges to the zero function. In particular, the relevant statement is that the smaller is the largest of the magnitudes of the proper values the more rapid is the convergence. If $\omega_{ij}^{(n)}$ can be expanded in terms of the functions (9), it may be seen that the rate of convergence of $\omega_{ij}^{(n)}$ to the zero function is measured by the quantity

$$\left\{ \sum_{p,q} | \lambda_{p,q} |^{2n} \right\}^{1/2} \leq \left\{ (I-1)(J-1) \right\}^{1/2} | \lambda_{MAX} |^{n}$$

and by another factor depending solely on the initial approximant ω_{ij}^{0}.

In equation (10) we let

$$t = \cos \frac{\pi p}{I} + \cos \frac{\pi q}{J}$$

and consider the roots of the equation (10) as functions of t and ω where $0 < t \leq t_M = \cos \pi/J + \cos \pi/I$, $4\omega - 1 > c$. From what has preceded we see that we wish to make as small as possible the magnitude of the largest root, i.e. we seek to determine

$$\min_{\omega} \max_{t} | \lambda(t, \omega) |$$

To do this consider two regions of the (ω, t) space: Region I, $\omega^2 t_M^2 - 4\omega + 1 \leq 0$; Region II, $\omega^2 t_M^2 - 4\omega + 1 \geq 0$. We see that

$$\phi(\omega) = \max_{t} | \lambda^{1/2}(t, \omega) | = \begin{cases} (4\omega - 1)^{1/2} & , \omega \text{ in} \\ \omega t_M + (\omega^2 t_M^2 + 1 - 4\omega)^{1/2} & , \omega \text{ in} \end{cases}$$

and further that

$$\omega t_M^2 + (\omega^2 t_M^2 + 1 - 4\omega)^{1/2} \geq (4\omega - 1)^{1/2}$$

Thus

$$\min_{\omega} \phi(\omega) = \min_{\omega \text{ in } I} (4\omega - 1)^{1/2} = (4\omega_M - 1)^{1/2}$$

where ω_M is the smaller root of

$$t_M^2 \omega^2 - 4\omega + 1 = 0$$

We accordingly have found the optimum value of ω, namely

$$\omega = \{2 - (4 - t_M^2)^{1/2}\} / t_M^2 \quad , \quad t_M = \cos \frac{\pi}{I} + \cos \frac{\pi}{J}$$

In terms of this quantity we find λ_{MAX}.

$$|\lambda_{MAX}| = 4\omega - 1 = \frac{4}{t_M^2} \{2 - (4 - t_M^2)^{1/2}\} - 1$$

$$\sim 1 - \sqrt{2} \cdot \pi (I^{-2} + J^{-2})^{1/2}$$

for large I, J. If we had chosen $\omega = 1/4$ (the Liebmann case), we would have had

$$|\lambda_{MAX}| = \frac{1}{2} \left(\cos \frac{\pi}{I} + \cos \frac{\pi}{J} \right) \sim 1 - \frac{\pi^2}{2} (I^{-2} + J^{-2})$$

If now for the moment at least we choose $I = J$, and if we define the rate of convergence C to be $\{1 - |\lambda_{MAX}|\}$, then (12) implies

$$C_F \sim 2\pi I^{-1}$$

and (13) implies

$$C_L \sim \pi^2 I^{-2}$$

Thus the so-called extrapolated Liebmann method of Frankel yields the following conclusion: If the parameter ω is chosen to satisfy (11),

then the rate of convergence of the iteration procedure (4), or equivalently (6), has a maximum for the choice (11) of the parameter ω. With this choice the measure of the rate of convergence varies linearly with I^{-1}; in the usual Liebmann procedure -- $\omega = 1/4$ -- this measure varies quadratically with I^{-1}.

In closing this discussion of Frankel's work we discuss the so-called Richardson (1910) method. Here we have in place of (4)

$$v_{ij} = \frac{1}{4}\left(v_{i-1,j} + v_{i+1,j} + v_{i,j-1} + v_{i,j+1}\right)$$

For the proper value problem in this case we set

$$v_{ij} = \lambda v_{ij} = \lambda \sin\frac{\pi i p}{I} \sin\frac{\pi i q}{J}$$

$$(i, p = 1, 2, \ldots I-1;\ j, q = 1, 2, \ldots J-1)$$

and find at once that

$$\lambda = \frac{1}{2}\left(\cos\frac{\pi p}{I} + \cos\frac{\pi q}{J}\right)$$

Thus

$$|\lambda_{MAX}| = \frac{1}{2}\left(\cos\frac{\pi}{I} + \cos\frac{\pi}{J}\right) \sim 1 - \frac{\pi^2}{4}\left(I^{-2} + J^{-2}\right)$$

and for $I = J$

$$c_R \cong \frac{\pi^2}{2} I^{-2} \sim \frac{1}{2} c_L \sim \frac{\pi}{4} I^{-1} c_F$$

In summary we see that the approximate measure of the rate of convergence for the Liebmann case is twice that for the Richardson case and it is $\pi/2I$ times that for the "extrapolated Liebmann" case.

We turn now to the work of D. M. Young (1951) on the general, linear elliptic partial difference system. Such a system may be put

in the form

$$\sum_{j=1}^{N} a_{ij} v_j + \alpha_i = 0 \quad , \quad i = 1, 2, \ldots N$$

In particular it is assumed that

(a) $a_{ii} > 0$ (all a_{ij} real)

(b) $a_{ii} \geq \sum_{j \neq i} |a_{ij}|$ (for some i the strict inequality holds)

(c) a_{ij} is such that given any two subsets S and T of the set $(1, 2, \ldots, N)$ such that S and T are disjoint there is an i in S, j in T for which $a_{ij} \neq 0$.

(d) there is a vector $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_N)$ such that $\gamma_i > \gamma_j$ and $a_{ij} \neq 0$ implies that under this ordering the i-th row of (a_{ij}) follows the j-th row.

(e) $a_{ij} = a_{ji}$

The following result is then readily seen to be valid: If (a_{ij}) is ordered by the vector γ, and if $a_{ij} \neq 0$, then $i < j$ implies $\gamma_j - \gamma_i = 1$ and $i > j$ implies $\gamma_i - \gamma_j = 1$.

We may replace the matrix (a_{ij}) by a new matrix (b_{ij}) as follows:

$$b_{ij} = \begin{cases} -a_{ij}/a_{ii} & , \quad i \neq j \\ 0 & , \quad i = j \end{cases}$$

Then we may express the system (14) as

$$v_i = \sum_{j=1}^{N} b_{ij} v_j - \alpha_i / a_{ii} \quad , \quad i = 1, 2, \ldots N$$

We shall now consider generalizations of the Richardson, Liebmann and Frankel procedures for the Poisson difference system to the equations (16). Since all our estimates involve differences between approximants and exact solutions of (16) we shall go directly to the perturbation equations corresponding to (16), i.e. to

$$\mu_i = \sum_{j=1}^{N} b_{ij} v_j \qquad , i = 1, 2, \ldots N.$$

whose solution is $v_i = 0$.

The generalization of Richardson's method is then:

$$\omega_i^{(n+1)} = \sum_{j=1}^{N} b_{ij} \omega_j^{(n)} \qquad , i = 1, 2, \ldots N$$

in which $\omega_i^{(n)}$ is the n -th approximant and $\omega_i^{(n+1)}$ is the $(n+1)$ -th approximant to the solution of (17).

We shall assume the proper values and associated vectors for this system, i.e. for

$$\lambda \omega_i = \sum_{j=1}^{N} b_{ij} \omega_j$$

to be λ_k , $\omega_{i,k}$, $(i, k = 1, 2, \ldots N)$

The generalization of Liebmann's method is:

$$\omega_i^{(n+1)} = \sum_{j=1}^{i-1} b_{ij} \omega_j^{(n+1)} + \sum_{j=i+1}^{N} b_{ij} \omega_j^{(n)} \qquad , (i = 1, 2, \ldots N)$$

We shall now determine the proper values and vectors of this system in terms of those of equations (19). The proper value problem at hand is

$$\mu \omega_i = \mu \sum_{j=1}^{i-1} b_{ij} \omega_j + \sum_{j=i+1}^{N} b_{ij} \omega_j \qquad , (i = 1, 2, \ldots N)$$

Let

$$w_i = \mu^{\gamma_i/2} w_{ik}$$

$$(i,k = 1, 2, \ldots\ldots N)$$

where γ_i is the i -th component of the vector γ mentioned in (d) above and w_{ik} is the k -th proper value for the system (19). We now determine μ so that (22) is a proper function for (21). Substituting (22) into (21) and making use of (19) and of the theorem stated directly below (d) above, we find readily that

$$\mu_k = \lambda_k^2 \qquad , k = 1, 2, \ldots\ldots N$$

Thus the numbers μ_k are the proper values for the system (21) corresponding to the functions (22).

The generalization of Frankel's method is:

$$w_i^{(n+1)} = 4\omega \left\{ \sum_{j=1}^{i-1} b_{ij} w_j^{(n+1)} + \sum_{j=i+1}^{N} b_{ij} w_j^{(n)} \right\} - (4\omega - 1) w_i^{(n)}$$

$$(i = 1, 2, \ldots\ldots N)$$

and the corresponding proper value problem is

$$\nu w_i = 4\omega \left\{ \nu \sum_{j=1}^{i-1} b_{ij} w_j + \sum_{j=i+1}^{N} b_{ij} w_j \right\} - (4\omega - 1) w_i$$

We again let

$$w_i = \nu^{\gamma_i/2} w_{ik}$$

and find

$$\nu \cdot \nu^{\gamma_i/2} \cdot w_{ik} = 4\omega \left\{ \nu \cdot \nu^{(\gamma_i - 1)/2} \sum_{j=1}^{i-1} b_{ij} w_{jk} + \nu^{(\gamma_i+1)/2} \sum_{j=i+1}^{N} b_{ij} w_{jk} \right\}$$

$$- (4\omega - 1) \nu^{\gamma_i/2} \cdot w_{ik}$$

Thus

$$= 4\omega \nu^{(\gamma_i+1)/2} \lambda_k w_{ik} - (4\omega - 1) \nu^{\gamma_i/2} \cdot w_{ik}$$

$$\nu - 4\omega \lambda_k \nu^{1/2} + (4\omega - 1) = 0$$

$$\nu - 4\omega\lambda_k \nu^{1/2} + (4\omega - 1) = 0$$

the generalization of equation (10).

Before analyzing the roots of equation (25), we discuss the reality of the λ_k . By (e) above the matrix (a_{ij}) and hence (b_{ij}) is symmetric. Thus for each k

$$\bar{\lambda} = \bar{\lambda} \sum_{i=1}^{N} |\omega_i|^2 = \overline{\sum_{j=1}^{N} \bar{\omega}_i b_{ij} \omega_j} = \sum_{j=1}^{N} \omega_i b_{ij} \bar{\omega}_j = \lambda \sum_{i=1}^{N} |\omega_i|^2 = \lambda$$

since b_{ij} is real and symmetric. Thus each λ_k is real. We turn now to a consideration of equation (25) and consider the quantity

$$\min_{\omega} \max_{k} |\nu(k, \omega)|$$

as before we distinguish two regions of the ω , k -space: Region I, $(2\omega\lambda_k)^2 - (4\omega - 1) \leq 0$; Region II, $(2\omega\lambda_k)^2 - (4\omega - 1) \geq 0$. Then we see that

$$\phi(\omega) = \max_{k} |\nu^{1/2}(k, \omega)| = \begin{cases} (4\omega - 1)^{1/2} & , \ \omega \text{ in } I \\ 2\omega\lambda_k + \{(2\omega\lambda_k)^2 - (4\omega - 1)\}^{1/2} & , \ \omega \text{ in } II \end{cases}$$

and further that in Region II

$$2\omega\lambda_k + \{(2\omega\lambda_k)^2 - (4\omega - 1)\}^{1/2} \geq (4\omega - 1)^{1/2}$$

Thus

$$\min_{\omega} \phi(\omega) = \min_{\omega \text{ in } I} (4\omega - 1)^{1/2} = (4\omega_M - 1)^{1/2}$$

where ω_M is, as before, the smaller root of

$$4\lambda_M^2 \omega^2 - 4\omega + 1 = 0$$

where λ_M is the numerically largest of the λ_k .

We are now able to compare the convergence rates of our various methods. To do this we may use Young's criterion or a somewhat simpler one.

If P is the absolute value of the largest proper value we may define the "rate of convergence" as $1-P$. Thus for the Richardson case we have

$$c_R = 1 - |\lambda_M|$$

for the Liebmann case

$$c_L = 1 - |\lambda_M^2|$$

and for the Frankel case

$$c_F = 1 - (4\omega - 1) = \frac{2}{\lambda_M^2}\left(1-\lambda_M^2\right)^{1/2}\left\{1-\left(1-\lambda_M^2\right)^{1/2}\right\} = \frac{2c_L^{1/2}}{1+c_L^{1/2}}$$

and thus we have for $|\lambda_M|$ near to 1.

$$c_L \sim 2c_R \quad , \quad c_F \sim 2c_L^{1/2} \sim 2^{2/3} c_R^{1/2}$$

3.2 Meteorological Introduction.

The problem of primary interest to the Meteorology Group during the year covered by this report was that of predicting the changes in atmospheric flow over periods of 12 to 48 hours. The accurate prediction of the field of motion is a necessary, though not a sufficient, prerequisite to the prediction of the more commonly thought of weather elements such as rainfall, cloudiness, humidity, etc. The difficulties associated with such prediction can be regarded as being of two kinds; firstly, our knowledge of the physical processes governing atmospheric motions is limited and, secondly, even the differential equations derived according to our present limited physical knowledge present considerable mathematical difficulties.

The physical assumption usually made is to assume that the flow is adiabatic, quasi-static, quasi-geostrophic and frictionless. Applying these approximations to the equations of motion and thermodynamic energy equation we can derive the following approximate equation for the pressure tendency

$$\left\{ \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{f(\zeta_g + f)}{g\,\partial \ln\theta/\partial z} \left(\frac{\partial^2}{\partial z^2} + \alpha \frac{\partial}{\partial z} + \beta \right) \right\} \frac{\partial p}{\partial t} = \gamma$$

in a rectangular coordinate system with x pointing east, y north and z upward. The quantity of f is the Coriolis parameter $2\Omega \sin\psi$, with Ω the angular speed of the earth's rotation and ψ the latitude; p is the pressure; α, β and γ are functions of p and its space derivatives; θ is the potential temperature; and ζ_g is the geostrophic vorticity:

$$\zeta_g \sim \frac{1}{\rho f} \left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \right) \equiv \frac{1}{\rho f} \nabla_h^2 p$$

where ∇_h is the horizontal operator, ρ the density and for simplicity of presentation the curvature of the earth is ignored.

Equation (1) is clearly elliptic or hyperbolic in the pressure tendency according as the factor multiplying the z-derivative terms is positive or negative. In the latter case the integration problem becomes very complicated and moreover there is some doubt whether the geostrophic assumption applies in such cases. Assuming, however, that the equation is elliptic everywhere and remains so we can, if we know the three-dimensional distribution of p at time t, by applying the boundary conditions that there shall be no influx or efflux of mass through the top and bottom of the atmosphere, solve the equation (1) by

some iterative process for the tendency $\partial p/\partial t$. Knowing this tendency we can then extrapolate to obtain p at a subsequent time $t + \Delta t$. The process may then be repeated a sufficient number of time to give a 12 or 24 hour forecast as required.

The procedure just outlined would, however, require such a great amount of computation and storage that it was felt to be unwise to start "numerical weather prediction" with such a complicated problem. More simplified models have therefore been derived to test the validity of some of the basic assumptions. Three such models have so far been used. These may be regarded as n level models for $n = 1, 2, 3$; n being the number of points in the vertical at which initial pressure data is required. The derivation of the n-level model will now be described and then a description of the numerical methods used in each of the three cases will be given.

3.3 The n-level Model.

The following notation in addition to that given in section I will be adopted: ϕ is the geopotential, $g z$, $\frac{D}{Dt}$ the individual derivative in an isobaric surface, $\zeta = f + \frac{1}{f} \nabla^2 \phi$ the vertical component of absolute vorticity, c_v and c_p the specific heats of air at constant volume and constant pressure, and ω the individual derivative of p. Neglecting vertical advection of vorticity and the conversion of horizontal to vertical vorticity, elimination of the divergence term between the continuity equation and the equation for the vertical component of vorticity gives

$$\frac{D}{Dt} \ln \zeta = \frac{\partial \omega}{\partial p}$$

Also the energy equation for adiabatic flow may be written in the form

$$\omega = -\frac{1}{\partial \ln\theta/\partial p}\,\frac{D}{Dt}\ln\left(-\partial\phi/\partial p\right)$$

since $\ln\theta$ = constant + $\left(\frac{c_v}{c_p}\right)\ln p + \ln\left(-\partial\phi/\partial p\right)$

Divide the interval $p = 0$ to $p = p_0$ in n equal subintervals, δp, and denote quantities at the mid-point of each interval by the subscript k ($k = 1, 2, \ldots, n$) and points at the upper and lower end-points of each interval by $k - 1/2$ and $k + 1/2$, respectively. If we put $k = 1$ at $p = 1/2\ \delta p$, and $k = n$ at $p_0 - 1/2\ \delta p$, the points $p = 0$ and $p = p_0$ correspond to $k = 1/2$ and $k = n + 1/2$, respectively.

Expressing all vertical derivatives as centered finite differences (and approximating θ when it occurs as a coefficient by its mean value) equations (1) and (2) become, for a layer centered at pressure p_k:

$$\left(\frac{D}{Dt}\right)_k \ln\zeta_k = \frac{1}{\delta p}\left\{\omega_{k+1/2} - \omega_{k-1/2}\right\}$$

$$\omega_{k+1/2} = \delta p \cdot \frac{\bar\theta_{k+1/2}}{\bar\theta_k - \bar\theta_{k+1}}\left(\frac{D}{Dt}\right)_{k+1/2}\ln\left(\phi_k - \phi_{k+1}\right)$$

$$\omega_{k-1/2} = \delta p \cdot \frac{\bar\theta_{k-1/2}}{\bar\theta_{k-1} - \bar\theta_k}\left(\frac{D}{Dt}\right)_{k-1/2}\ln\left(\phi_{k-1} - \phi_k\right)$$

Approximating ϕ_k by $\left(\phi_{k+1/2} + \phi_{k-1/2}\right)/2$ it follows from the geostrophic assumption that

$$\left(\frac{D}{Dt}\right)_{k+1/2} Z_1 = \left(\frac{D}{Dt}\right)_k Z_1 = \left(\frac{D}{Dt}\right)_{k-1/2} Z_1$$

where $\zeta_k = \ln(\phi_k - \phi_{k+1})$ or $\ln(\phi_{k-1} - \phi_k)$. Hence eliminating $\omega_{k-1/2}$ and $\omega_{k+1/2}$ between equations (3), (4), and (5) we obtain, for $k \neq 1, n$, (since κ is a constant)

$$\left(\frac{D}{Dt}\right)_k \ln \zeta_k = \left(\frac{D}{Dt}\right)_k \ln \left\{ \frac{(\phi_k - \phi_{k+1})^{\kappa_{k+1/2}}}{(\phi_{k-1} - \phi_k)^{\kappa_{k-1/2}}} \right\}$$

or

$$\left(\frac{D}{Dt}\right)_k \left\{ \zeta_k \frac{(\phi_{k-1} - \phi_k)^{\kappa_{k-1/2}}}{(\phi_k - \phi_{k+1})^{\kappa_{k+1/2}}} \right\} = 0$$

where

$$\kappa_{k+1/2} = \bar{\theta}_{k+1/2} \Big/ (\bar{\theta}_k - \bar{\theta}_{k+1})$$

At the levels adjacent to the ground and the top of the atmosphere we substitute the boundary condition $\omega_{1/2} = 0 = \omega_{n+1/2}$. For these levels therefore

$$\left(\frac{D}{Dt}\right)_1 \left\{ \zeta_1 \cdot \frac{1}{(\phi_1 - \phi_2)^{\kappa_{1/2}}} \right\} = 0$$

$$\left(\frac{D}{Dt}\right)_n \left\{ \zeta_n (\phi_{n-1} - \phi_n)^{\kappa_{n-1/2}} \right\} = 0$$

Equations (7), (8), and (9) thus give us n independent equations for the n unknowns ϕ_1 , ϕ_2 , ..., ϕ_n . It should be noted, however, that the solution of this set of equations as n tends to infinity does not tend to the solution of the more general equation (3.2.1) as additional assumptions have been made in the derivation of the n -layer model.

3.4 Integration of the barotropic (one-level) model.

The barotropic model may be regarded as the n -level

model for which $n = 1$. This gives us the following equation at the mid level of the atmosphere

$$\frac{D\zeta}{Dt} = \frac{\partial \zeta}{\partial t} + \vec{v} \nabla \zeta = 0$$

To discuss the equation, we map the spherical earth conformally onto a plane and introduce a Cartesian coordinate system (x , y) in the plane. If m is the magnification factor, the del operator on the earth becomes multiplied by m in passing to the plane. Introducing the notation $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ for the Laplace operator in the plane and $J(\alpha, \beta) = \partial(\alpha, \beta)/\partial(x, y)$ for the Jacobian of α and β, we can write (1) in the form

$$\begin{cases} \nabla^2 \phi = \zeta \\ \zeta = \frac{m^2}{f} \zeta + f \\ \frac{\partial \zeta}{\partial t} = J(\zeta, \phi) \end{cases}$$

Two possible computation schemes can be set up in order to solve these equations. Mathematically these are exactly equivalent, the only difference being that in the first the history of the motion is carried by ϕ while in the second it is carried by ζ.

Method of solution A: The following computations leading from time t to time $t + \Delta t$ are performed. Starting with $\phi^{t-\Delta t}$ and ϕ^t

1. Calculate $(\partial \zeta/\partial t)^t$ from the finite difference analogue of

$$(\partial \zeta/\partial t)^t = J\left(f^{-1} m^2 \nabla^2 \phi^t + f, \phi^t \right)$$

2. Solve for $(\partial \phi/\partial t)^t$ from the finite difference analogue

of the Poisson equation

$$\nabla^2 \left(\partial\phi/\partial t\right)^t = \left(\partial\zeta/\partial t\right)^t$$

3. Calculate $\phi^{t+\Delta t}$ from

$$\phi^{t+\Delta t} = \phi^{t-\Delta t} + 2\Delta t \left(\partial\phi/\partial t\right)^t$$

Method of solution B. Starting with $\zeta^{t-\Delta t}$ and ζ^t

1. Solve for ϕ^t from the finite difference analogue of the Poisson equation,

$$\nabla^2\phi^t = \zeta^t$$

2. Calculate $\left(\partial\zeta/\partial t\right)^t$ from the finite difference analogue of

$$\left(\partial\zeta/\partial t\right)^t = J\left(f^{-1}m^2\zeta^t + f, \phi^t\right)$$

3. Calculate $\zeta^{t+\Delta t}$ from

$$\zeta^{t+\Delta t} = \zeta^{t-\Delta t} + 2\Delta t \left(\partial\zeta/\partial t\right)^t$$

Procedure A was used when the first predictions were made in 1950 using the Eniac and reported by Charney et. al. (1950). In this report it was shown that for a region bounded by a single closed curve the boundary conditions on (12) are: ϕ must be prescribed on the boundary for all time: η (or ζ) must be prescribed as a function of time when fluid is entering the region, but must not be prescribed when fluid is leaving the region. From the geostrophic assumption it follows that influx or efflux is determined only by the boundary values of ϕ . The simplest condition on ϕ is that it remain constant with time. For short periods of time this rather unrealistic condition will not effect

the internal motion. These boundary conditions were used in the Princeton computations for the barotropic and two-layer models but alternative ones were experimented with on the barotropic model and used in the three-layer model. These conditions will be described in a later paragraph.

The stipulation ϕ = constant implies the homogeneous boundary condition $(\partial\phi/\partial t)^t$ = 0 on the boundary for the Poisson equation (4). With this condition, and with a rectangular boundary, the Fourier transform method is well adapted to a computer with a small variable-storage capacity such as the Eniac. This method is described in the above mentioned report and also by Charney and Phillips (1953). In the latter report it is shown that for a grid of $p \times q$ points $(2p+2q-4) \times$ $\times (p-1)(q-1)$ multiplications are required per time step and that storage must be provided for three quantities per grid point.

In procedure B the boundary conditions on ϕ for the finite difference analogue of the Poisson equation (6) are not homogeneous but state that ϕ be a given function of time on the boundary. Therefore, unless we set ϕ = 0 on the boundary -- an obviously unreasonable procedure -- the Fourier transform method is not conveniently applicable. An alternative, however, is to use an iterative procedure related to the Southwell "relaxation" method. It should be mentioned that, whereas the Fourier transform yields an exact solution within the limits of round off error to the finite difference Poisson equation, the iterative methods in general do not. If, however, extreme accuracy is not required,the disadvantage is far outweighed by other advantages. One is that the methods are logically simpler and require few instructions in the machine, thus taking up less memory space and also if too great

accuracy is not required (or if the initial guess is good) few iterations are needed, and the computation is correspondingly faster -- for the fastest barotropic code the number of multiplications is approximately 10 pq as compared with $(2p+2q-4)(p-1)(q-1)$ for the Fourier series method. The principal consideration, however, which led to the choice of an iterative method was that it could also be used for solving the elliptic partial differential equations with variable coefficients that occur in the more complicated models and therefore its application to the barotropic model could be expected to provide valuable experience.

The iterative procedure adopted was Frankel's (1950) "extrapolated Liebmann method". Dropping the time subscript on ϕ, equation (6) may be written

$$\mathcal{R}\phi_{ij} = \phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - 4\phi_{ij} - (\Delta s)^2 J_{ij}$$

and Frankel's method is defined by the iterative equation

$$\phi_{ij}^{\nu+1} = \phi_{ij}^{\nu} + \alpha \mathcal{R}\phi_{ij}^{\nu,\nu+1}$$

where the values of ϕ at the ν-th stage of the iteration are denoted by ϕ^{ν} and where the double index ν, $\nu+1$ signifies that the index ν or $\nu+1$ is to be used according as the subscripts are i,j; $i+1, j$; $i, j+1$, or $i-1, j$; $i, j+1$. Frankel (1950) has determined the optimum value of the overrelaxation factor α, together with the corresponding convergence rates which are compared with those for the method of Liebmann (1918) in which $\alpha = 1/4$. In order to find an α which is only a function of p and q Frankel had to assume

nothing about the initial error distribution. However, in the first few iterations the size of the residuals depends on the initial error distribution which has in the meteorological problem some physical significance and therefore certain predominating components. Thus in actual practice a value of α lying between Frankel's and Liebmann's was found by experiment to give the most rapid convergence to the accuracy warranted by the problem.

As many steps in the iteration process are performed as are needed to reduce the absolute value of each residual below a certain prescribed value. The number of iterations will therefore depend on the accuracy of the initial approximation. In procedure B an obvious initial approximation is the value of ϕ_{ij} at the previous time step. If, however, the penultimate ϕ_{ij} is stored, a better approximation can be obtained by linear extrapolation from the preceding two time steps. Tests were also run using quadratic and least square extrapolations from the preceding three time steps but without any improvement. Whether or not improvement takes place is clearly associated with the time period of the motions being considered.

It can be shown (see Charney et. al., 1950) that the choice of the space and time increments, Δs and Δt, is restricted by the computational stability criterion.

$$\Delta s / \Delta t \geq \sqrt{2}\, m\, |v|_{max}$$

where $|v|$ max is the maximum particle speed in the forecast region. This criterion must be satisfied if small perturbations are not to be amplified exponentially. This ratio $\Delta s/\Delta t$ does not, however, correspond to the optimum ratio demanded by consideration of truncation

error. For this the increments Δs and Δt should be chosen so as to give the same definition of the field of motion along the time axis as along the space axis since nothing is gained if the increment in one variable is chosen so small that the truncation error in the variable is small, as long as the truncation error in the other variable remains large. This requires that the ratio $\Delta s/\Delta t$ should be equal to the local speed of propagation of the flow pattern and not the local particle velocity, i.e.

$$\Delta s/\Delta t \sim \sqrt{2}\, m\, c$$

where c is the typical speed of displacement of the flow patterns. At the 500 mb level $|v|$ max is of the order of 50 m/sec whereas a typical c is at most 20 m/sec. Hence, for a given Δs the Δt called for by the criterion (11) is perhaps two or three times too small. The computation will therefore take two or three times as long as it would if only (12) had to be satisfied.

There appears to be a way out of the difficulty. From the point of view of computational stability, (1) behaves purely as an advective equation when Δs is sufficiently small, i.e. as though the velocity were a fixed function of space. It does not matter that ζ is itself a function of derivatives of ϕ . Hence in using the method of integration B, the stability criterion (11) applies only to (7) and (8) and not to (6). This suggests that a new field of ϕ need not be calculated from ζ at the end of each time step. If, for example, Δt is required by (11) to be one hour, we may advect the vorticities for two or three one-hour steps while holding ϕ fixed as a function of time, and only after the second or third step redetermine a new field of ϕ and therefore of velocity. Thus, the solution of the Poisson equation

(6) need be performed only as often as is required by consideration of truncation error.

The above method cannot be employed with procedure A, because in this system it is ϕ and not ζ (or J) that carries the history of the motion and therefore ϕ must be determined at each time step.

We now turn more specifically to actual procedures used in obtaining a series of 24 hour barotropic forecasts.

Forecasts were desired for an overall area of 5400 by 5400 km, with an interior region of 3300 by 3300 km which was expected to be unaffected by the boundary conditions after 24 hours. This interior region lay almost entirely between 30° and 60° N. For an area of this size a Lambert conformal map projection with standard parallels at 30 and 60 deg. is well suited since it distorts distances between these latitudes by less than 3.5 per cent. For its projection we can approximate the mapping factor m and the sine of latitude, both of which are required in the computations, by

$$m^2 = A\rho^2 (1 - \sin^2\alpha)^{-1}$$

$$\sin\psi = B - C\left(\frac{\rho}{\rho_E}\right)^2$$

where ρ is the polar distance on the map, ρ_E the distance from pole to equator on the map and A, B, C are constants. These formulae give sin ψ to within 1.0 per cent and m^2 to within 2.0 per cent.

The mesh size of 628.5 km at 45° N used in the Eniac computations proved too large and for the Princeton computations it was therefore reduced to 300 km. This was justified by the following reasoning: considering the Taylor's series expansions of $f(x+\Delta s)$ and $f(x-\Delta s)$

we find that

$$\frac{1}{f'(x)}\left\{\frac{f(x+\Delta s)-f(x-\Delta s)}{2\Delta s}-f'(x)\right\}\simeq\frac{\Delta s^2}{6}\cdot\frac{f'''(x)}{f'(x)}+\ldots(\Delta s^4)$$

let $f=\cos(2\pi x/L)$ then

$$\frac{\text{Error in } f'(x)}{f'(x)}=\frac{2\pi^2}{3}\left(\frac{\Delta s}{L}\right)^2$$

Taking a value of L = 2000 km, corresponding to relatively small scale motion, the percentage error introduced by evaluating a first derivative by a centered finite difference with Δs = 300 km is 15 per cent as compared with 65 per cent for Δs = 628.5 km.

The corresponding value of Δt was determined by the stability criterion (11). This led to a maximum value for Δt of one hour and 11 minutes and therefore the actual choice for Δt was one hour.

The actual procedure used for the main series of forecasts is now given in more detail.

1. With $\zeta_{ij}^{\tau-1}$ and ζ_{ij}^{τ} stored for all grid points -- determined at τ = 0 from ϕ^0 by (6) -- equation (9) is solved by means of the iteration procedure (10) with use of $\phi_{ij}^{\tau-1}$ as the initial guess.

2. $(\partial\zeta/\partial t)_{ij}^{\tau}$ is determined from

$$(\partial\zeta/\partial t)_{ij}^{\tau}=\frac{1}{(2\Delta s)^2}\left\{(\zeta_{i+1,j}^{\tau}-\zeta_{i-1,j}^{\tau})(\phi_{i,j+1}^{\tau}-\phi_{i,j-1}^{\tau})-(\zeta_{i,j+1}^{\tau}-\zeta_{i,j-1}^{\tau})\times(\phi_{i+1,j}^{\tau}-\phi_{i-1,j}^{\tau})\right\}$$

where

$$\zeta_{ij}^{\tau}=\frac{m_{ij}^2}{f_{ij}}\nabla^2\phi_{ij}^{\tau}+f_{ij}$$

The quantities f_{ij} and m_{ij} are evaluated from (13) and (14). The relevant quantity here is $v=(p/p_E)^2$ which may be written

$$U = \tfrac{1}{2}\varepsilon^{-2}(\Delta s)^2 \sum_{\ell}\left\{(i_n - i)^2 + (j_n - j)^2\right\}_{j}$$

where the pole is given the integral coordinates i_p and j_p.

3. At interior points $\zeta_{ij}^{\tau+1}$ is obtained from

$$\zeta_{ij}^{\tau+1} = \zeta_{ij}^{\tau-1} + \epsilon(\tau)\,\Delta t\,(\partial\zeta/\partial t)_{ij}^{\tau}$$

At $\tau = 0$ it is necessary to employ non-centered differences, that is, to set $\zeta_{ij}^{-1} = \zeta_{ij}^{0}$ and $\epsilon(0) = 1$. At all other times $\epsilon(\tau) = 2$.

At outflow points $\zeta_{ij}^{\tau+1}$ is obtained by linear extrapolation from the two nearest points along the normal to the boundary at that point, while at inflow points $\zeta_{ij}^{\tau+1}$ is set equal to the initial value ζ_{ij}^{0} (inflow or outflow is determined by differencing the ϕ_{ij} on the boundary adjoining the boundary point in question).

4. Steps 1 to 3 are now repeated 24 times in order to obtain the required 24 hour forecast.

The storage of data clearly depends on the amount of digital significance in the data. In the barotropic forecasts the 500-mb height used varies by approximately 3500 ft. over the forecast area and is probably only known accurately to within 50 ft. Hence the accuracy is one part in seventy and if the height is expressed as a deviation from the mean value, it can be stored in seven binary places. Actually in the course of the computation more than seven digits must be stored in order to allow for round off errors. Whenever a number is stored to n places it is rounded off thus producing an error lying between $2^{-(n+1)}$ and $-2^{-(n+1)}$. If the process during which

the number is stored is repeated m times, the probable error will be increased by the factor $m^{1/2}$, provided the round off errors are random.

Round-off errors in ϕ and J also enter indirectly, as for example, in affecting the accuracy of $(\partial J/\partial t)^{\tau}_{ij}$ in (15), which in turn affects the accuracy of $J^{\tau+1}_{ij}$ in (18) and therefore of $\phi^{\tau+1}_{ij}$ obtained from the Poisson equation (6). It should be pointed out that, when an iterative process is used to solve (6), the error in ϕ is not only caused by round off per se, but is also a function of the accuracy with which the final iterated solution satisfies (9). The ϕ's were stored in the machine as scaled quantities $\hat{\phi}$, such that $1 > \hat{\phi} > 0$, and $d\phi/d\hat{\phi} \sim g \times 3500$ ft. The iterative process was stopped when $|\hat{\phi}^{\nu+1} - \hat{\phi}^{\nu}| < \delta$ for all i and j. Clearly if $\delta = 2^{-N}$, the successive guesses of $\hat{\phi}$ must be stored to at least $N+2$ places, if the pure round off error incurred in storing $\hat{\phi}^{\nu}$ is not to interfere greatly with the convergence of (10).

To calculate the resultant effect of these errors ab initio is a difficult, if not impossible, task. Accordingly a short series of 12- and 24-hour numerical integrations were performed, with varying digital significance for the ϕ's and J's. A basic forecast was first obtained with $\delta = 2^{-14}$ and J stored to twenty places, and this was then compared to forecasts from the same initial data where δ was gradually increased to 2^{-8} and the storage of J decreased to 14 binary places. It was found that the error in ϕ at the end of 24 one-hour time steps remained negligible in comparison with the error of observation until δ became greater than 2^{-9}. On the basis of these experiments, twelve places for ϕ ($\delta = 2^{-10}$) and 14 for each J were

decided upon. (It later developed that as few as ten places could be used for ζ .)

With regard to the time truncation error it was observed in the Eniac calculations, Charney et. al. (1950), that increasing the time interval from one to two and then three hours produced no appreciable change in a 24-hour forecast. The computation proved stable for even the three-hour interval because the space increment was over twice its present value, and the particle speeds were lower than 50 m/sec. In the present case, integrations were performed for $\Delta t = 1$, $4/3$ and two hours to test the application of the stability criterion. Instability manifests itself by the exponential growth of small disturbances in the field of geopotential or vorticity. Their rate of amplification can be estimated from the theory given in the above mentioned report. One obtains the result that the maximum amplification factor, after n time steps, is $e^{n\theta}$ where $\theta = \operatorname{arc\,cosh}\left\{\sqrt{2}\, m |v|_{max} \cdot \frac{\Delta t}{\Delta s}\right\}$. In the example tested instability was observed for $\Delta t = 4/3$ hours and $\Delta t = 2$ hours. In the case $\Delta t = 4/3$ hours, θ was determined from the rate of growth of small deviations from the computation for $\Delta t = 1$ hour which was stable. With use of this value of θ , the corresponding value for the case $\Delta t = 2$ hours and the number of time steps required for the amplitude of a given perturbation to increase by a certain amount were computed and found to agree with observation.

As previously observed, the computational stability condition permits the advection of vorticity for several (n) time steps in a velocity field that is held stationary during this period. The velocity field is redetermined by (9) only at the end of the period. We

will use the notation nJ/L — n Jacobian operations per extrapolated Liebmann operation -- to denote this procedure. Two experiments were made to test its accuracy. These consisted of the preparation of two 12-hour forecasts, the first by means of the $2J/L$ procedure and the second by the $3J/L$ procedure. These forecasts were then compared with the $1J/L$ forecast from the same data. In all these forecasts Δt remained one hour. Although both the $2J/L$ and $3J/L$ forecasts were stable, they differed too much from the $1J/L$ forecast to warrant continued use of the process.

The following scheme, involving the storage of four quantities per grid point, has since been found to eliminate the above mentioned errors. At time t, suppose we have the quantities ϕ^{t-n}, ϕ^{t}, J^{t-1} and J^{t} stored. In the first step J^{t+1} is calculated from (15) and (18); the quantities J^{t-1} are replaced by J^{t} in the memory and J^{t} is replaced by J^{t+1}. The value of ϕ^{t+1} is then determined not by solution of the Poisson equation (9), but by linear extrapolation from ϕ^{t-n} and ϕ^{t}. With the aid of this new value J^{t+2} is calculated, and a ϕ^{t+2} is determined again by extrapolation from ϕ^{t-n} and ϕ^{t}. The whole process is carried out n times, until the quantities ϕ^{t-n}, ϕ^{t}, J^{t+n-1} and J^{t+n} are held in the memory. At this stage ϕ^{t-n} is replaced by ϕ^{t} and ϕ^{t+n} is determined from J^{t+n} by solution of the Poisson equation. The induction is then complete. Initially it is necessary to advance in single time steps, until n time steps have been taken. Only then do we have two quantities for the extrapolation, namely ϕ^{0} and ϕ^{n} and are ready to begin the induction.

The above considerations suggest an improvement of procedure B, in which a new ϕ is computed at every time step from each new ζ. Assuming that it is possible to store four quantities per grid point, we suppose that $\phi^{\tau-2}$ is stored in addition to $\phi^{\tau-1}$, $\zeta^{\tau-1}$ and ζ^{τ} at the stage just preceding the calculation of ϕ^{τ} from ζ^{τ}. Instead of taking $\phi^{\tau-1}$ as the initial guess in the iterative process, we extrapolate linearly from $\phi^{\tau-2}$ and take instead $2\phi^{\tau-1} - \phi^{\tau-2}$. The initial error is thereby considerably reduced, and fewer iterations are required to reduce the residual below a prescribed value.

The main series of six 24-hour forecasts was made using the straightforward procedure B, a time interval of one hour, and a rectangular grid of 19 x 19 = 361 points. These forecasts each took 48 minutes at full speed. (In actual fact the machine was run at half speed most of the time for engineering reasons.) In each step the machine performed 64,000 multiplications, 4,200 divisions, 314,000 additions and subtractions and 1,467,000 additional orders. Of the 64,000 multiplications, 52,000 were used in solving the Poisson equation and 12,000 in computing the Jacobian. Altogether, six-fold more time was spent in the Poisson part of the calculation than in the Jacobian part. With the ϕ 's stored as numbers smaller than 1, approximately 13 iterations of the extrapolated Liebmann process were required to reduce the quantity $|\hat{\phi}^{\nu+1} - \hat{\phi}^{\nu}|$ below 2^{-10}.

As it was felt that unless the simple code could be run quickly the more complicated methods would be prohibitively long, it was decided to speed up the barotropic code as much as possible. While for research purposes the speed is not unduely important, it would become a primary

consideration in any routine use of numerical methods in weather forecasting. The first change consisted of adopting the modified procedure B in which the ϕ guess is obtained by extrapolation, and by rationalizing the remainder of the code. This led to a computation time of 24 minutes at full speed for a 24-hour forecast. The code was then further speeded up by storing, in addition to the quantities required by the modified B procedure, the values of f and m^2/f at each grid point. Further the differentiation made in stage 3 of procedure B between inflow and outflow points was removed and all boundary points treated as inflow points. Physically it would seem that this condition is definitely wrong but whereas the old one produced a certain amount of mathematical instability near the boundary the new one appeared to be quite stable. This fact was reflected in the decreased number (approximately three) iterations required to solve the Poisson equation. This resulted in approximately equal time being spent in the two sections of the code and a total computing time of about six minutes for a 24-hour forecast, at full speed. In comparison to this time it is of interest to note that the first 24-hour barotropic forecast performed on the ENIAC took 24 hours to complete.

3.5 Integration of the two-level model.

In the case $n = 2$ the system of equations (3.3.7), (3.3.8), and (3.3.9) for the n-layer model reduce to the two equations

$$\left(\frac{D}{Dt}\right)_1 \frac{3_1}{(\phi_1-\phi_2)^k} = 0 \quad , \quad \left(\frac{D}{Dt}\right)_2 \frac{3_2}{(\phi_1-\phi_2)^{-k}} = 0$$

where ϕ_1 and ϕ_2 are the geopotentials at the 250 mb and 750 mb levels, respectively, and the exponent k has a value of approximately 9.

The difference $\phi_1 - \phi_2$ is proportional to the mean temperature in the layer between 750 and 250 mb. Its deviation δD $(\equiv \phi_1 - \phi_2 - D)$ from its mean value D is not greater than 6 or 7 per cent. Hence, we may write with fair approximation

$$(\phi_1 - \phi_2)^{\kappa} = D^{\kappa-1}(D + \kappa \delta D)$$
$$(\phi_1 - \phi_2)^{-\kappa} = D^{-\kappa-1}(D - \kappa \delta D)$$

The equations (1) then become

$$\left(\frac{D}{Dt}\right)_1 \frac{\zeta_1}{D + \kappa \delta D} = 0 \quad , \quad \left(\frac{D}{Dt}\right)_2 \frac{\zeta_2}{D - \kappa \delta D} = 0$$

We now, as in the treatment of the barotropic model, introduce a Lambert conformal mapping and a Cartesian system of coordinates x, y in the conformal plane. With the notation $\phi_2 = \phi$, $\phi_1 = \phi'$, $\zeta_2 = \zeta$, $\zeta_1 = \zeta'$ (2) becomes

$$\partial q / \partial t = m^2 f^{-1} J(q, \phi)$$

and

$$\partial q' / \partial t = m^2 f^{-1} J(q', \phi')$$

where

$$q = \left(m^2 f^{-1} \nabla^2 \phi + f\right)\left\{(\kappa+1)D - \kappa(\phi' - \phi)\right\}^{-1}$$

and

$$q' = \left(m^2 f^{-1} \nabla^2 \phi' + f\right)\left\{\kappa(\phi' - \phi) - (\kappa-1)D\right\}^{-1}$$

These equations, as stated above, refer to the 250 and 750 mb surfaces which unfortunately are not standard meteorological surfaces. However, because of the crude manner in which the vertical scale of

motion is treated, there is no difficulty in deriving an equivalent set of equations which apply at the standard levels, 300 and 700 mb, the only change being that the quantity κ is now given by $0.64\ \bar{\theta}_5 \left(\bar{\theta}_3 - \bar{\theta}_7\right) \sim 8$ instead of the previous value of 9.

The corresponding lateral boundary conditions may be found using the same heuristic reasoning as was used in thecase of the barotropic model. This results in exactly analogous conditions the only difference being that we now have to consider everything at two levels instead of one.

The integration is carried out quite by analogy with the procedure B used for the integration of the barotropic equation. At time t, we suppose that the quantities q'^{t-1}, q'^{t}, q^{t-1} and q^{t} are stored. The difference analogues of (5) and (6), namely

$$\nabla^2\phi_{ij} = \frac{f_{ij}\, q_{ij}}{m_{ij}^2}\left\{(\kappa+1)D - \kappa(\phi_{ij}' - \phi_{ij})\right\} - \frac{f_{ij}^2}{m_{ij}^2}$$

and

$$\nabla^2\phi_{ij}' = \frac{f_{ij}\, q_{ij}'}{m_{ij}^2}\left\{\kappa(\phi_{ij}' - \phi_{ij}) - (\kappa-1)D\right\} - \frac{f_{ij}^2}{m_{ij}^2}$$

are then solved for ϕ' and ϕ' subject to the conditions that ϕ and ϕ' are constant on the boundary.

After ϕ^t and ϕ'^t have been determined, $(\partial q'/\partial t)^t$ and $(\partial q/\partial t)^t$ are evaluated from (3) and (4), and $q_{ij}'^{t+1}$ and q_{ij}^{t+1} are calculated from the formulae

$$q_{ij}'^{t+1} = q_{ij}'^{t-1} + 2\Delta t\,(\partial q'/\partial t)_{ij}^{t}$$

and

$$q_{ij}^{t+1} = q_{ij}^{t-1} + 2\Delta t\,(\partial q/\partial t)_{ij}^{t}$$

Again, in the evaluation of the finite difference Jacobian at points adjacent to the boundary, we invoke the boundary conditions in ψ' and ψ .

The principal problem is the solution of (7) and (8). Subtraction of these two equations gives

$$\nabla^2 \chi_{ij} - \frac{K f_{ij}}{m_{ij}^2} (\psi'_{ij} + \psi_{ij}) \chi_{ij} = - \frac{r_{ij}}{m_{ij}^2} \frac{D}{2} [(K+1)\psi_{ij} + (K-1)\psi'_{ij}]$$

where $\chi_{ij} = \phi'_{ij} - \phi_{ij}$. This is a difference analogue of a non-homogeneous self-adjoint linear partial differential equation of elliptic type in the single variable χ_{ij} , a solution of which in the continuous case is known to exist. It can be solved for $\chi_{ij} = \phi'_{ij} - \phi_{ij}$ and substitution of ϕ'_{ij} ϕ_{ij} into (7) yields the difference analogue of a Poisson equation in the single variable ϕ_{ij} . Solving the latter for ϕ_{ij} , we obtain φ'_{ij} by adding ϕ_{ij} to $\phi'_{ij} - \phi_{ij}$.

With the definitions

$$\gamma_{ij} = K f_{ij} \; m_{ij}^{-2} (\psi'_{ij} + \psi_{ij})(\Delta s)^2$$

and

$$\alpha_{ij} = - r_{ij} \; m_{ij}^{-2} \frac{D}{2} [(K+1)\psi_{ij} + (K-1)\psi'_{ij}] (\Delta s)^2$$

(9) becomes

$$\bar{\chi}(\chi_{ij}) = \chi_{i+1,j} + \chi_{i-1,j} + \chi_{i,j+1} + \chi_{i,j-1} - 4(1+\gamma_{ij})\chi_{ij} - \alpha_{ij} = 0$$

We may solve this equation by an extrapolated Liebmann process, exactly

as we did the Poisson equation (3.4.9). By analogy with (3.4.10) we define the iteration process

$$\chi_{ij}^{\nu+1} = \chi_{ij}^{\nu} + \alpha \bar{\chi}^{\nu,\nu+1}$$

It is difficult to obtain the optimum α and exact convergence criteria, except where γ_{ij} is constant. In this case, setting $a = 1 + \bar{\gamma}_{ij}$ we can show by an adaptation of Frankel's (1950) method that the optimum α is $\left[2a(1+\sin\theta)\right]^{-1}$ where $\cos\theta = (2a)^{-1}\left(\cos\frac{\pi}{n} + \cos\frac{\pi}{m}\right)$. It was found experimentally that the α corresponding to the mean value of γ_{ij}, $\bar{\gamma}_{ij}$, gives as rapid convergence for the solution of (12) with variable γ_{ij} as constant $\bar{\gamma}_{ij}$. If a is appreciably greater than one, the convergence rate of the extrapolated Liebmann method is considerably better for (12) than for (9).

The computational stability criteria for the system (3) to (6) may be derived in an analogous manner to that used for the barotropic equation. If terms that do not lead to an exponential amplification when $\Delta s \to 0$ ar ignored, the stability criteria become

$$\Delta s/\Delta t > \sqrt{2}\,|v'| \quad , \quad \Delta s/\Delta t > \sqrt{2}\,|v|$$

where $|v'|$ is the maximum wind speed at 300 mb, and $|v|$ is the maximum at 700 mb. Thus from the standpoint of computational stability the finite difference analogues of (3) and (4) behave precisely as advection equations in which the velocity field is independent of the quantity being advected.

Because of the high particle velocities at 300 mb it was necessary to take Δt = 1/2 hour for Δs = 300 km; but to shorten the computation time, the accelerating scheme outlined earlier for the barotropic

integration was used; the potential vorticities q and q' were advected in their respective velocity fields for three successive half-hour time steps, and only then were the new ϕ and ϕ' computed by solving the difference analogues of (7) and (8) (the $3J/L$ process). Here, because of the small value of Δt it was thought that the truncation errors introduced by using non-contemporaneous velocity fields would be small. As a check, a single 12-hour forecast was made using $1J/L$. The difference between the two methods was found to be large. Unfortunately, this comparison was made only after the main series of six forecasts had been completed. Those forecasts in which the error was expected to be large, however, were later recomputed using the $1J/L$ method. A more accurate time saving method would be to employ a procedure analogous to that recommended for the barotropic integration, namely to store the penultimate ϕ and ϕ' and extrapolate to obtain centered values for the Jacobian computation.

On the basis of the experience gained from the study of round-off error in the barotropic integrations it was decided to allot twelve binary digits for the storage of ϕ^{t}, and 14 binary digits each for the storage of q^{t} and q^{t-1} at each level. It was thus possible to store all the data pertaining to one grid point as two forty-digit "words".

With use of the same horizontal grid dimensions as in the barotropic model the length of time required to compute a 24-hour forecast was a little more than twice that for the main barotropic model. Each 1 1/2 hour forecast involved the calculation of three fields of Jacobians for each level, and the solution by iteration of (9) and (7). With the machine operating at full speed, 1 1/2 minutes sufficed for the six

Jacobian calculations, 2 1/2 minutes for the solution of (9) and 3 1/2 minutes for the solution of (7). Thus a total time of 7 minutes per 1 1/2 hour time step, or one hour and 52 minutes per 24-hour forecast, was taken. This time could no doubt be considerably reduced by the use of proced[illegible] similar to those by which the barotropic forecast time was reduced from 48 to 6 minutes.

It is not necessary to solve the system (7) and (8) by reduction to two equations each for a single dependent variable. The system may be solved implicitly by an extension of the extrapolated Liebmann method. As this alternative method was used in the 3-layer model a description will be left until the next section.

3.6. Integration of the 3-level model.

The system of equations (3.3.7), (3.3.8), and (3.3.9) for the n layer model are clearly non linear and therefore it is convenient for computational purposes to derive a linear approximation to them. This is done by replacing $(D/Dt)\ln x$ by $(1/\bar{x})(D/Dt)x$ where logarithms occur in equations (3.3.3), (3.3.4), and (3.3.5). The elimination of $\omega_{k-1/2}$ and $\omega_{k+1/2}$ now gives

$$\left(\frac{D}{Dt}\right)_k \left\{ \zeta_k - K_{k+1/2}\frac{\bar{\zeta}_k}{\bar{\phi}_k - \bar{\phi}_{k+1}}(\phi_k - \phi_{k+1}) + K_{k-1/2}\frac{\bar{\zeta}_k}{\bar{\phi}_{k-1} - \bar{\phi}_k}(\phi_{k-1} - \phi_k) = 0 \right. \qquad k \neq 1, n.$$

together with equations analogous to (3.3.8) and (3.3.9) for $k = 1, n$.

The equations so far derived have been for an atmosphere divided into n equal pressure intervals. By transforming the vertical coordinate and using centered finite differences in terms of n equal intervals of the new coordinate a similar set of equations, differing only in the value of constants from the previous set, can be obtained. A

convenient transformation is $\sigma = (p/p_0)^2$. Making the transformation and linearizing we obtain the set of equations

$$\left(\frac{D}{Dt}\right)_k \left\{ \zeta_k - \frac{\bar{\zeta}_k}{\bar{\phi}_k - \bar{\phi}_{k+1}} \kappa_{k+\frac{1}{2}} \frac{(\sqrt{\sigma})_k}{(\sqrt{\sigma})_{k+\frac{1}{2}}} (\phi_k - \phi_{k+1}) + \frac{\bar{\zeta}_k}{\bar{\phi}_{k-1} - \bar{\phi}_k} \kappa_{k-\frac{1}{2}} \frac{(\sqrt{\sigma})_k}{(\sqrt{\sigma})_{k-\frac{1}{2}}} (\phi_{k-1} - \phi_k) \right\} = 0$$

$k \neq 1, n$

with again corresponding equations for $k = 1$ and n .

With either the p or σ -coordinate system the linearized equations for 3 layers can be written

$$\left(\frac{D}{Dt}\right)_k \left\{ \zeta_k + \alpha_{k1}\phi_1 + \alpha_{k2}\phi_2 + \alpha_{k3}\phi_3 \right\} = 0 \quad , \quad k = 1, 2, 3.$$

where α_{k1} , α_{k2} , α_{k3} are constants.

We now define

$$q_k = \nabla^2\phi_k + \frac{f}{m^2}\left\{ \alpha_{k1}(\phi_1 - \bar{\phi}_1) + \alpha_{k2}(\phi_2 - \bar{\phi}_2) + \alpha_{k3}(\phi_3 - \bar{\phi}_3) \right. \quad , \quad k = 1, 2, 3$$

The assumption is now made that the (f/m^2) in the expression for q_k can be regarded as a constant. (This is equivalent to having a North-South gradient of static stability.) We therefore have

$$q_k = \nabla^2\phi_k + \beta_{k1}(\phi_1 - \bar{\phi}_1) + \beta_{k2}(\phi_2 - \bar{\phi}_2) + \beta_{k3}(\phi_3 - \bar{\phi}_3) \quad , \quad k = 1, 2, 3$$

where β_{k1} , β_{k2} and β_{k3} are constants.

Equation (3) can now be written

$$\frac{\partial q_k}{\partial t} = J\left(\frac{m^2}{f} q_k \, , \, \phi_k\right) + \bar{J}(f, \phi_k) \quad , \quad k = 1, 2, 3.$$

We now approximate equation (6) by taking $1/f$ to be constant during the evaluation of the left-hand Jacobian at a point(i , j , k) and m^2 to be constant over the whole grid, i.e.

$$\frac{\partial q_k}{\partial t} = \frac{m^2}{f} J(q_k, \phi_k) + J(f, \phi_k)$$

Equations (5) and (6) together with the appropriate boundary conditions form the basis of the forecasting system, the procedure being as follows:

1. With q_{ijk}^{t-1} and q_{ijk}^{t} stored for all grid points -- determined at $t = 0$ from ϕ^0 by (5) -- equation (5) is solved for ϕ_{ijk}^{t} using $2\phi_{ijk}^{t-1} - \phi_{ijk}^{t-2}$ as an initial guess and using the boundary condition that ϕ_{ijk} is constant on the lateral boundaries.

2. $(\partial q_k / \partial t)_{ij}^{t}$ is determined from the finite difference analogue of (7)

3. At interior points q_{ijk}^{t+1} is obtained from

$$q_{ijk}^{t+1} = q_{ijk}^{t-1} + \epsilon(t)\, \Delta t\, (\partial q / \partial t)_{ijk}^{t} \quad , k = 1, 2, 3$$

At $t = 0$ it is necessary to employ non-centered differences, that is, to set $q_{ijk}^{-1} = q_{ijk}^{0}$ and $\epsilon(0) = 1$. At all other times $\epsilon(t) = 2$. On the boundaries q_{ijk}^{t} is kept equal to its initial value of $\nabla^2 \phi_{ijk}^{t=0}$.

4. Steps 1 to 3 are repeated until the required forecast time is spanned.

Two iterative methods are available for the solution of equation (5). The extrapolated Liebmann process may be applied either to solving for all the ϕ's simultaneously, or to solving a transformed system each of whose equations contain but one dependent variable as was done in the two layer integrations. This second system, however, suffers from the disadvantage that more conservative scaling has to be

used in order to ensure that all transformed quantities remain scaled, that the transformation process gives rise to round-off errors and that the resultant simplification of the equations does not compensate, in the case of 3-layers, for the additional time taken to perform the transformation. For these reasons the method of simultaneous solution was eventually adopted after trial computations using both methods.

Equation (5) can be written in the form

$$\mathcal{L}(\hat{\phi}_{ijk}) = \nabla^2 \hat{\phi}_{ijk} + s_{k1}\hat{\phi}_1 + s_{k2}\hat{\phi}_2 + s_{k3}\hat{\phi}_3 - \frac{1}{B}\mathcal{Q}_k \qquad k = 1, 2, 3.$$

where $\phi_k - \mathcal{F}_k = B\hat{\phi}_k$ and $\nabla^2\hat{\phi}_{ijk} = \hat{\phi}_{i+1,j,k} + \hat{\phi}_{i-1,j,k} + \hat{\phi}_{i,j+1,k} + \hat{\phi}_{i,j-1,k} - 4\hat{\phi}_{ijk}$

We may now define an iteration procedure

$$\hat{\phi}_{ijk}^{\nu+1} = \hat{\phi}_{ijk}^{\nu} + \gamma_k \mathcal{L}(\hat{\phi}_{ijk}^{\nu,\nu+1}) \quad , k = 1, 2, 3$$

This procedure is applied in turn to each k level until $|\phi_{ijk}^{\nu+1} - \phi_{ijk}^{\nu}| < \varepsilon$

Previous experience with the barotropic model showed that the optimum convergence rate for the iteration process occurs for a value of the relaxation constant γ_k lying between those given by Frankel (1950) and Liebmann (1918), respectively. Therefore, in the 3-layer computations the relaxation constant γ_k for each k level was computed first on Liebmann's basis (this gives no over-relaxation) and then these were adjusted to give approximately 15% over-relaxation. Owing to the satisfactory behavior of this method no attempt was made to obtain better values of the relaxation constants.

The time interval over which $\mathcal{Q}_k$ can be extrapolated by means of equation (8) is, as in the previous models, limited by the necessity to maintain computational stability. The criterion as before is that

$$\Delta s/\Delta t \geq \sqrt{2}\,|v|_{max}$$

where $|v|_{max}$ is the maximum particle speed in the forecast area. As in most of the 3-layer forecasts the $\frac{1}{2}$ = 1 was taken to be 200 mb it was necessary, for a grid interval, Δs, of 300 km, to limit the time interval Δt to one-half hour. No attempt was made to apply an $n\bar{J}/L$ method as in the two level forecasts. A time interval of one-half hour was thus chosen and in no case considered, in so far as the number of iterations taken to solve equation (9) could be regarded as a criterion, was the computation, for this reason, found to be unstable.

The lateral boundary condition stated previously was also, by the same criterion, found to be stable. This was not the case with computations made, as in the previous models, using an extrapolated value of $q_{\frac{1}{2}}$ on the boundary at an outflow point when evaluating the Jacobian in (7).

One memory word was used to store the data for each grid point, the information being stored in the following way: q^t was stored in ten binary places, $4(q^t - q^{t-1})$ in 8 binary places, ϕ^{t-1} in 10 binary places and ϕ^t in 12 binary places. For a 16 x 16 grid this means that the storage of data for the 3 levels occupied three quarters of the Williams memory. As the remaining 256 words were not sufficient to contain the required code the code was divided into two parts corresponding respectively to the Liebmann and Jacobian plus extrapolation sections. These two sections were put on the magnetic drum and alternately read into the Williams memory as required by the procedure previously outlined.

The iterative process was continued until $|\hat{\phi}^{\nu+1} - \hat{\phi}^{\nu}| \leq 2^{-9}$.

This is equivalent to continuing the iterative process until the error in the height deviation from the standard atmosphere height is less than approximately four feet.

The main series of six 24-hour forecasts for a 16 x 16 grid was made using the p -coordinate system in the following way: Considering the top and bottom of the atmosphere to be at 25 mb and 1000 mb, respectively, the levels k = 1, 2, 3 correspond to 187.5, 512.5 and 837.5 mb, respectively. Since these are not synoptic levels it was assumed that the height deviations from the standard atmosphere values at these levels were the same as those at 200, 500, and 850 mb, respectively. The initial data were thus taken from these levels and the resultant predictions assumed to be for the same levels.

Two additional 24-hour forecasts were made using the σ -coordinate system derived in the following way. Considering the top and bottom of the atmosphere to be at 0 and 986 mb, respectively, the levels k = 1, 2, 3 correspond to 402, 697, and 900 mb, respectively. Following the procedure adopted in the p -coordinate system it was assumed that the height deviations at these levels were the same as those at 400, 700, and 900 mb, respectively.

The computing time for a twenty-four-hour forecast for a 16 x 16 grid was 48 minutes divided approximately equally between the Jacobian and Liebmann sections of the code. As in the two previous models it would no doubt, by a careful reconsideration of the code, be possible to reduce the computing time. Owing to the limited size of the Williams memory, however, any speeding up in the more complicated models is limited by the amount of the memory which has to be devoted to data storage even

when several quantities are stored per word, a process which in itself is quite time consuming.

3.7 Summary of results.

The period selected for trial forecasts with the various methods was 0300 GMT 23 November to 1500 GMT 26 November 1950, and the forecast region was the eastern part of the United States and southern Canada, an area with a relatively dense network of observing stations. This particular period was selected because of the unusually large amount of cyclonic development which occured in the 24-hour period following 1500 GMT 24 November. This storm, which was not correctly predicted at the time, caused a considerable amount of damage to Princeton and neighborhood. The Meteorology Group thus felt a personal interest in this situation.

Twelve and 24-hour forecasts were prepared using each of the three previously described models from each of the six initial maps: 0300 and 1500 GMT, 23, 24 and 25 November 1950. Each 24-hour forecast from an initial map overlapped the first half of the succeeding 24-hour forecast, with the exception, of course, of the last. For each forecast a chart of the forecast height changes of the relevant pressure surface was made and correlated against the observed changes. These results are shown in Figure 1 which gives the correlation coefficients between the 24-hour forecast and observed changes from six initial maps for each of the three methods. It will be seen that while all the methods give reasonable results for the beginning and end of the period, only the three-layer model gave a good prediction for the central period when the storm was developing.

As mentioned previously two forecasts were made using a three-layer model with the σ -coordinate system. Figure 2 shows the 24-hour forecast from November 24, 1950, 1500 GMT for the 900 mb level. (Charts are of height deviations from the U. S. Standard Atmosphere values in 10's ft.) Forecasts for 6, 12, 18 and 24 hours are given together with the correct 24-hour map. It will be seen that a completely new center of low pressure developed. The full line in (f) shows the actual position and movement of the new center and the dashed line shows the forecast movement. The differences between (e) and (f) are thus seen to be due to the slightly wrong position of the initial development. The intensity of the low pressure region and the very great increase in wind strength (from 30 to 100 knots) over the eastern United States are, however, correctly predicted. As this is the first case of development being predicted by numerical methods it gives rise to the hope that more physically correct models will give even better numerical predictions and increased accuracy in the forecasting of the associated weather phenomena such as cloud amount and precipitation.

REFERENCES for Meteorology

Charney, J. G., R. Fjortoft, and J. von Neumann, 1950: Numerical integration of the barotropic vorticity equation. Tellus 2, 237-254.

Charney, J. G. and N. A. Phillips, 1953: Numerical integration of the quasi-geostrophic equations for barotropic and simple baroclinic flows. Journal of Meteor. 10, 71-99.

Liebmann, H., 1918: Die angenäherte Ermittelung harmonischer Functionen und konformer Abbildungen. Bayer. Akad. Wiss. math.-phys. Klasse, Sitz. 1918, 385-416.

Frankel, S. P., 1950: Convergence rates of iterative treatments of partial differential equations. Math. Tables and other Aids to Computation, 4, No. 30, 65-70.

(REFERENCES for Meteorology - continued)

Young, D. M., 1951: Iterative methods for solving partial difference equations of elliptic type. Seminar given at Numerical Analysis Seminar of Ballistic Research Laboratories, Aberdeen Proving Ground, Maryland.

Richardson, L. F., 1910: The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. Phil. Trans. Roy. Soc. London, A, 210, 307-357.

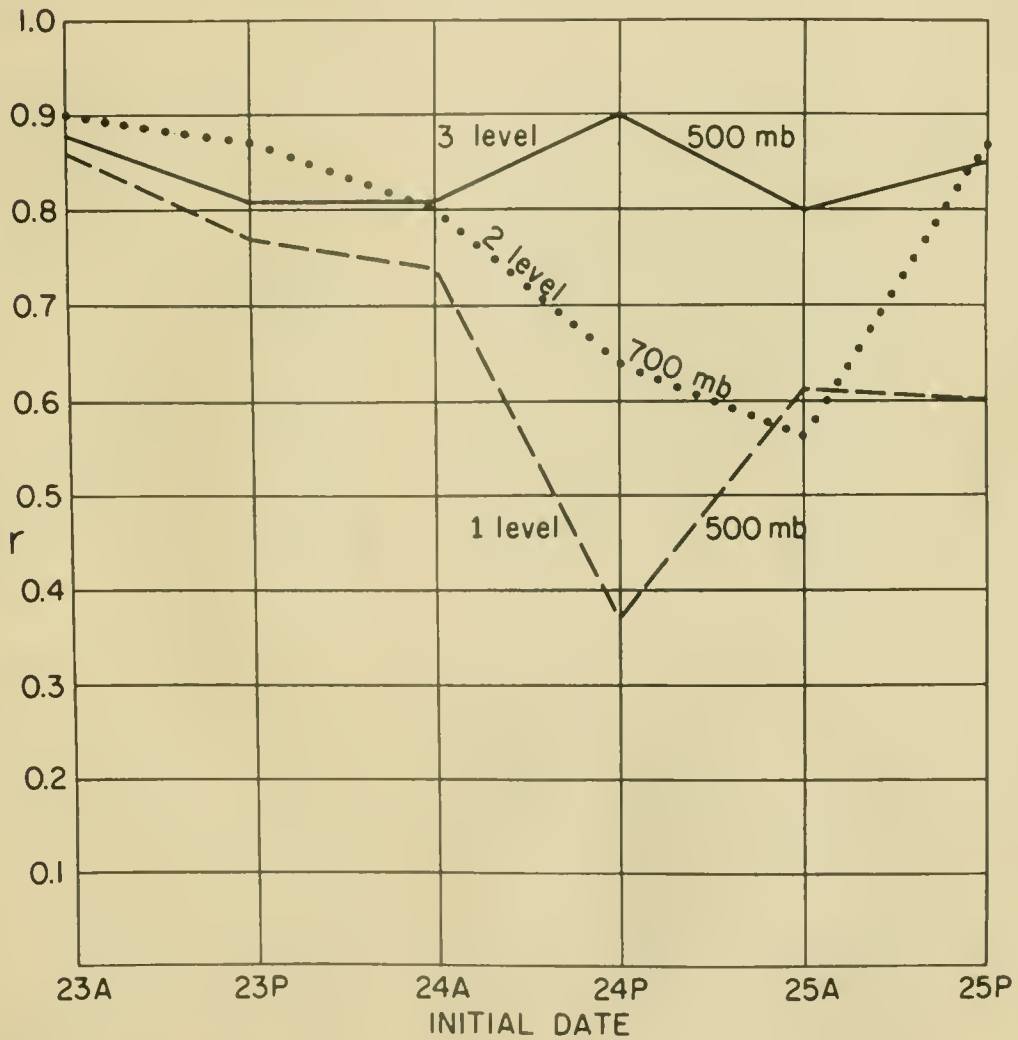


Figure 1.

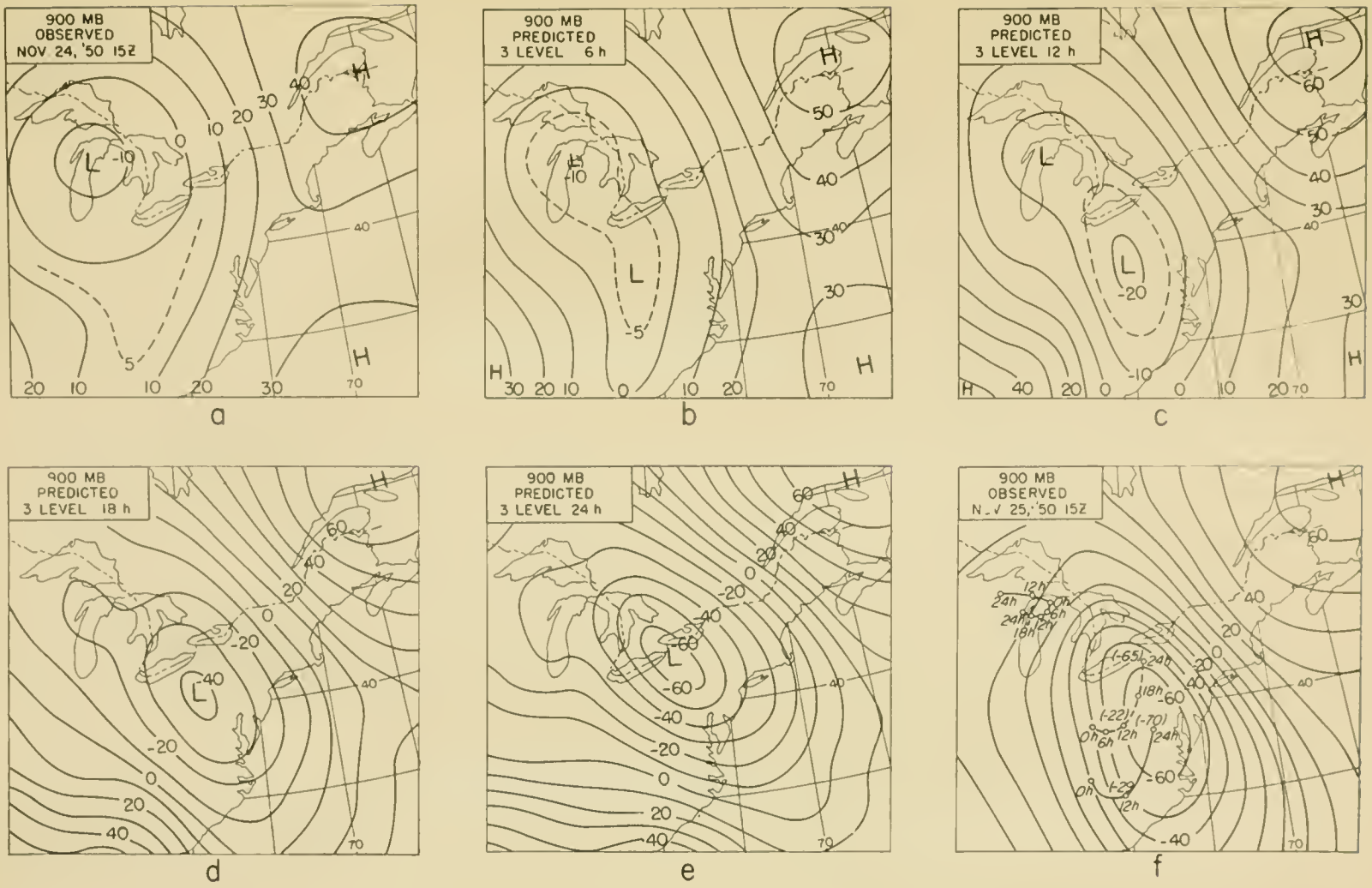

900 MB
OBSERVED
NOV 24, '50 15Z
a
900 MB
PREDICTED
3 LEVEL 6 h
b
900 MB
PREDICTED
3 LEVEL 12 h
c
900 MB
PREDICTED
3 LEVEL 18 h
d
900 MB
PREDICTED
3 LEVEL 24 h
e
900 MB
OBSERVED
N.V 25, '50 15Z
f

Figure 2.

vanced Study, Princ , N. J.
uter Project.
on contract no. DA-36-034-ORD-
ff, Electronic Computer Project
.
tables, diagrs. 28cm.
the numerical analysis, program
carried out under the terms of
-ONR-388, T. O. I. and Contract
T. O. I.
calculating-machines.